Relations structure - Fonction dans la superfamille des Cytochromes P450

Thien-An Nguyen

UNIVERSITÉ PARIS . DIDEROT
PARIS 7
ÉCOLE DOCTORALE INTER /// BIO

RELATIONS STRUCTURE - FONCTION DANS LA SUPERFAMILLE DES CYTOCHROMES P450. ÉTUDES BIOINFORMATIQUES

THÈSE DE DOCTORAT

présentée et soutenue publiquement le 29 Octobre 2007

pour l'obtention du

DOCTORAT DE L'UNIVERSITÉ PARIS DIDEROT
ANALYSE DE GÉNOMES ET MODÉLISATION MOLÉCULAIRE

Par

Thiên-Ân NGUYÊN

Composition du jury

Rapporteurs :	Dr. Philippe URBAN Dr. Alexandre de BREVERN
Examinateurs :	Pr. Catherine ETCHEBEST , présidente Dr. Isabelle CALLEBAUT Dr. Patrick DANSETTE
Directeurs de thèse :	Dr. Jean-Michel NEUMANN Dr. François ANDRÉ

CEA/iBiTec-S/SB2SM/LSOD, URA 2096 CNRS CEA Saclay F91191 Gif-Sur-Yvette Cedex

Résumé

Les cytochromes P450 (CYP) sont des enzymes responsables de la biotransformation de composés exogènes, aussi bien dans les phénomènes de détoxication que d'intoxication par formation d'entités réactives. La forme hépatique humaine la plus abondante (CYP3A4) est responsable du métabolisme de plus de 60 % des médicaments utilisés actuellement, entraînant de nombreuses interactions médicamenteuses indésirables. La connaissance des mécanismes moléculaires de fonctionnement de ces CYPs au moyen de modèles prédictifs est d'un intérêt primordial pour les industriels. L'obtention de ces modèles par modélisation comparative est toutefois pénalisée par la dispersion en séquences dans cette superfamille.

Une méthode originale de reconstruction des CYPs basée sur l'identification au sein de cette famille des blocs structuralement conservés (CSB) est proposée ici. Ces CSBs définissent un repliement commun aux CYPs et sont considérés comme la signature structurale de la superfamille. Les CSBs sont codés en termes d'informations statistiques (*profile*) puis alignés sur les séquences de CYP de structure inconnue, par un outil d'alignement multiple (Caliseq) créé pour produire l'alignement multiple optimal pour les reconstructions par modélisation comparative. Caliseq sert aussi à détecter des séquences originales de CYP dans une banque ou un génome.

Le modèle structural obtenu permet de suggérer des mutations pour observer les modifications du comportement de la protéine vis-à-vis de ses substrats spécifiques. Le cas du CYB2B6 est un exemple concret où le modèle a suggéré des mutations permettant d'augmenter l'affinité de l'enzyme pour un substrat spécifique utilisé en chimiothérapie.

Mots clef : Cytochrome P450, Modélisation comparative à bas taux d'identité, Génomique exploratoire à partir de signature structurale, CYP2B6.

Abstract

Cytochromes P450 (CYP) are monoxygenase enzymes involved in biotransformations of exogenous compounds in detoxification processes, but also in intoxication by generation of reactive species. CYP3A4, the most abundant isoform present in human liver, is responsible for the metabolisation of more than 60% of drugs used in therapy, leading to unwanted drug-drug interactions. The knowledge and understanding of mechanisms underlying substrate recognition and transformation is of outstanding interest for setting up reliable predictive tools in pharmaceutical companies. Achievement of CYP models by homology modeling is however limited by the high diversity of sequences in this superfamily.

An original method based on the identification of Common Structural Blocks (CSB) within the family, is proposed here to rebuild structural models despite their low sequence identity. CSBs define a common fold of the CYPs and are used as a structural signature of the superfamily. A multiple alignment tool able to align successive profiles (calculated in each CSB) on the CYP sequences of unknown structure, has been developed to make use of the structural information beneath the CSBs, in one hand to give reliable multiple alignments used in homology modeling, and in the other hand, to search for new CYP sequences in databanks or genomes.

When the structural model is obtained, *in silico* mutation experiments can be performed to monitor changes in the behavior of the protein towards its specifics substrates. A successful example is shown through CYP2B6 case, in which mutations suggested by the model led to significant increase of its affinity towards a specific substrate used in chemotherapy.

Keywords: Cytochrome P450, low identity Homology Modeling, Genomic exploration using structural signature, CYP2B6.

Table des matières

Abréviations et acronymes

ADN	=	Acide Désoxyribonucléique
CASP	=	Critcal Assessment of protein Structure Prediction
CATH	=	Class Architecture Topology Homology
CPA	=	Chlorophosphamide
CPI	=	4-(4-chlorophenyl)imidazole
CSB	=	Common Structural Block (fr. block structuralement commun ou conserve)
CYP	=	Cytochrome P450
Da	=	Dalton
EMBL	=	European Molecular Biology Laboratory
EMX	=	Enzymes du Métabolisme des Xénobiotiques
GDEPT	=	Gene-Directed Enzyme Prodrug Therapy
HMM	=	Hidden Markov Model (fr. modèle de Markov caché)
indel	=	Insertion Délétion
Kd	=	Constante de Dissociation
Km	=	Constante de Michaelis
M	=	Molaire
MD	=	Molecular Dynamics (fr. Simulation de Dynamique Moléculaire)
P450	=	Cytochrome P450
PDB	=	Protein Data Bank
PSSM	=	Position Specific Scoring Matrix ou Profil
RMN	=	Résonance Magnétique Nucléaire
RMSD	=	Root Mean Square Deviation (fr. écart quadratique moyen)
SCOP	=	Structural Classification Of Proteins
SRS	=	Substrate Recognition Site (fr. Site de reconnaissance du substrat)
SSE	=	Structural Secondary Element (fr. élément de structure secondaire)

Introduction générale

En l'espace d'un demi-siècle, le Cytochrome P450, est devenu un véritable emblème dans la communauté scientifique des biologistes tant l'engouement que cette protéine suscite est intense. Depuis sa découverte dans les années 50, elle ne cesse de poser de nouvelles questions aux chercheurs. En réalité, il n'existe pas un Cytochrome P450, mais toute une variété : ces cytochromes P450s correspondent en effet à une large superfamille d'hémoprotéines mono-oxygénases, présente à la fois chez les procaryotes et les eucaryotes. Ces enzymes jouent un rôle important dans le métabolisme oxydatif d'une grande diversité de substrats pour la plupart hydrophobe, aussi bien d'origine endogène qu'exogène. Ainsi, chez les cellules de mammifère, ils sont notamment responsables de la métabolisation et de la détoxication des composés exogènes, également appelés xénobiotiques. On les trouve aussi impliqués dans les phénomènes d'intoxication par formation d'entités réactives : époxydes, radicaux, etc. Ces enzymes de phase I du métabolisme, assurent une modification du potentiel redox de ces xénobiotiques par mono-oxygénation ou réduction, modifiant leur liposolubilité et rendant possible leur excrétion après conjugaison ou non par des enzymes de phase II. Les P450s, de 400 à 500 acides aminés environ, contiennent une proto-porphyrine IX de fer (l'hème) qui joue le rôle de catalyseur d'oxydo-réduction, et peuvent être solubles ou membranaires. Les réactions d'oxydo-réduction entraînent l'activation de l'oxygène qui est transférée sur le substrat positionné entre 3 et 6 Å du fer (Johnson, 2003).

En raison de leurs propriétés physico-chimiques et de leur implication dans les réactions de détoxication, les cytochromes P450 constituent donc un enjeu à la fois économique et pharmaceutique majeur : les industriels pharmaceutiques doivent en effet faire face à ces enzymes qui reconnaissent et dégradent la majeure partie de leurs médicaments actuellement mis sur le marché. De plus, cette dégradation peut entraîner des interactions médicamenteuses non désirées capable de provoquer des effets toxiques. Les laboratoires de recherche doivent pourtant composer avec ces protéines qui constituent une des premières barrières naturelles aux agressions extérieures. C'est pourquoi il est impératif d'étudier et de comprendre les mécanismes impliqués dans la reconnaissance du substrat par ces CYPs, pour pouvoir espérer soit contourner cette barrière par diminution les doses de médicaments prescrits, soit agir directement sur ces enzymes et les modifier à notre avantage, par des expériences de mutation.

La thématique des P450s depuis, tout particulièrement les P450s impliqués dans la cascade de détoxication chez l'homme est une des activités majeures de l'équipe dirigée par M. Delaforge et F. André au sein de laquelle j'ai effectué mon travail de thèse. Elle concerne aussi bien l'activité catalytique de ces enzymes (multi-spécificité et coopérativité) que leurs relations structure-activité vis-

à-vis de composés hydrophobes (médicaments, pesticides, mycotoxines, ….). Fort de leurs connaissances et de leurs savoir-faire expérimentaux, les membres de cette équipe ont décidé d'aborder la problématique de la **compréhension des mécanismes de reconnaissance par des approches nouvelles : celles de la bioinformatique**. En effet, la bioinformatique est devenue aujourd'hui un outil puissant et complémentaire des autres outils qu'on utilisait jusqu'alors en biologie pour étudier le fonctionnement des protéines (études spectroscopies, études enzymologiques d'activités et mutagénèse, etc.). Le développement d'algorithmes de plus en plus nombreux et la montée en flèche de la puissance de calcul des ordinateurs, ont fait de l'informatique un outil parfaitement adapté à la biologie dont on attend beaucoup, en particulier dans le domaine de l'ingénierie et la conception de médicaments. C'est dans ce contexte que j'ai pris mes fonctions dans cette équipe en Octobre 2004, pour explorer *in silico* les mécanismes généraux de reconnaissance par les CYPs, pour expliquer à la fois la sélectivité à l'intérieur d'une famille de substrats, la multi-spécificité et les effets coopératifs entre substrats de famille chimique différente (interaction médicamenteuse).

En bioinformatique structurale, le point de départ avant toute investigation est la disposition d'un support de travail qui est soit la structure de la protéine, soit un modèle. Même si aujourd'hui un nombre important de structure de CYPs est disponible, il demeure insignifiant comparé au nombre de gènes connus dans cette superfamille. À noter que la plupart des structures de P450s connus correspondent à celles qui ont un intérêt clinique ou pharmaceutique. Bref, pour étudier des P450s de structure inconnue, il est nécessaire de produire des modèles. Toutefois, la diversité en séquence (principalement en partie N-terminale, et à un degré moindre en C-terminale) qui existe au sein de cette superfamille est un vrai frein à l'obtention d'un modèle 3D par modélisation comparative.

Le premier point sur lequel j'ai travaillé durant mes trois années de thèse, a consisté à développer **une méthodologie fiable de reconstruction de P450s à bas taux d'identité**. Pour cela, j'ai conduit mes expériences sur un cas particulier des P450s : celui du CYP 3A4. En effet, chez l'homme, la famille de P450 la plus abondante est celle du CYP3A, qui comprend notamment l'isoforme P450 3A4 (CYP3A4, 503 acides aminés). Celui-ci est majoritaire dans le foie, responsable à lui seul du métabolisme de près de 60% des médicaments utilisés actuellement (Lewis, 2003), mais aussi de nombreux produits présents dans notre environnement : pesticides, mycotoxines, additifs alimentaires, etc. Des composés endogènes tels que la testostérone ou des petits dérivés peptidiques (Delaforge et *al*., 1997) sont aussi reconnus et oxydés par ce cytochrome. Les substrats pris en charge sont

généralement hydrophobes, mais de taille, de nature chimique, et de poids moléculaire très variables : de 150 à 1400 Da (Rendic, 2002, Ekins et *al*., 2003).

Un premier modèle moléculaire du CYP3A4 a été développé au laboratoire au cours du travail de thèse de N. Loiseau (Loiseau, 2002). Il a été obtenu par modélisation comparative basée sur les six structures de cytochromes P450 disponibles à l'époque dans la PDB (Protein Data Bank), dont cinq d'espèces bactériennes solubles, avec une moyenne de 20 % d'identité. Ces isozymes partagent une homologie fonctionnelle (activité mono-oxygénase), mais reconnaissent et métabolisent des substrats très différents, parfois spécifiques (camphre par exemple, pour le P450cam ou Camphor 5-monoxygenase, le plus étudié). Un autre modèle de la famille 3A, l'isoforme CYP3A7 qui présente 88% d'identité avec le CYP3A4, et est exprimée principalement dans le foie durant la période fœtale et néonatale (Lacroix et *al*., 1997, de Wildt et al ., 1999), a été construit en 2003 par M. Cottevieille avec un jeu réduit à quatre structures PDB de cytochrome P450, dont trois bactériennes. Le modèle de CYP3A7 devait permettre de comparer les mécanistiques 3A4/3A7 chez l'homme, et d'utiliser leurs propriétés opposées dans le métabolisme des stéroïdes, afin de réaliser des validations croisées des modèles. Toutefois, les modèles obtenus avec pourtant la même stratégie de modélisation, ne sont pas en accord. Ce désaccord provient de l'étape d'alignement des séquences de référence sur la séquence cible, étape cruciale en modélisation comparative. Dans les deux cas pourtant, l'information structurale spécifique des CYPs était utilisée pour améliorer l'alignement multiple. En effet, en dépit de la diversité en séquence, on observe une bonne conservation du repliement global dans la superfamille des P450, au travers des différentes structures déposées dans la PDB qui sont d'origines diverses (bactérienne, fongique ou de mammifère). Cette information structurale servait d'ancrage à l'alignement multiple.

J'ai donc été amené à étudier cette information de conservation structurale au sein de la superfamille des CYPs, puis vérifier et valider l'intérêt de son utilisation pour améliorer l'alignement multiple. Pour cela, il a fallu mettre en place et développer un outil fiable d'alignement multiple (*Caliseq*) permettant de positionner correctement les séquences des P450s de structures connus, sur la famille des 3A. Les alignements produits par cette méthode servent de points de départ aux reconstructions de modèles 3D de CYP 3A et par extension à d'autres P450s ne disposant pas d'autres homologues à plus de 30% d'identité (limite acceptable pour la reconstruction de modèles par des méthodes classiques).

Par ailleurs, le programme développé m'a également permis d'aborder la bioinformatique génomique : en effet, ce logiciel initialement mis au point pour produire les alignements multiples était également en mesure d'exploiter l'information structurale comme signature spécifique des P450s pour **des expériences de génomique exploratoire**. *Caliseq* permet en effet de détecter des séquences de P450s (dans la mesure où l'information de conservation structurale est tirée de la superfamille des P450s) dans des banques de séquences. Cet aspect de génomique exploratoire présente un intérêt dans la construction d'une banque de séquences spécifique au P450s. L'utilisation de cette banque, combinée à celle de la reconstruction de modèles pourrait théoriquement aboutir à une constitution d'une base de modèle de site actif, sur laquelle les chimiothèques pourraient être confrontées. Les résultats des criblages obtenus, apporteront probablement des éléments de réponses, ou du moins une idée sur les mécanismes de reconnaissance du substrat par ces enzymes.

Outre l'aspect prédictif de la bioinformatique, j'ai été également conduit au cours de ma thèse à travailler sur une **application concrète de la bioinformatique structurale dans un traitement par thérapie génique**, impliquant une autre P450, la CYP 2B6. Ce travail a été initié par l'équipe de Toxicologie Moléculaire à la Faculté de Médecine de Paris V qui a développé les techniques de thérapie génique autour d'une prodrogue, dont le produit de dégradation par le CYP présente un effet chimiothérapique. Pour améliorer l'affinité et l'activité enzymatique du CYP 2B6 pour ce substrat, des expériences de mutations dirigées ont été réalisées. En raison du nombre important de mutants opérés, cette équipe a sollicité notre aide pour lui fournir un support bioinformatique en vue de tester *in silico* les résultats. Ce fut donc pour moi l'occasion d'aborder un autre aspect de la bioinformatique apportant une véritable finalisation au travail *in silico*, à savoir utiliser les ressources et les informations disponibles pour fournir une base destinée et une description moléculaire à des expériences biochimiques.

Ce travail de collaboration m'a ainsi permis de confronter des résultats obtenus (mutations *in silico*, docking de la prodrogue dans le site actif, simulations de dynamique moléculaire…) sur des modèles de CYP 2B6 par des résultats expérimentaux et au plan plus général, d'apporter (ou de confirmer) des éléments de réponse aux mécanismes de reconnaissances de substrat par les P450s.

L'ensemble de tous mes travaux est présenté au travers de ce manuscrit, agencé de la manière suivante : dans un premier chapitre, j'exposerai l'état des connaissances actuelles disponibles, aussi bien sur les P450s que sur les méthodes bioinformatiques classiques d'étude de protéine. Je décrirai dans un second chapitre les méthodes que j'ai développées pour compléter les méthodes

bioinformatiques existantes, et en dernier chapitre, je présenterai les résultats obtenus par cette méthode et ceux issus de la collaboration avec l'équipe de Paris V.

Première Partie

État des connaissances

CHAPITRE 1

Cytochromes P450 : pourquoi un tel intérêt ?

« Nous ne connaissons pas le vrai si nous ignorons les causes »

Aristote (384-322 av JC)

1.1 A la découverte d'une superfamille d'enzyme…

1.1.1 Présentation d'une protéine hors norme

Le cytochrome P450 constitue un des sujets de recherche les plus en vogue au cours de ce dernier demi-siècle, aussi bien en biochimie qu'en biologie moléculaire. Il doit également son principal essor au développement de ressources de l'industrie pharmaceutique mondiale. Sous le terme général de cytochrome P450, se regroupe une superfamille multigénique de protéines enzymatiques dites «hemo-thiolate» à propriétés redox. Aussi connu sous le terme de mono-oxygénases, il a la capacité d'activer les molécules de dioxygène en des entités hautement réactives (ROS) et d'insérer ensuite l'oxygène moléculaire dans un nombre important et varié de substrats tant au niveau d'un atome de carbone, que d'azote ou de soufre. Cette activation du dioxygène est rendue possible par la présence d'un atome de fer inclus dans l'hème qu'il porte. Ces cytochromes P450 (CYPs ou P450s) sont des enzymes ubiquitaires qu'on retrouve dans tous les organismes vivants à l'exception près de certains micro-organismes primitifs qui auraient évolué il y a plus de 3,5 milliards d'années. De même, substrats et réactions générés par ces enzymes sont multiples et ubiquitaires : les CYPs sont responsables du métabolisme oxydatif de molécules très diverses, comprenant aussi bien des substances endogènes (hormones stéroïdiennes, acides gras, vitamines D, prostanoïdes, alkanoïdes, terpènes et autres pytoalexines…) que les xénobiotiques (médicaments, pesticides polluants, toxiques, cancérigènes, carcinogènes…). Les réactions de biotransformation des xénobiotiques catalysées par les CYPs s'inscrivent en effet dans un processus de détoxication évitant l'accumulation de substances potentiellement toxiques dans l'organisme. Ce sont d'ailleurs ces mêmes substrats qui auto induisent leurs propres réactions de détoxication. Paradoxalement, les CYPs peuvent parfois catalyser l'activation chimique de certains composés (procarcinogènes…) et produire des métabolites toxiques, mutagènes voir cancérogènes (génération de ROS qui endommagent l'ADN). Cette ambivalence et les conséquences majeures qui en découlent ont conduit à s'intéresser aux cytochromes P450, tant du point de vue structural que fonctionnel.

1.1.2 Identification et caractérisation des CYPs

Cela fait à présent cinquante ans que Garfinkel et Kligenberg (1958) ont reporté pour la première fois l'apparition d'un pigment jaune orangé dans des fractions microsomales hépatiques de rat et de cochon en présence de monoxyde de carbone. Ce pigment fut alors dans un premier temps caractérisé comme une hémoprotéine, identifiée à un cytochrome de type *b* peu commun, puis deviendra plus tard

ce qu'on connaît sous la désignation « P450 ». Cette désignation de « P450 » a été attribué en raison de sa propriété d'absorption maximal à 450 nm dans le spectre UV (cf. Figure 1-1) lorsque le fer de l'hème, à l'état réduit, est complexé au monoxyde de carbone (Omura et Sato, 1962). Le 'P' de P450 fait référence quant à lui au « pigment » d'où le terme P450 pour Pigment à 450 nm. La fonction biochimique des cytochromes P450 fut alors établie vers le début des années 60 aussi bien dans la biotransformation des stéroïdes (Estrabook et *al.*, 1963) que dans l'oxydation des composés exogènes (Cooper et *al.*, 1965).

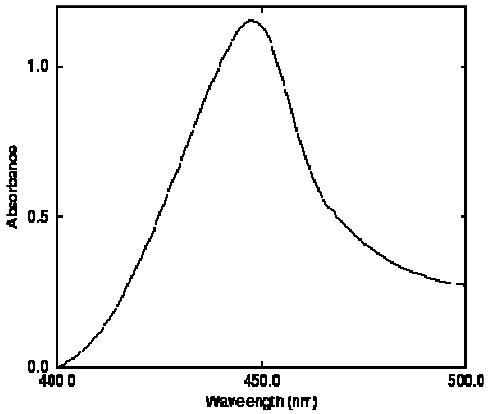


Figure 1-1 Spectre d'absorption du P450 complexé à du monoxyde de carbone montrant le pic caractéristique aux alentours de 450 nm

C'est donc, d'une part l'observation d'une pigmentation cellulaire jaune-orangé apparue à la suite d'une complexion de préparation microsomale avec un monoxyde de carbone et d'autre part, la présence d'un pic distinct à 450 nm en spectre UV visible d'absorption (Garfinkel, 1958 ; Klingenberg, 1958) qui ont permis la découverte de l'enzyme. Les origines phylogéniques des P450s quant à elles, demeurent discutables et restent encore à démontrer. On pense toutefois que le système des P450s aurait évolué de façon constante il y a plus de 3,5 milliards d'années, avant même l'arrivée de l'oxygène atmosphérique (Wickramasinghe et Villee, 1975). Au départ, il semble que les organismes présents se servaient du potentiel chimique des molécules d'oxygène pour traiter les composés organiques, grâce à la formation contrôlée d'entités réactives (*i.e.* superoxydes, peroxydes, etc.). Ces mécanismes de transformation de l'oxygène seraient rendus possibles par la médiation d'un complexe fer-porphyrine dont la structure est similaire de la protoporphyrine IX (ou hème cf. Figure 1-1). La porphyrine est en effet une entité appropriée pour la régulation de l'équilibre de l'état de spin du fer (Frausto da Sliva et Williams, 1991), occupant donc un rôle important dans les échanges électroniques du fer (et donc par la même occasion, dans la catalyse par les hémoprotéines). Ce type de réaction peut être de plus favorisé par l'influence d'une cystéine proximale (thiolate) présente dans l'apoprotéine du P450 qui se charge, d'acheminer les molécules d'eau, les éléments réducteurs (*i.e.* proton/électrons)

nécessaires à la réaction (Lewis et Pratt, 1998) et les substrats organiques à lier. De toute évidence, l'apoprotéine P450 semblerait avoir évolué de façon à incorporer la partie héminique et à « orienter » les substrats potentiels vers un métabolisme oxydatif d'une position particulière. L'apport en électron peut être obtenu quant à lui par transfert à partir de partenaires redox pouvant appartenir à un système immergé dans une bicouche phospholipidique (comme celle de la membrane du réticulum endoplasmique dans le cas des P450s hépatiques microsomaux).

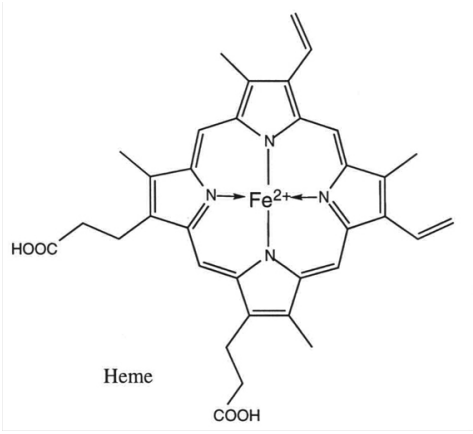


Figure 1-2 Protoporphyrine IX de fer constituant l'hème des cytochromes P450

Au final, tous les P450s identifiés à ce jour désignent des hémoprotéines constituées d'un hème relié à l'apoprotéine par un groupement cystéinate complexant le fer de groupe prosthétique (cf. Figure 1-3). Schématiquement, on a coutume de représenter les CYPs comme une grosse hémoprotéine constitué d'une chaîne polypeptidique unique, l'*apoprotéine* (d'un poids moléculaire compris entre 45 et 60 kDa) et d'un groupement prosthétique central, l'*hème* (catalyseur de la réaction enzymatique). Ce dernier est lié de façon « non covalente » à l'apoprotéine par l'intermédiaire d'une cystéine. Le site catalytique de l'enzyme est hydrophobe, permettant l'accueil de substrats.

- La diversité des P450s est naturellement liée à la séquence primaire en acides aminés de l'apoprotéine.
- L'hème des CYPs est constitué d'une protoporphyrine XI de Fer. Au sein des P450s, ce métal est lié aux quatre azotes pyrroliques de la porphyrine et au soufre de la cystéine axiale. Cette cinquième liaison de coordination du fer est réalisée avec le groupement thiolate de la cystéine présente dans la région C-terminale. Le fer peut être hexacoordonné, le sixième ligand étant par exemple, O_2, H_2O, CO ou une autre molécule. Dans le P450, le plan de l'hème définit deux régions : (a) la face *proximale* située du côté du ligand cystéinate et (b) la face *distale* contenant le site actif de l'enzyme.

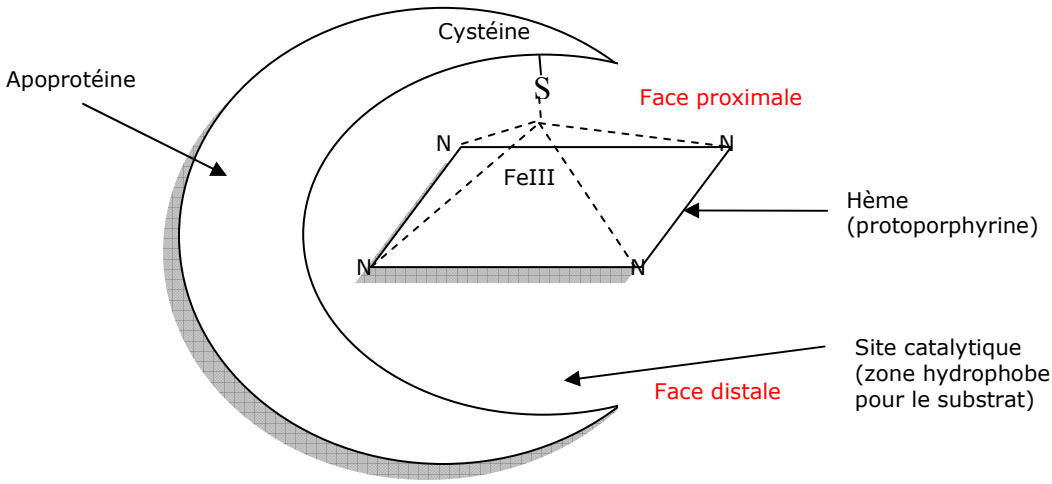


Figure 1-3 Représentation schématique d'un CYP

1.1.3 Diversité des cytochromes P450

Depuis sa découverte, l'intérêt que cette superfamille suscite n'a cessé d'augmenter de façon considérable : aujourd'hui encore, le CYP reste un challenge scientifique majeur. Les trois décennies qui séparent la première purification d'un P450 en 1970 (Yu et Gunsalus, 1970) de la première résolution d'une structure d'un P450 de mammifère en 2000 (Williams et *al.*, 2000b), ont ainsi permis d'appréhender efficacement la complexité et la diversité de cette superfamille.

1.1.3.1 Multiplicité en gènes

On remarque tout d'abord chez cette superfamille une multiplicité en gènes : Lewis reportait déjà en 2000 plus de 1200 gènes individuels de P450. Selon lui, beaucoup d'autres nouveaux gènes viendront grossir les rangs dans les années à venir, tant l'état de connaissance dans ce domaine est vaste, profond ne demandant qu'à être exploré. Ses intuitions se sont révélées exactes, puisque aujourd'hui (2007), nous atteignons déjà 7700 gènes répartis entre les bactéries, végétaux, champignons, insectes et animaux. De plus, d'une espèce à l'autre, le nombre de gènes codant pour les P450s est très variable.

À partir d'informations basées sur des alignements de séquences et d'arbres phylogéniques d'évolution Nelson a émis l'hypothèse d'un gène ancestral des P450s qui serait apparu il y a environ 3.5 milliard d'années chez la bactérie (Nelson, 1999; Nelson et *al.*, 1993, 1996). Ce dernier serait très proche du seul P450 présent dans tous les organismes : le CYP51. Ce P450 catalyse la 14-α-

déméthylation du lanostérol, un intermédiaire qui intervient dans les premières phases de la chaîne de synthèse des stérols. La diversification des CYPs pourrait donc être expliquée par une complexification de la biosynthèse des stérols au cours de l'évolution. Par ailleurs, l'apparition d'une « menace oxygène » dans le monde végétal ou animal expliquerait également la différentiation des CYPs, spécialisés dans la détoxication des xénobiotiques. Malgré l'importante diversité génique des cytochromes, cette superfamille est caractérisée par deux séquences protéiques consensus :

- Une séquence consensus située du côté proximal de l'hème **FxxGx(R/H)CxG** (il s'agit là du motif décrit de façon historique). Appelé « Cys-Pocket », elle contient la cystéine ligand du fer et est considérée comme la signature peptidique de la superfamille des P450s. D'ailleurs, ce motif est également désigné « signature cytochrome P450 – hème » et est utilisé pour identifier les P450s dans la base PROSITE (voir le paragraphe sur Prosite page 79) pour la recherche des P450s. Le motif de la base PROSITE s'écrit sous cette forme : [FW]-[SGNH]-x-[GD]-{F}-[RKHPT]-{P}-C-[LIVMFAP]-[GAD]. Cette séquence participe au maintien de l'hème au sein de l'apoprotéine.

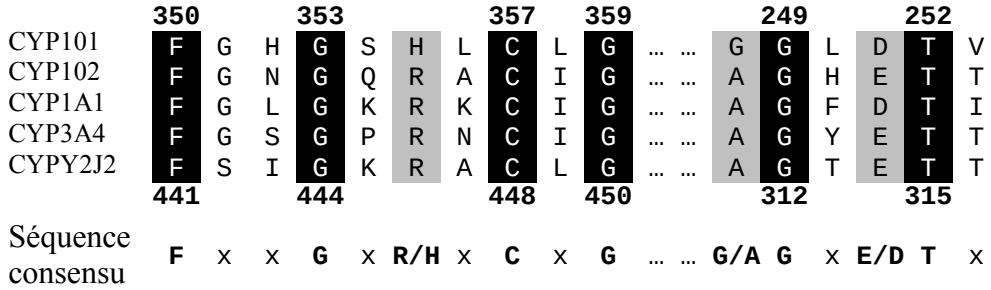


Figure 1-4 Séquences consensus « Cys-Pocket » caractéristiques des cytochromes P450 (source : thèse de P. Lafite)

- Une séquence consensus distale **(G/A)Gx(E/D)T** moins bien conservée au sein des P450s, elle joue un rôle dans l'activation de l'oxygène moléculaire lors de la catalyse.

En raison du nombre important de P450s exprimés dans le monde vivant, une nomenclature en famille et sous famille basée sur l'identité en séquence a été élaborée. Cette nomenclature sera détaillée la section 1.2.1.

1.1.3.2 Localisation hétérogène

Les CYPs sont représentés dans tous les organismes vivants, à différents niveaux de localisation cellulaire : dans le cas des procaryotes, les CYPs sont des protéines cytosoliques, tandis que chez les eucaryotes, ils sont membranaires, localisés dans différents compartiments cellulaires. Ainsi, on

distingue les P450s mitochondriaux, les P450s microsomaux, fixés aux membranes du réticulum endoplasmique (RE), et les cytochromes plastidiaux chez les végétaux.

1.1.3.3 Variabilité en forme

On pensait initialement que le P450 n'était qu'un simple et unique cytochrome. Très vite, on se rendit compte que cette enzyme pouvait exister sous de multiples formes, chacune possédant des propriétés différentes qui dépendent à la fois de leur sélectivité au substrat et de certaines caractéristiques physicochimiques (Omura et *al.*, 1993). A titre d'exemple, des variations distinctes sont relevées autour du pic d'absorption des 450 nm pour des enzymes isolées de sources différentes, ou issues de préparations microsomales provenant d'animaux traités par des composés chimiques différents (Ruckpaul et Rein, 1984). Cette multiplicité en formes de cytochromes P450 n'est en fait qu'une conséquence d'une multiplicité en gènes (Gonzalez, 1988). Elle a ainsi conduit l'enzyme dans un premier temps, à l'appellation d'oxydase à « fonction mixte » puis, plus récemment, à celle de « mono-oxygénase » pour décrire sa capacité à insérer un atome d'oxygène dans une large variété de substrats et classes structurales de composés (Porter et Coon, 1991 ; Okita et Masters, 1992 ; Ortiz de Montellano, 1986, 1995). Encore largement utilisée, l'appellation de « cytochrome » n'est pourtant plus considérée comme la plus appropriée pour décrire ces hémoprotéines : non seulement les P450s diffèrent significativement des autres cytochromes en de nombreux points, ils sont également plus reconnus comme des enzymes dites « hemo-thiolates » que des protéines redox (cas des cytochromes a, b ou c par exemple) en dépit du fait qu'ils possèdent, un potentiel redox, bien plus bas que leurs homologues (de -420 mV).

1.1.3.4 Diversité des réactions catalysées

Une des particularités des CYPs est leur capacité à catalyser une multitude de réaction de monooxgénation, dont l'hydroxylation est la plus répandue. Ils réalisent également des réactions d'époxydation, de déshydratation, d'isomérisation ou de réductions (Mansuy et Battioni, 2000 ; Guengerich, 2001 ; Ortiz de Montellano, 2005) sur des motifs carbonés aliphatiques ou aromatiques en transformant un atome d'oxygène à partir de dioxygène et nécessitant quelques protons et un donneur d'électrons.

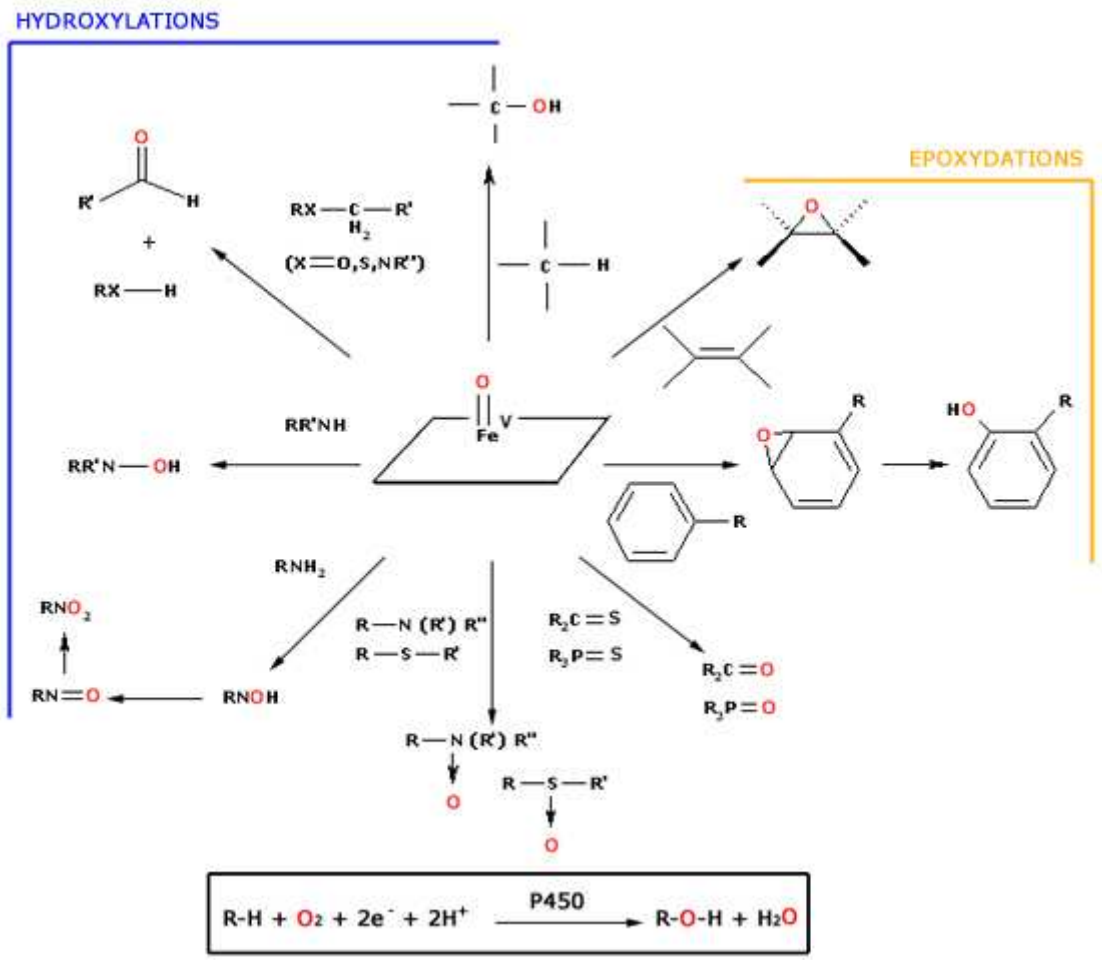


Figure 1-5 Réactions d'oxydation généralement catalysées par les P450s. D'après (Mansuy et Battioni, 2000)

1.1.3.5 Rôles multiples

Cette variété de réactions catalysées confère aux P450s un rôle important aussi bien dans la transformation de molécules endogènes qu'exogènes. Selon les organismes dans lesquels ils sont exprimés, les CYPs ne jouent pas le même rôle. Comme il a été déjà évoqué, les P450s sont les principales enzymes impliquées dans le métabolisme des xénobiotiques. Ils participent entre autre à l'élimination des composés exogènes comme les médicaments, les toxines ou les polluants en complétant l'action défensive du système immunitaire destiné à prendre en charge les macromolécules. Dans certains cas, pourtant, la dégradation de certains xénobiotiques peut former des métabolites réactifs électrophiles et devenir source de toxicité pour le P450 ou d'autres protéines cellulaires (cf. paragraphe 1.6.2). Par ailleurs, les CYPs occupent une place importante dans la biosynthèse et la biodégradation des composés endogènes. Chez les procaryotes, certains CYPs participent à la

biosynthèse de composés endogènes tels que les stérols ou des antibiotiques assurant la défense du microorganisme. Chez les eucaryotes, c'est également les P450s qui assurent la synthèse des stérols nécessaires aux membranes plasmiques comme le cholestérol ou l'ergostérol, ainsi que la biosynthèse des hormones stéroïdiennes, des eïcosanoïdes, des rétinoïdes et de certaines vitamines (Guengueric, 2005).

1.1.3.6 Un large spectre de substrats

Les CYPs sont en mesure de catalyser un nombre très important et varié de substrats : les CYPs participant à des chaînes de biosynthèse endogènes sont généralement spécifiques à un seul substrat, tandis que ceux impliqués dans le métabolisme de xénobiotiques montrent une très faible spécificité de substrat. Ces derniers sont capables de métaboliser des substrats de polarité ou taille très variable. L'exemple qu'on a coutume de citer pour illustrer ce phénomène est celui du CYP3A4, cytochrome humain qui à lui seul est en mesure de métaboliser jusqu'à 50% des médicaments prescrits chez l'homme (Rendic, 2002). Il reconnaît des substrats allant du paracétamol (151 Da) pour le plus petit, à la cyclosporine (1,2 kDa) pour la plus volumineuse. À noter qu'un substrat peut être oxydé de plusieurs façons différentes au sein d'un même P450 et que la même activité d'oxydation d'un substrat peut être catalysée par plusieurs P450s.

1.1.3.7 Un corpus de connaissance croissant

Enfin, outre la diversité en gènes, en formes, en réaction et en rôle, le corpus de connaissance dans ce domaine est lui même profond et étendu : le nombre de publications sur les P450s (ou relatives aux P450s) ne cesse d'augmenter depuis cette dernière décennie, de façon régulière (voir exponentielle) et avoisinant la barre des 1000 publications scientifiques par an (Estabrook, 1996 ; Koymans et *al.*, 1993). La Figure 1-6 montre justement les différents domaines de recherche et autres champs d'application des P450s.

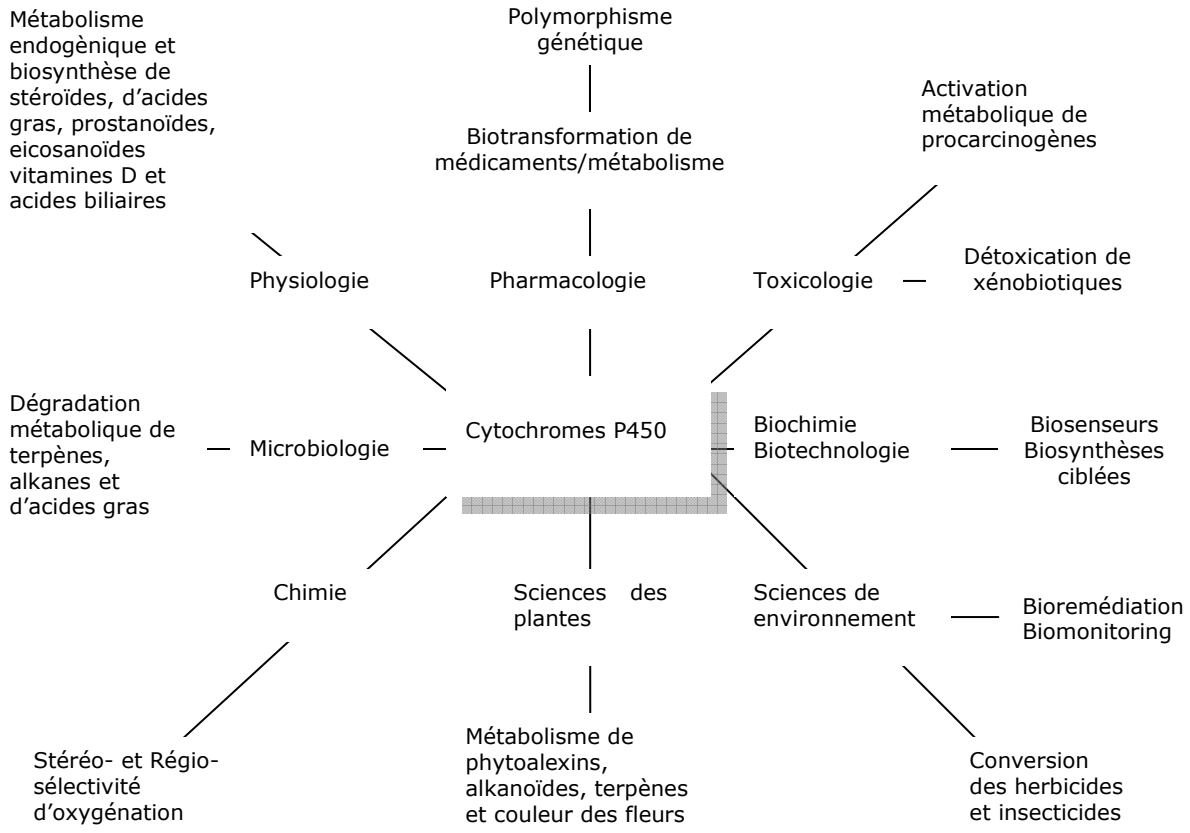


Figure 1-6 Domaines de recherches du P450 et champs d'application (Adapté de Bernhardt, 1996 ; récupéré de Lewis, 2001)

En résumé, il est facile de comprendre toute cette agitation « cérébrale » autour des CYPs, tant leurs fonctions, leurs implications, leurs mécanismes sont importants et variés. Ils présentent non seulement une véritable multiplicité en gènes, une grande diversité en réactions catalytiques mais montrent également des rôles variés, pouvant aller du métabolisme de xénobiotiques à la biosynthèse et biodégradation des composés endogènes. Dans cette diversité pourtant, le mécanisme sous-jacent semble imposer une contrainte à cette superfamille : sa structure très conservée.

1.1.4 Conséquences structurales

Étonnamment et en dépit de leur diversité, les CYPs semblent partager en commun un repliement très conservé. En effet, compte tenu de l'évolution fonctionnelle des CYPs, la structure tertiaire des

P450s telle que nous la connaissons[1] au travers des différents cristaux publiés dans la Protein Data Bank (PDB) (http://www.rcsb.org) résulterait de contraintes apportées par divers éléments nécessaires à sa fonction catalytique (cf. Tableau 1.1) : liaison au substrat, hème, oxygène et partenaire(s) redox.

Tableau 1.1 Rôles structuraux de l'apoprotéine P450 (source Lewis, 2001)

Fonction du P450	***Dispositifs structuraux impliqués***
1. Liaison à l'hème	Hélice I et L, cystéine invariante et usuellement deux résidus basiques pour la liaison ionique avec les propionates de l'hème
2. Liaison et activation de l'oxygène	La région de l'hélice I distale à la partie héminique
3. Liaison des partenaires redox	Résidus basiques pour les liaisons ioniques
4. Transfert du proton à l'oxygène	Liaisons ioniques internes établies par les résidus avoisinant l'hème
5. Liaison des différents substrats	Les hélices B', F et I et les brins β1 et β4 (Région SRS)
6. Régulation de ces activités à l'intérieur de la membrane	Peptide N-terminal de 30-40 résidus de long

SRS = Substrate recognition site (Gotoh, 1992)

Ces contraintes liées à la fonction sembleraient exercer une certaine pression conduisant à la conservation structurale des P450s : en effet, d'un P450 à un autre, la structure des P450s est inchangée, et ce en dépit de leurs localisations environnementales, que ce soit chez la bactérie (où le P450s est cytoplasmique), les mitochondries ou encore dans les systèmes microsomaux (où les P450s sont ancrés à la membrane) (Degtyarenko et Archakov, 1993). Il existe bien entendu quelques exceptions connues à cette conservation structurale. C'est le cas des P450s dont la fonction est quelque peu inhabituelle, comme par exemple l'allène oxyde synhtase (AOS), la prostacyclin synthase (PGIS) ou encore la thromboxane synthase (TXAS). En fait, les différentes fonctionnalités montrées par ces enzymes peuvent être associées à de petits changements dans la séquence (par exemple sur l'hélice I), mais suffisamment cruciaux (cf. Tableau 1.2) à l'intérieur du centre catalytique du P450 pour entraîner une légère modification de la structure de la protéine.

[1] De 16 à 18 hélices α numérotées de A à L avec quelques variations et 4 à 6 feuillets β numérotés de 1 à 6. La structure des CYPs sera vue plus loin.

Tableau 1.2 Comparaison de l'hélice I entre plusieurs P450s.

Espèces	***CYP***	***Abréviation***	***Séquence***										***Type de réaction enzymatique***
						↓	↓		↓	↓			
Humain	3A4	Nif	F	I	F	A	G	Y	E	T	T	S	Nifedipine N-Oxidase
Humain	5A1	TXS	F	L	I	A	G	Y	E	I	I	T	Thromboxane synthase
Rat	5A1	TXS	F	L	I	A	G	H	E	I	T	T	Thromboxane synthase
Souris	5A1	TXS	F	L	I	A	G	H	E	V	I	T	Thromboxane synthase
Cochon	5A1	TXS	F	L	I	A	G	Y	E	I	I	T	Thromboxane synthase
Lin	74	AOS	N	S	W	G	G	F	K	I	L	L	Allène oxyde synthase
Guayule	74	AOS	N	T	F	G	G	V	K	I	L	F	Allène oxyde synthase
Humain	19	Arom	M	L	I	A	A	P	D	T	M	S	Estrogène synthase
Vache	19	Arom	M	L	I	A	A	P	D	T	M	S	Estrogène synthase
Rat	19	Arom	M	L	I	A	A	P	D	T	M	S	Estrogène synthase
Souris	19	Arom	M	L	I	A	A	P	D	T	M	S	Estrogène synthase
Poulet	19	Arom	M	L	I	A	A	P	D	T	L	S	Estrogène synthase
Truite	19	Arom	M	V	I	A	A	P	D	T	L	S	Estrogène synthase
Levure	51	14DM	V	L	M	G	G	Q	Q	T	S	A	Lanosterol 14α-démétylase
Rat	51	14DM	L	L	L	A	G	Q	Q	T	S	S	Lanosterol 14α-démétylase
Humain	51	14DM	L	L	L	A	G	Q	Q	T	S	S	Lanosterol 14α-démétylase
Candida	51	14DM	V	L	M	G	G	Q	Q	T	S	A	Lanosterol 14α-démétylase

Note :

1. La séquence habituelle de l'hélice distale est montrée par la séquence AGxET du CYP3A4 (ou x est n'importe quel résidu
2. CYP5A1 ne possède pas la thréonine distale conservée
3. CYP74 ne possède ni la thréonine distale conservée ni le résidu acide précédent
4. CYP19 possède une proline avant l'aspartate
5. CYP51 ne possède pas de résidu acide avant la thréonine distale
6. ↓. Pointe les résidus normalement conservés

Références : Nelson et *al.*, 1996 ; Mansuy et Renaud, 1995, Tableau tiré de Lewis, 2001

En conclusion, ce qui a été un jour vu en tant qu'un simple enzyme est désormais connu pour sa multiplicité de formes, disposant de fonctionnalités variées et d'un vaste nombre de substrats potentiels (Porter et Coon, 1991 ; Guengerich, 1991a et b ; Stegeman et Livingstone, 1998 ; Mansuy, 1998; Nelson, 1999).

1.2 Nomenclature, Classification et Phylogénie

Vu l'importance en nombre et l'extrême diversité des P450s exprimés dans le monde vivant, sans méthodologie stricte, on risquerait de se perdre dans un brouillard d'informations. Ainsi, il a été convenu de répertorier chaque P450 selon une **nomenclature** particulière. Cette dernière a été arbitrairement choisie, en fonction du degré d'identité entre chaque P450. Un autre regroupement a pu également être décrit, reposant cette fois-ci sur la fonctionnalité des P450s, et plus précisément, la manière dont ces derniers reçoivent les électrons nécessaires à leurs activités. On parlera alors de **classification**. Bien évidemment, ces deux éléments permettent d'apporter suffisamment d'informations pertinentes à la réalisation d'un modèle d'évolution, qui peut être décrit par la **phylogénie**.

1.2.1 Nomenclature

Depuis 1989, une appellation systématique des P450s a été mise en place, périodiquement mise à jour depuis (Nebert et *al.*, 1989a et b, 1991a ; Nelson et *al.*, 1993, 1996). Cette nomenclature utilise le symbole CYP (ou Cyp chez la souris et la drosophile) comme abréviation pour le terme cytochrome P450 (aussi bien ADNc, ARNm que protéine), symbole qu'on retrouve en italique (*CYP*) lorsqu'il fait référence au gène. La lettre P qui est parfois trouvé après un gène informe quant à lui qu'on a affaire à un pseudogène. Il a donc été convenu que les CYPs soient classés en fonction de leur degré de similitude dans leur séquence primaire[2] (séquence en acides aminés) (voir aussi le site internet de David Nelson à l'adresse http://drnelson.utmem.edu/CytochromeP450.html) comme suit :

- Appartiennent à la même **famille** (désigné par un chiffre arabe, CYP**1**, CYP**2**, …), les CYPs ayant un degré de similitude supérieur à 40% ;
- A la même **sous-famille** (représentée par une lettre majuscule, CYP2**B**, CYP2**C**, …), les CYPs ayant un degré de similitude supérieur à 55%
- Enfin, les **isoformes** appartenant à une même sous famille sont différenciés par un chiffre arabe (CYP2C**8**, CYP2C**9**,…).

[2] Il existe des cas particuliers pour les CYP responsable du métabolisme des stéroïdes où l'ancienne nomenclature subsiste : le nombre juste après CYP correspond alors au numéro de l'atome du substrat où s'opère la réaction de monoxygénation. C'est le cas pour le CYP 19 par exemple.

Alors que ce système fut initialement mis en place de façon arbitraire, ce choix de classification par degré de similitude s'est révélé être assez judicieux et approprié, compte tenu qu'il permet de séparer les différentes familles et sous-familles de façon satisfaisante dans la majorité des cas. Bien évidemment, certains regroupements demeurent inexacts.

En se basant sur cette similitude entre les séquences et dans certains cas en tenant compte des activités catalytiques, il a donc été possible de déterminer si deux gènes étaient issus ou non d'une duplication de gènes phylogénétiques. En raison du nombre de familles devenu très important, un niveau supplémentaire de regroupement est requis. Les clans font alors leur apparition et rassemblent les familles qui appartiennent à un même groupe (issu d'un même gène ancestral) d'après de nombreux arbres phylogénétiques établis auparavant (Nelson, 1999). Ces clans sont désignés par le chiffre le plus petit des familles qu'ils regroupent ou par le chiffre de la famille majoritaire. Ainsi, le clan 2 regroupe la famille de CYP2 de même que les familles CYP1, 17, 18 et 21.

Outre cette classification standard, les divers CYP se distinguent aussi par deux autres critères : leur spécificité de substrat et la classe de molécules entraînant l'augmentation de l'expression. Néanmoins, bien que les CYPs puissent se distinguer par leur spécificité au substrat, cette dernière caractéristique peut être chevauchante comme il a été déjà mentionné précédemment : un CYP peut ainsi métaboliser plusieurs substrats et un substrat peut être métabolisé par plusieurs CYPs. On peut donner à titre d'exemple le cas de la morphine qui peut être soit métabolisée par les CYP3A4 et 2C8 (Projean et *al.*, 2003). Un aperçu des CYPs et de leurs substrats est disponible sur le site Internet http://www.edhayes.com/startp450.html.

1.2.2 Classification

Les CYPs catalysant principalement des réactions de mono-oxygénation, un apport d'électron au substrat est donc nécessaire. Ces électrons sont fournis par le NADPH ou le NADH. Ainsi, les CYPs ont été divisés en plusieurs grandes classes selon différents critères, tels que la façon dont les électrons sont transportés du NAD(P)H au site catalytique. A ce jour, deux nomenclatures sont utilisées pour différencier ces CYPs : (a) une ancienne nomenclature comprenant seulement 4 classes, adoptée de tous, et (b) une plus récente décrit par Hannemann en 2006 qui distingue 10 classes. L'intérêt de cette dernière nomenclature est discutable, mais semble être plus rigoureuse pour départager certains regroupements qui hébergent des cas particuliers. Néanmoins, dans ce manuscrit, c'est la classification en 4 classes qui sera utilisée.

1.2.2.1 Classification à 4 classes

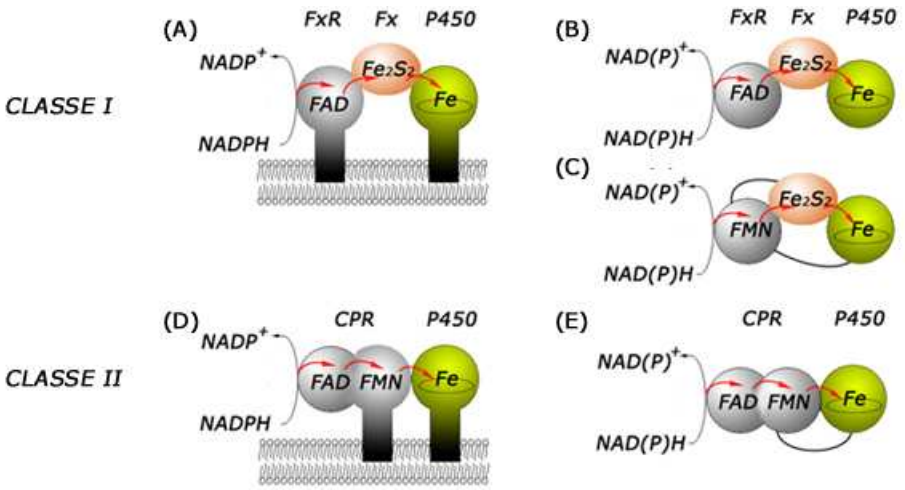


Figure 1-7 Partenaires de transfert d'électrons des CYPs (d'après Paine et *al.*, 2005). La classe I comprend les P450s mitochondriaux (A), la majorité des P450s solubles (B) et certains systèmes fusionnés, dont le CYP RhF est un représentant (C). La classe II comprend les P450s du RE (D) et les P450s qui incluent la ferredoxine à deux flavines comme P450BM3 (E) Fx : ferredoxine, FxR : ferredoxine réductase, CPR : CYP réductase

Dans la classification originale, on distingue deux classes majoritaires de transfert d'électrons (Paine et *al.*, 2005) (cf. Figure 1-7). Deux autres classes sont également rencontrées, mais beaucoup plus rarement.

- La **classe I** fait intervenir dans la chaîne de transfert d'électrons des enzymes à centre fer-soufre. Les électrons du NAD(P)H sont prélevés par une ferrodoxine réductase à flavine (FAD) puis transmis à une ferrodoxine à centre fer-soufre Fe_2S_2. Cette ferrodoxine transfère ensuite les électrons au P450 pour permettre la catalyse. Cette classe contient la très grande majorité des P450s bactériens et les P450s mitochondriaux. Il conviendrait d'ajouter à cette classe des nouveaux systèmes identifiés récemment comme le P450RhF (Roberts et *al.*, 2002), qui est une protéine de fusion contenant une réductase à flavine et FMN, une protéine type ferredoxine à centre fer-soufre et le cytochrome P450.
- La **classe II** des P450s est la plus commune chez les eucaryotes, à l'origine de réactions catalytiques extrêmement diverses et variés. Les CYPs de cette classe sont trouvés au niveau du RE et requièrent une NADPH-Cytochrome P450 réductase (CPR) qui contient les groupements prosthétiques FAD et FMN. Cette dernière est issue d'une fusion de deux protéines ancestrales : elle montre une partie N-terminale homologue avec une flavodoxine bactérienne à FMN et une partie C-terminale homologue avec une ferrodoxine NADP+

réductase à FAD et avec une NADPH-cytochrome b5 réductase. Dans le système microsomal de P450s, en plus de la CPR précédemment décrite, on retrouve aussi un autre système qui n'est pas spécifique des P450s, faisant intervenir un cytochrome b5. Ce dernier est en mesure de transférer lui aussi des électrons au P450, après avoir reçu ces électrons soit par la CPR soit par la NADH-cytochrome b5 réductase. Dans la nomenclature à 4 classes, le $P450_{BM3}$ issu de *Bascillus megaterium* et la NO-synthase apparaissent toutes deux dans cette classe, ce qui n'est pas le cas dans la nomenclature à 10 classes.

- Le $P450_{nor}$ catalyse la réduction du monoxyde d'azote NO et reçoit ses électrons directement du NADH, sans protéine intermédiaire (Takaya et *al.*, 1999). Ce P450 est le seul représentant de la **classe IV**.
- D'autres P450 catalysent des isomérisations ou des déshydratations, et ne nécessitent pas d'apport d'électrons extérieurs ou d'oxygène moléculaire. Les substrats transformés sont généralement riches en électrons, comme des hydroperoxydes ou des endoperoxydes. Bien qu'aucun système de transfert d'électrons ne soit présent, la **classe III** est tout de même défini et comprend ces systèmes. La récente $P450_{PCIS}$ humaine (qui est une isomérase) fait partie de cette classe (Strushkevich et *al.*, en attente de publication).

1.2.2.2 Classification à 10 classes

La classification précédemment décrite est la plus connue, et admise de toute la communauté scientifique. Néanmoins, il existe des cas pour lesquels le regroupement tel qu'il a été décrit ne semble pas des plus appropriés, notamment pour des CYPs bactériens. Ceux-ci ont amené Hannemann à proposer une autre classification (Hannemann et *al.*, 2006), toujours basée sur le transfert d'électrons, mais plus adaptée pour ces cas particuliers. Les principaux changements observés, outre le nombre de classes, est principalement la sortie de certains de ces P450s de procaryotes d'une certaine classe (des classes I et II) pour former une nouvelle classe à eux seuls.

Le Tableau 1.3 et la Figure 1-8 qui suivent, résument et illustrent les principales caractéristiques permettant de regrouper les P450s selon cette nouvelle classification.

Tableau 1.3 Classes des systèmes P450s, classées selon la topologie des partenaires redox impliqués dans le transfert d'électrons au CYP (d'après Hannermann, 2006).

Classe / Source	***Chaîne de transport d'électron***	***Localisation / Remarques***
Classe I Bactérienne Mitochondriale	NAD(P)H > [FdR] > [Fdx][a]> [P450] NADPH > [FdR] > [Fdx] > [P450]	Cytosolique, soluble P450 : membrane interne des mitochondries FdR : membrane associée Fdx : matrice mitochondriale, soluble
Classe II Bactérienne Microsomale A Microsomale B Microsomale C	NADH > [CPR] > [P450] NADPH > [CPR] > [P450] NADH > [CPR] > [cytb5] > [P450] NADH > [cytb5Red] > [cytb5] > [P450]	Cytosolique, soluble ; *Streptomyces carbophilus* Ancrée à la membrane, ER Ancrée à la membrane, ER Ancrée à la membrane, ER
Classe III Bactérienne	NAD(P)H > [FdR] > [Fldx] > [P450]	Cytosolique, soluble ; *Citrobacter braakii* Similaire à la classe II, mais sans l'entité CPR
Classe IV Bactérienne	Pyruvate, CoA > [OFOR] > [Fdx] > [P450]	Cytosolique, soluble ; *Sulfolobus tokadaii* Classe des P450s thermophiles (ex CYP 119)
Classe V Bactérienne	NADPH > [FdR] > [Fdx–P450]	Cytosolique, soluble ; *Methylococcus capsulatarus*
Classe VI Bactérienne	NAD(P)H > [FdR] > [Fldx–P450]	Cytosolique, soluble ; *Rhodococcus rhodochrous* strain 11Y
Classe VII Bactérienne	NADH > [PFOR–P450]	Cytosolique, soluble ; *Rhodococcus sp.* strain NCIMB 9784, *Burkholderia sp.*, *Ralstonia metallidurans* Cas du P450RhF
Classe VIII Bactérienne, fungique	NADPH > [CPR–P450]	Cytosolique, soluble ; *Bacillus megaterium, Fusarium oxysporum* Cas du $P450_{BM3}$
Classe IX Seule NADH dépendant, fungique	NADH > [P450]	Cytosolique, soluble ; *Fusarium oxysporum* Seul représentant : P450nor
Classe X Indépendant chez la plante/mammifère	[P450]	Liée à la membrane, ER Anciennement la classe III

Abréviation pour les partenaires contenant les centres redox suivants : Fdx (complexe fer-sulfure) ; FdR, Ferrodoxine réductase (FAD); CPR, cytochrome P450 réductase (FAD, FMN) ; Fldx, Flavodoxine (FMN) ; OFOR, 2-oxyacid:ferrodoxine oxydoréductase (complexe thiamine phosphate, [4Fe-4S]) ; PFOR, phthalate-family oxygénase réductase (FMN, complexe [2Fe-2S]).

[a] Fdx contenant un complexe fer-sulfure de type [2Fe-2S], [3Fe-4S], [4Fe-4S], [3Fe-4S]/ [4Fe-4S]

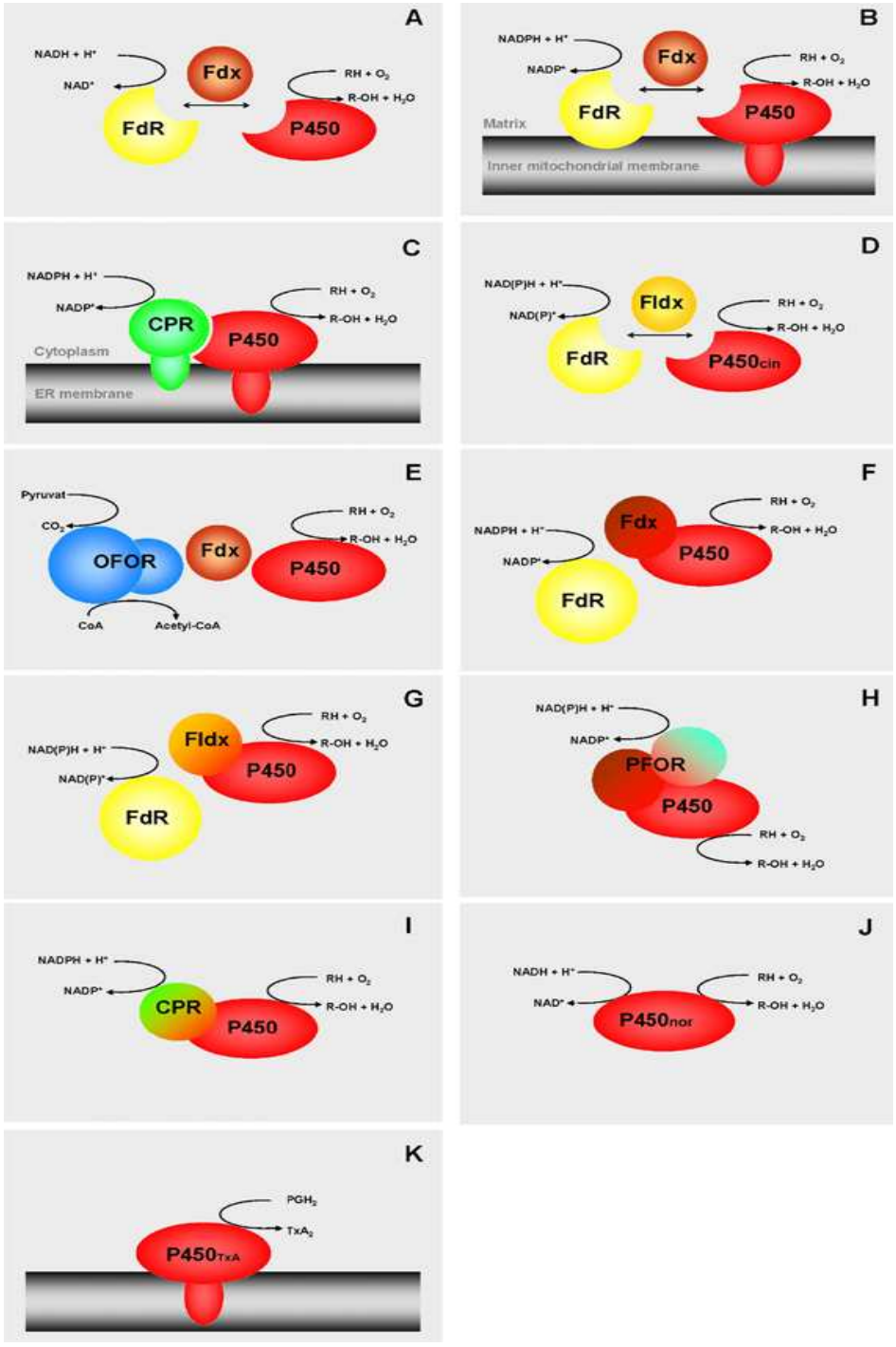


Figure 1-8 Organisation schématique des différents systèmes de P450s. (A) Classe I, système bactérien ; (B) Classe I, système mitochondrial ; (C) Classe II, système microsomal ; (D) Classe III, système bactérien, ex. du P450cin ; (E) Classe IV, système bactérien thermophile ; (F) Classe V, système bactérien à [Fdx]–[P450] fusionné ; (G) Classe VI, système bactérien à [Fldx]–[P450] fusionné ; (H) Classe VII, système bactérien à [PFOR]–[P450] fusionné ; (I) Classe VIII, système bactérien à [CPR]–[P450] fusionné ; (J) Classe IX, système eucaryote P450nor ; (K) Classe X, système eucaryote indépendant, cas du P450TxA (source Hannemann et *al.*, 2006)

1.2.3 Phylogénie

Pour remonter aux origines des P450s, les chercheurs ont essayé de retracer leur évolution au moyen de techniques phylogéniques. Cependant, la construction d'arbres phylogénétiques n'est pas aisée pour cet enzyme car elle se base sur des matrices de distances calculées à partir d'un alignement multiple en séquences primaires (MSA). Cet alignement dépend fortement du degré de similarité entre séquences utilisées. En raison des divergences en séquence primaire, il est facile de comprendre qu'un alignement multiple qui prendrait en compte tous les P450s serait difficilement réalisable. La classification des P450s en famille et sous famille, basée sur la similitude en séquence, a rendu plus abordable dans un premier temps, la construction d'arbres phylogénétiques ainsi que d'autres formes d'analyses qui étudient l'évolution des relations entre gènes de P450s.

Un nombre très varié de méthodes a été utilisé pour comparer les séquences de P450s conduisant la plupart du temps à des arbres assez proches pour un gène ancestral commun à 2 milliards d'années. La méthode largement utilisée pour investir l'évolution et les relations entre P450s est celle de l'UPGMA (Unweighted Pair Group Method of Analysis) (Nelson et Strobel, 1987 ; Gotoh et Fujii-Kuriyama, 1989 ; Nebert et *al.*, 1991 ; Nebert et Gonzalez, 1987). Nebert et Nelson ont également exploré les variations au sein des arbres phylogénétiques produits par le Neighbor Joining (NJ) et l'UPGMA pour 39 séquences, où ils trouvèrent une légère différence entre les deux méthodes principalement en raison des familles de CYP1 et CYP2. Toutes les méthodes d'analyse de séquences ont montré que la forme bactérienne CYP102 ($P450_{BM3}$) du *Bascillus megaterium* se regroupe avec les P450s d'eucaryotes, tandis que les autres formes de procaryotes ségrégent clairement dans des classes différentes. Une des raisons évoquées serait liée à la similitude des partenaires redox partagés par CYP102 et les P450s microsomaux d'eucaryotes comme il a été vu précédemment lors des classifications à 4 formes.

Au final, un arbre simplifié (cf. Figure 1-9) a été proposé où partant d'un gène ancestral commun (il y a 2 milliards d'années) qui se serait dupliqué. Les gènes ainsi créés auraient alors divergé, faisant apparaître progressivement une multitude de familles, sous-familles et d'isoformes de CYP. Au cours de l'évolution, ces gènes qui se sont considérablement diversifiés, ont vraisemblablement participé à l'adaptation des organismes aux changements de biotope. Cette diversification est corrélée au passage d'une atmosphère originellement réductrice à une atmosphère plus oxydante sous l'effet de la production de dioxygène par les organismes photosynthétiques. La pression actuelle en oxygène a

ainsi crû rapidement de façon exponentielle avec le temps. Ceci serait une cause probable de la non linéarité de l'évolution des CYPs.

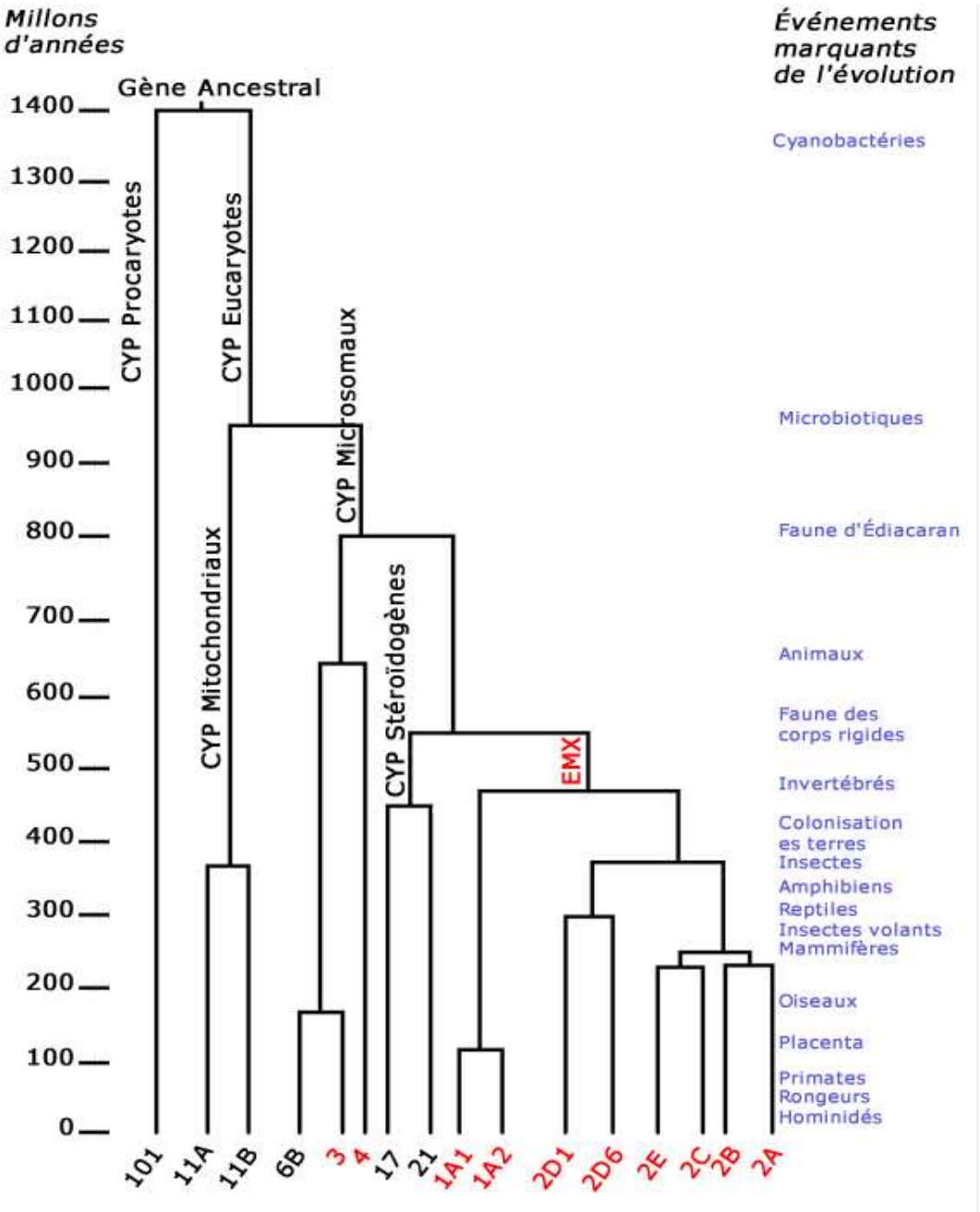


Figure 1-9 Arbre phylogénique simplifié des cytochromes P450 (d'après Lewis, 2001). Les CYPs (CYP1 à 4) qui interviennent dans le métabolisme des xénobiotiques sont annotés en rouge (où EMX signifie Enzyme du métabolisme des Xénobiotiques). Il est admis que ces familles CYPs ont évolué puis divergé vers le monde animal pour permettre une adaptation et/ou une protection vis-à-vis de composés nouveaux et/ou toxiques synthétisés par les végétaux. Les CYP11A, 11B, 17,19 et 21 interviennent dans la synthèse d'hormones stéroïdiennes en catalysant des réactions d'hydroxylations.

1.3 Occurrence, Distribution et Localisation

1.3.1 Occurrence et Distribution

Comme mentionné précédemment, les CYPs sont des enzymes ubiquitaires, largement représentées dans tous les organismes vivants, de la bactérie aux mammifères, en passant par la plante et les champignons (Nerbert et *al.*, 1989a et b), bien qu'il semble que certaines espèces primitives de bactéries en soient dépourvues. La répartition de ces CYPs au sein des différents organismes est consultable sur le site (http://drnelson.utmem.edu/CytochromeP450.html) de David Nelson, site complet qui est tenu régulièrement à jour par l'auteur de façon bénévole depuis plus de 10 ans. On y retrouve donc non seulement la répartition entre organismes et les statistiques sur les P450s, mais également tout un ensemble d'informations relatives aux P450s. Il permet ainsi à n'importe quel utilisateur d'avoir accès aux connaissances actuellement disponibles sur les P450s telles que les bases de données ou encore les séquences de P450s. Concernant la répartition, à la date du 7 septembre 2007, Nelson a répertorié pas moins de 7703 séquences de CYPs provenant de 866 familles différentes qui se répartissent entre les animaux, les plantes, les champignons, les bactéries, les protistes et les archées. Ces données sont issues d'un comptage et d'une vérification manuelle par l'auteur du site (cf. Tableau 1.4 et Figure 1-10). Elles mettent d'ailleurs en évidence une proportion très élevée de CYPs chez la plante, phénomène qui s'explique en partie par le fait que de nombreux génomes ont été séquencés chez la plante.

Tableau 1.4 Répartitions des CYPs dans les différents organismes vivants (source de Nelson)

Organismes	***Nombre de CYPs compté***	***Nombre de familles de CYPs***
Animaux	2740	109
Plantes	2675	94
Champignons	1231	309
Bactéries	813	290
Protistes	226	54
Archées	18	10
Total	7703	866

Les statistiques données ici ont été enregistrées le 7 septembre 2007.

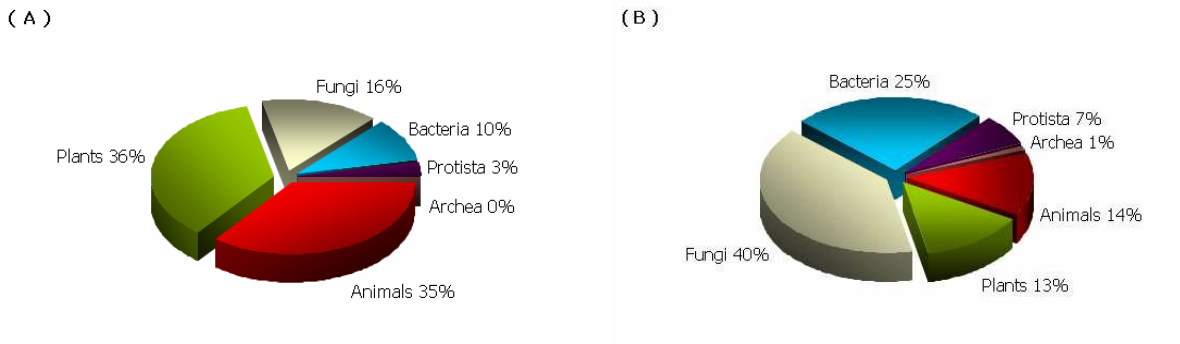


Figure 1-10 Répartition des séquences (a) et des familles (b) de CYPs dans les différents organismes vivants. On remarque que les CYPs sont largement représentés dans chez la plante et l'animal. Cela est lié à l'effort des investigations dans chaque organisme.

Même si les P450s ont été caractérisés dans de nombreuses espèces telles que les oiseaux, les poissons, les reptiles, les insectes, les mollusques, les arthropodes, les crustacés, les champignons, les plantes et les bactéries (Schenkman et Griem, 1993 ; Pinot et *al.*, 1999 ; Gorman et *al.*, 1998 ; Chapple, 1998 ; Scott et *al.*, 1998 ; Walker, 1998 ; Hallahan et *al.*, 1993 ; Hallahan et West, 1995 ; Nelson, 1998) (Une revue spéciale de « *Comparative Biochemestry and Physiology* » (Volume 121C, Nos. 1–3, 1998) traite justement de ces P450s non mammifères), c'est pourtant chez les mammifères que le P450 suscitent le plus d'intérêt, en particulier les formes humaines de l'enzyme (Guenguerich, 1989a et b, 1992a–d, 1994, 1995a ; Guenguerich et *al.*, 1998, 1992 ; Rendic et DiCarlo, 1997). En conséquence, les P450s humains ont été intensément investis et étudiés, et les systèmes d'expression hétérologues sont devenus à présent un moyen fiable d'analyse pour les interactions enzyme–substrat de P450s isolés (Estabrook et *al.* 1991 ; Gonzalez et Korzerkwa, 1995).

1.3.2 Localisation

Chez les mammifères, les P450s sont présents dans la plupart des tissus, mais ils sont particulièrement abondants en quantité et en diversité au niveau du foie (cf. Figure 1-11). Pour l'être humain, le foie est un organe très important tant au niveau de sa corpulence physique que son rôle fonctionnel. C'est le premier organe, après l'intestin grêle, à être en contact avec les xénobiotiques *via* le système porte hépatique. Il représente en effet à lui seul 1/50^{e} du poids total du corps humain. Il se compose pour 70% de cellules parenchymateuse ou hépatocytes et pour 30% de cellules diverses dont les cellules de Ito, les cellules composant les canicules biliaires et les cellules de Küpffer. Ses fonctions sont nombreuses et variées : fondamentalement, le foie est impliqué dans la sécrétion et l'évacuation de la bile par les canalicules biliaires, mais du fait qu'il soit extrêmement vascularisé et qu'il soit situé à l'interface de l'appareil digestif et l'appareil circulatoire, il joue également un rôle

dans la distribution de nombreux métabolites issus de la nutrition. En outre, le foie est capable d'assurer un grand nombre de fonctions métaboliques comme la synthèse des protéines plasmatiques, des lipides (dont le cholestérol), de l'urée et le stockage et la libération du glucose dans le sang, la production de substrats à haut potentiel énergétique, mais aussi le stockage de nombreuses vitamines. Enfin, il joue un rôle central au cours du processus de détoxication.

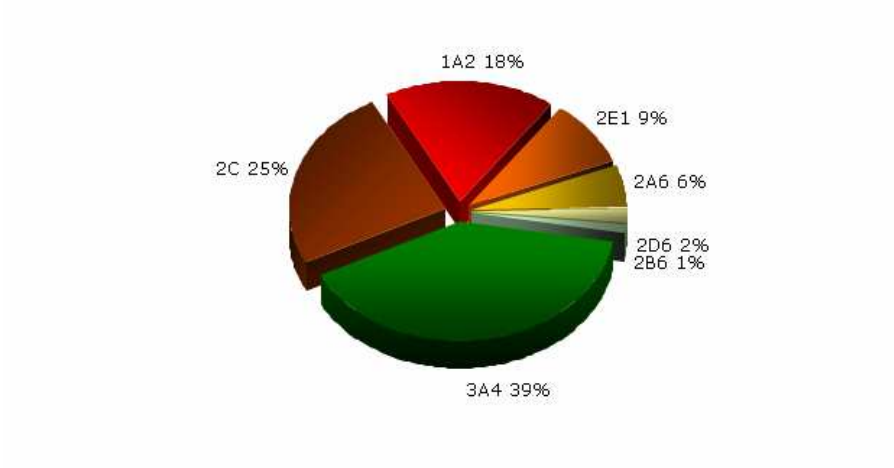


Figure 1-11 Répartition des CYPs dans le foie Humain (d'après Guengerich, 2005)

Ces deux dernières (fonctions de synthèse et de détoxication par le foie) ne sont en fait que le reflet macroscopique des acteurs principaux de l'échelle microscopique : les CYPs. Les CYPs sont donc en nombre abondant dans le foie, et plus précisément localisés dans les membranes du réticulum endoplasmique (RE) des hépatocytes (Ruckpaul et Rein, 1984 ; Stier, 1976). Le RE est en effet une structure lipidique particulière, très favorable aux échanges, et disposant d'une surface étonnamment large (comprenant 7 à 11 m²/g du poids du foie) comparé à son volume, ce qui fait de lui la structure idéale pour héberger les enzymes de Phase I comme les P450s (Lewis et Pratt, 1998 ; Gibson et Skett, 1994). Sachant que 12 à 15% du RE est composé de cytochromes P450, on peut estimer approximativement que les P450s comptent pour un peu moins d'1% du poids total d'un hépatocyte (Ruckpaul et Rein, 1984).

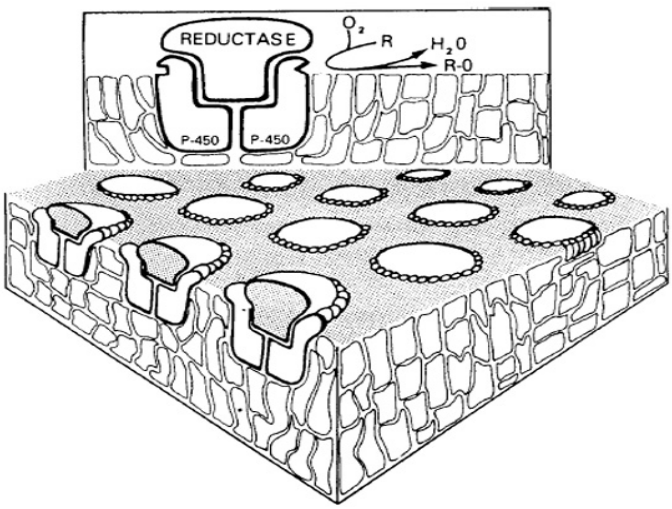


Figure 1-12 Localisation dans le réticulum endoplasmique lisse (source Nebert, 1987). Six à huit P450s se regroupe autour d'une réductase qui leur apporte les électrons nécessaires au métabolisme des substrats.

Au sein de la membrane du RE, des études stœchiométrique (Finch et Stier, 1991 ; Schwarz, 1991) ont suggéré que les P450s se regrouperaient par 6 ou par huit autour d'une réductase, un partenaire fournisseur en électrons. Ces unités hexamèriques (ou octamèriques) subissent alors une diffusion transversale à l'intérieur de cette membrane phospholipidique du RE, elle-même sujette à une diffusion latérale au moyen du « flip-flop ». Par conséquent, les substrats relativement lipophiles se retrouvent alors facilement acheminés vers le centre réactionnel des P450s afin d'y être métabolisés.

Bien que majoritairement concentrés au niveau du foie, les P450s sont également exprimés dans d'autres organes et tissus (cf. Tableau 1.5) tels que le rein, les seins, la prostate (Williams et *al.*, 2000a) la peau, l'épithélium nasal, le placenta, le cerveau, le poumon, la rate, le pancréas et la région gastro-intestinale (Waterman, 1992 ; Schenkmann et Griem, 1993 ; Hellmold et *al.*, 1998 ; Hakkola et *al.*, 1998, 1996). Toutefois, même si les P450s sont identifiés dans ces autres organes, ils demeurent en concentration faible et leur participation dans les biotransformations des xénobiotiques n'est pas bien définie. Certains CYP (CYP11A et CYP11B) sont également présents du côté du feuillet externe de la membrane nucléaire et dans la membrane interne des mitochondries (Montoliu et *al.*, 1995). Ces isoformes sont relativement peu impliquées dans la biotransformation des xénobiotiques mais elles jouent un rôle important dans le métabolisme des composés endogènes comme les stéroïdes, le cholestérol, la vitamine D3, et les acides biliaires.

Tableau 1.5 Expression tissulaire et cellulaire des CYPs

Tissus	***Types cellulaires***
Foie	Hépatocytes (surtout centrobulaires)
	Cellules de Kupffer
Intestin	Entérocytes
Poumon	Cellules de Clara
	Pneumocytes II
	Cellules endothéliales
Rein	Cellules tubulaires proximales
Cerveau	Cellules endothéliales cérébrales
	Cellules du plexus choroïdes
Peau	Kératocytes
Autres :	
Muqueuse Buccale	Cellules épithéliales
Cœur	Myocytes
Pancréas	Cellules pancréatiques
Sang	Leucocytes
Ovaire	Cellules Lutéiniques
Testicule	Cellules de Leydig

En raison de leur diversité, tant au niveau de leur localisation, de leur sélectivité et réactivités vis-à-vis des substrats dans des espèces différentes (Guengerich, 1997), on comprend mieux pourquoi il est relativement difficile d'étudier cette superfamille. Parfois, au sein même des mammifères, des différences de comportements, d'inductibilités et de sélectivités sont observées (Lewis et *al.*, 1998a ; Soucek et Gut, 1992 ; Nedelcheva et Gut, 1994 ; Smith, 1991 ; Guengerich, 1997). En conséquence, il peut être parfois très difficile d'extrapoler des résultats obtenus à partir d'expériences menées sur des rongeurs de laboratoire, par exemple, sur les résultats attendus de métabolisme de composé chez l'Homme (Lewis et *al.*, 1998a). On sait par ailleurs qu'il existe chez les mammifères une différence sexuelle de P450s (Schenkman et *al.*, 1967) et que les niveaux de P450s chez les individus varient en fonction de leur régime alimentaire (Ioannides, 1999 ; Parke et Ioannides, 1994). Il existe enfin une variation de facteurs cellulaires contrôlant l'expression de ces P450s dans les tissus, qui conduit à une altération du niveau de ces enzymes dans les différents tissus (Wolff et Strecker, 1992). Chez l'homme, cette variation se retrouve au niveau hépatique, en raison des différences génétiques liées aux groupes ethnographiques et des changements phénotypiques (Price-Evans, 1993 ; Bachmann, 1996), dont certains ont été associés à la susceptibilité au cancer (Crespi et *al.*, 1991 ; Nebert et *al.*, 1996 ; Crofts et *al.*, 1993 ; Puga et *al.*, 1997).

1.4 Un repliement très conservé

1.4.1 De la bactérie aux mammifères : les premières structures X

La grande diversité des CYPs, tant au point de vue de sa composition en acides aminés, de ses formes dans les différents organismes, que des différents substrats qu'ils sont en mesure de prendre en charge, a été largement soulignée tout au long des précédents paragraphes. En dépit de cette diversité importante et contre toute attente, tous les CYPs partagent une structure tertiaire, un repliement, extrêmement conservé, de la bactérie, aux mammifères. Ce phénomène est d'autant plus remarquable que les CYPs bactériennes et les CYPs des eucaryotes présentent une différence majeure liée à leur localisation : à l'opposé de son homologue bactérien qui est cytoplasmique, les CYPs des eucaryotes sont membranaires, ancrés à la membrane au moyen d'une courte séquence hydrophobe transmembranaire de 20 à 30 acides aminés en N-terminale de la protéine.

C'est d'ailleurs en raison de leur solubilité que les premières structures de CYP à avoir été publiées furent celles de bactéries. Celles d'eucaryotes sont en effet nettement plus difficiles à purifier et à stabiliser au cours de leur extraction, les rendant moins favorable à une expérience de cristallographie. Or, jusqu'à présent, les expériences de **cristallographie par rayon X** (RX) produisent les modèles les plus fiables pour décrire la conformation tridimensionnelle de protéine – en raison de leur taille trop importante, il n'est pas possible d'appliquer la méthode de **résonance magnétique nucléaire** (ou RMN) sur les CYPs – dans un état solide (ces deux techniques seront décrites en annexes). Ce qu'il faut retenir concernant la cristallographie, c'est que le cristal X d'une structure n'est qu'un « instantané » (en anglais on utilisera le terme de « snapshot ») moléculaire de la protéine. Elle correspond à un « cliché photographique » d'une situation dynamique du fonctionnement de la protéine (où par exemple un ligand se lie à l'enzyme, ou au cours de processus de métabolisation…). Même si un cristal donne beaucoup d'informations sur le repliement d'une protéine, il ne tient pas en compte la protéine dans son vrai milieu et dans son fonctionnement habituel. De plus, les contraintes de préparations pour une cristallographie sont nombreuses et il n'est pas possible à ce jour de cristalliser une membrane lipidique ou un élément ancré dans une membrane lipidique.

Ainsi, il y a encore moins d'une dizaine d'années, aucune structure de mammifère n'était publiée : toutes les études et les hypothèses sur la structure et sur la fonction des P450s de mammifère reposaient alors sur des structures connues de P450s bactériennes (ou micro-organismes). Jusqu'au

début des années 90, le seul cristal de P450 disponible est celui du P450cam (CYP101) provenant de *Pseudomonas putida* (Poulos et *al.*, 1985). Cette structure haute résolution est restée longtemps un paradigme pour l'étude de la relation structure-fonction de P450 pendant plusieurs années, jusqu'à l'obtention d'autres structures de P450s solubles, dont les principales sont répertoriées dans le Tableau 1.6. Parmi elles, les quatre premières structures qui ont vu le jour après $P450_{cam}$, sont le $P450_{BM3}$ (CYP102), le $P450_{terp}$ (CYP108), le $P450_{eryF}$ (CYP107) et le $P450_{nor}$ (CYP55). Ce n'est qu'au cours de ces 7 dernières années qu'on assiste à un progrès considérable dans la résolution de structure de CYPs de mammifères avec en 2000 le premier cristal de CYP2C5 un P450 de lapin (Williams et *al.*, 2000c). Celle-ci a été rendue possible au moyen de procédés particuliers qui ont consisté en la protection de la partie C-terminale du P450 par un tag d'histidine et le remplacement de l'ancre hydrophobe N-terminale par une chaîne hydrophile (une séquence MAKKTSSKGR dans le cas du CYP2C9 (Williams, 2003)) pour rendre les enzymes plus solubles. Il est généralement accepté que ces changements opérés sur les CYPs de mammifères n'affectent en rien, ni leur repliement, ni leurs fonctions dans la mesure où ils ne changent pas les paramètres d'activité ou de cinétique de l'enzyme (Cosme et Johnson, 2000 ; von Wachenfeldt et *al.*, 1997 ; Pernecky et *al.*, 1993). A partir du moment où l'on a compris comment cristalliser des structures de mammifère, d'autres P450s de mammifère ont été résolus, tous en majorité de la famille 2, excepté les CYP3A4, CYP 8A1 et CYP1A2.

Entre la première structure de P450 soluble publiée en 1985 (Poulos et *al.*, 1985) et la structure du CYP2D6 humain résolue fin 2005 (Rowland et *al.*, 2005) environs 140 structures tridimensionnelles résolues par cristallographie ont été déposées à la PDB. À ce jour, on peut en répertorier environs 160, bien d'autres sont déposées sur la PDB et attendent d'être publiées. Les progrès technologiques et les moyens mis en œuvre pour l'étude de cette superfamille sont tels qu'on s'attend en moyenne à avoir une à deux structures X par mois. La Figure 1-13 et la Figure 1-14 qui correspondent aux statistiques de dépôt des structures de P450s sur la PDB illustrent justement ce phénomène. A noter que sur les 150 structures qu'on répertorie à ce jour, on peut identifier 29 formes différentes comprenant 19 P450s solubles (dont 1 fongique) et 10 P450s microsomaux de mammifères (dont 8 humaines).

Tableau 1.6 Principales structures tridimensionnelles des P450s déposées à la PDB

CYP	***Organisme***	***Entrées PDB***	***Res (Å)***	***Références (de la 1ere entrée)***
Microorganismes				
CYP 101 ($P450_{cam}$)	*Pseudomonas putida*	56 structures dont 2CPP	1,63	(Poulos et *al.*, 1987)
CYP 102 ($P450_{BM3}$)	*Bacillus megaterium*	21 structures dont 2HPD	2,00	(Ravicahndran et *al.*, 1993)
$P450_{Terp}$	*Pseudomas sp.*	1CPT	2,30	(Hasemann et *al.*, 1994)
CYP 107 ($P450_{eryF}$)	*Saccharopolyspora erythrea*	9 structures dont 1OXA	2,35	(Cupp-Vickery et Poulos., 1995)
CYP 55A1 ($P450_{nor}$)	*Fusarium oxysporum (fungique)*	12 structures dont 1ROM	2,00	(Park et *al.*, 1997)
CYP 119	*Sulfobus solfactaricus*	5 structures dont 1F4T	1,93	(Yano et *al.*, 2000)
CYP 51	*Myobacterium tuberculosis*	5 structures dont 1E9X	2,10	(Podust et *al.*, 2001)
$P450_{OxyB}$	*Amycolatopsis orientalis*	3 structures dont 1LFK	1,70	(Zerb et *al.*, 2002)
$P450_{Epok}$	*Polyangium cellulosum*	5 structures dont 1Q5E	2,65	(Nagano et *al.*, 2003)
CYP 121	*Myobacterium tuberculosis*	4 structures dont 1N40	1,06	(Leys et *al.*, 2003)
$P450_{OxyC}$	*Amycolatopsis orientalis*	1UED	1,90	(Pylypenko et *al.*, 2003)
CYP 152A1	*Bacillus subtilis*	1IZO	2,10	(Lee et *al.*, 2003)
CYP 175A1	*Thermus thermophilus*	2 structures dont 1N97	1,80	(Yano et *al.*, 2003)
CYP 154C1	*Streptomyces coelicolor*	1GWI	1,92	(Podust et *al.*, 2003)
CYP 154A1	*Streptomyces coelicolor*	1ODO	1,85	(Podust et *al.*, 2004)
$P450_{st}$	*Sulfobus tokodaii*	IUE8	3,00	(Oku et *al.*, 2004)
CYP 158A2	*Streptomyces coelicolor A3(2)*	5 structures dont 1S1F	1,50	(Zhao et *al.*, 2005)
$P450_{PKIC}$	*Srteptomyces venezuelae*	3 structures dont 2C7X	1,75	(Sherman et *al.*, 2006)
CYP 199A2	*Rhodopseudomonas palustris*	2FR7	2,01	(Xu et *al.*, en attente)
Mammifères				
CYP 2C5	*Oryctolagus cuniculus*	3 structures dont 1DT6	3,00	(Williams et *al.*, 2000)
CYP 2C9	*Homo sapiens*	3 structures dont 1OG5	2,55	(Williams et *al.*, 2003)
CYP 2C8	*Homo sapiens*	1PQ2	2,70	(Schoch et *al.*, 2004)
CYP 2B4	*Oyctolagus cuniculus*	3 structures dont 1SUO	1,90	(Scott et *al.*, 2004)
CYP 3A4	*Homo sapiens*	6 structures dont 1TQN	2,05	(Yano et *al.*, 2004)
CYP 2A6	*Homo sapiens*	6 structures dont 1Z10	1,90	(Yano et *al.*, 2005)
CYP 2D6	*Homo sapiens*	2F9Q	3,00	(Rowland et *al.*, 2006)
CYP 8A1	*Homo sapiens*	2IAG	2,15	(Chiang et *al.*, 2006)
CYP 2R1	*Homo sapiens*	2OJD	2,70	(Strushkevich et *al.*, en attente)
CYP 1A2	*Homo sapiens*	2HI4	1.95	(Sansen et *al.*, en attente)

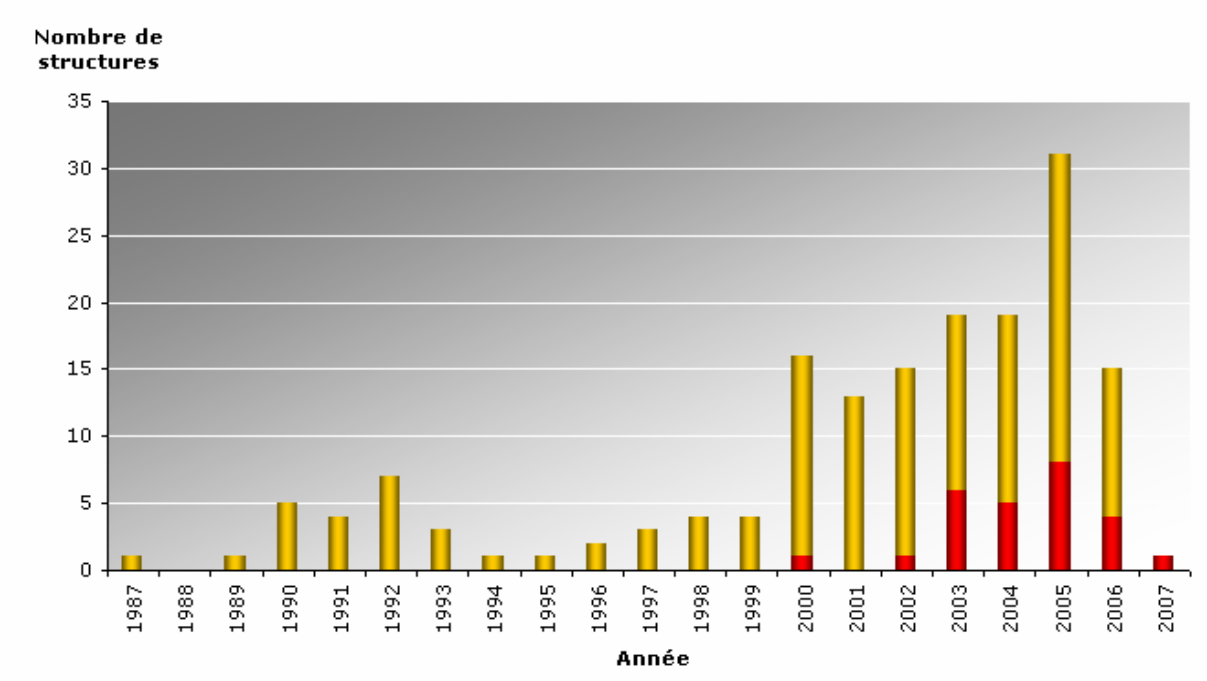


Figure 1-13 Statistiques sur le nombre de structures de P450s publiées dans la PDB. La partie de l'histogramme en jaune représente le nombre de structures bactériennes de P450s publiées dans la PDB, en rouge le nombre de structures de mammifères. La première structure de 1985 du P450cam par l'équipe de Poulos n'est plus disponible (PDB 1CPP). Les structures de mammifères commencent à apparaître en nombre depuis 2000, bien que généralement, ce sont les structures solubles qui soient les plus publiées.

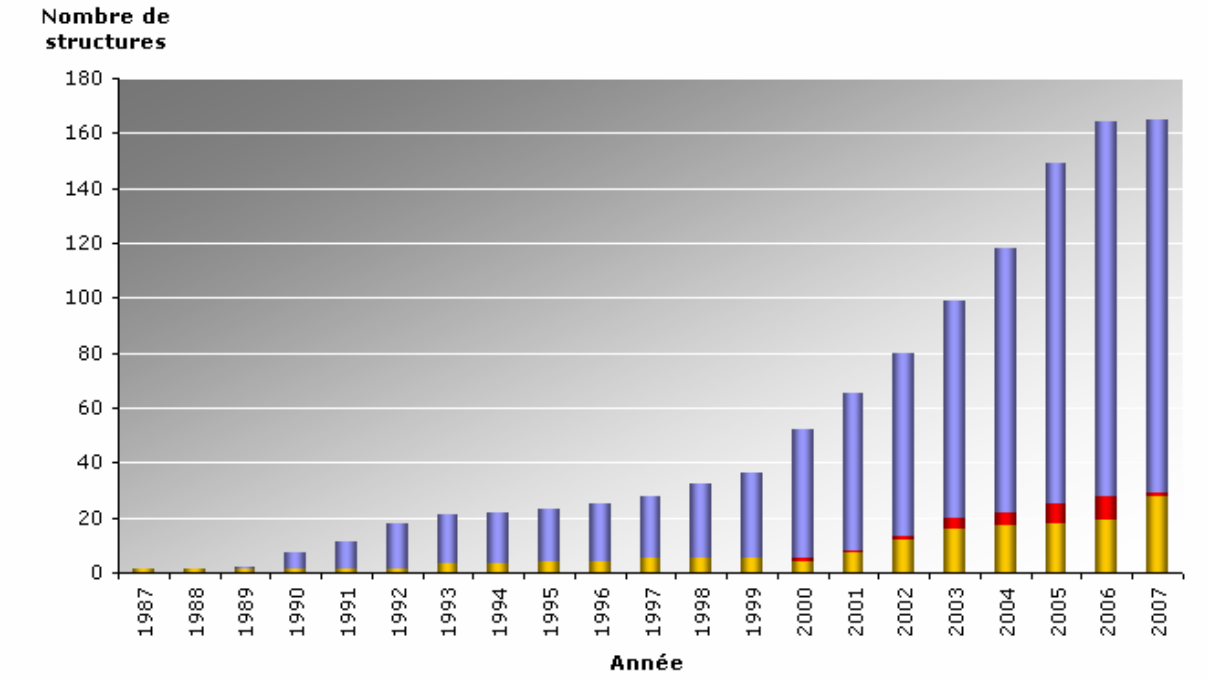


Figure 1-14 Effectif cumulé des structures de P450s publiées depuis 1987. Le nombre de templates est également montré, en jaune pour les bactériens et en rouge pour les microsomaux.

1.4.2 Un repliement général pour tous

En dépit des disparités évoquées précédemment, un nombre suffisant de structures a été résolu pour confirmer que le repliement général et la topographie des P450s sont globalement identiques pour tous les P450s, solubles ou membranaires, quelle que soit leur classe d'appartenance (cf. Figure 1-15). Nombreux sont les articles où cette conservation structurale soulignée par comparaison des structures entre elles avec le dernier en date, celui de M. Otyepka (Oteypka et *al.*, 2006) qui fait un inventaire des différences et conservations entre les structures des P450s de mammifères. Ce repliement est de façon surprenante spécifique au P450s : à ce jour, il n'existe aucune autre protéine non-P450 publiée qui partage cette structure. Les structures ont une forme de prisme triangulaire dont la hauteur mesure environ 65 Å et la base 35 Å. Les éléments de structures secondaires suivent une nomenclature établie sur le $P450_{cam}$ et donnée par Poulos en 1985 (Poulos et *al.*, 1985). Cette nomenclature s'est vue bien entendu modifiée légèrement au fur et à mesure de la découverte de nouvelles structures secondaires d'autres P450s. Un schéma topographique du $P450_{BM3}$ est représenté en Figure 1-16. Les hélices α sont identifiées par des lettres majuscules (auxquels des « ' » ou « '' » viennent s'ajouter) tandis que les feuillets β sont désignés par des chiffres de 1 à 5. Sur cette figure, deux régions dans les structures se distinguent : une région riche en hélices α, bien structurée à droite et une seconde région constituée principalement de feuillets β et de boucles, donc plus flexible, à gauche et en haut. L'hème situé au centre de la molécule et de cette vue, semble assez accessible de la surface. Dans la région riche en hélices α, la longue hélice I de 32 résidus environ traverse la molécule de part et d'autres de la molécule. Au centre de cette hélice I, près du fer de l'hème, se trouve une thréonine (en position 252 dans la numérotation du $P450_{cam}$) impliquée dans l'activation de l'oxygène moléculaire. Le rôle de la thréonine sera décrit dans les paragraphes à venir. Enfin, cette longue hélice I se retrouve dans toutes les P450s et sert souvent d'ancrage lors d'alignement structural.

Les régions structurales les plus conservées dans toutes les structures sont celles qui participent à la chimie d'activation de l'oxygène par le complexe hème-thiolate (Figure 1-16). La première d'entre elles, contient une partie de l'hélice L et la cystéine proximale. Afin d'assurer une stabilisation du ligand cystéinate, une structuration particulière de la boucle contenant la cystéine (Cys-pocket) est observée dans toutes les structures. La seconde région relativement bien conservée est située du côté distal de l'hème et fait partie de l'hélice I.

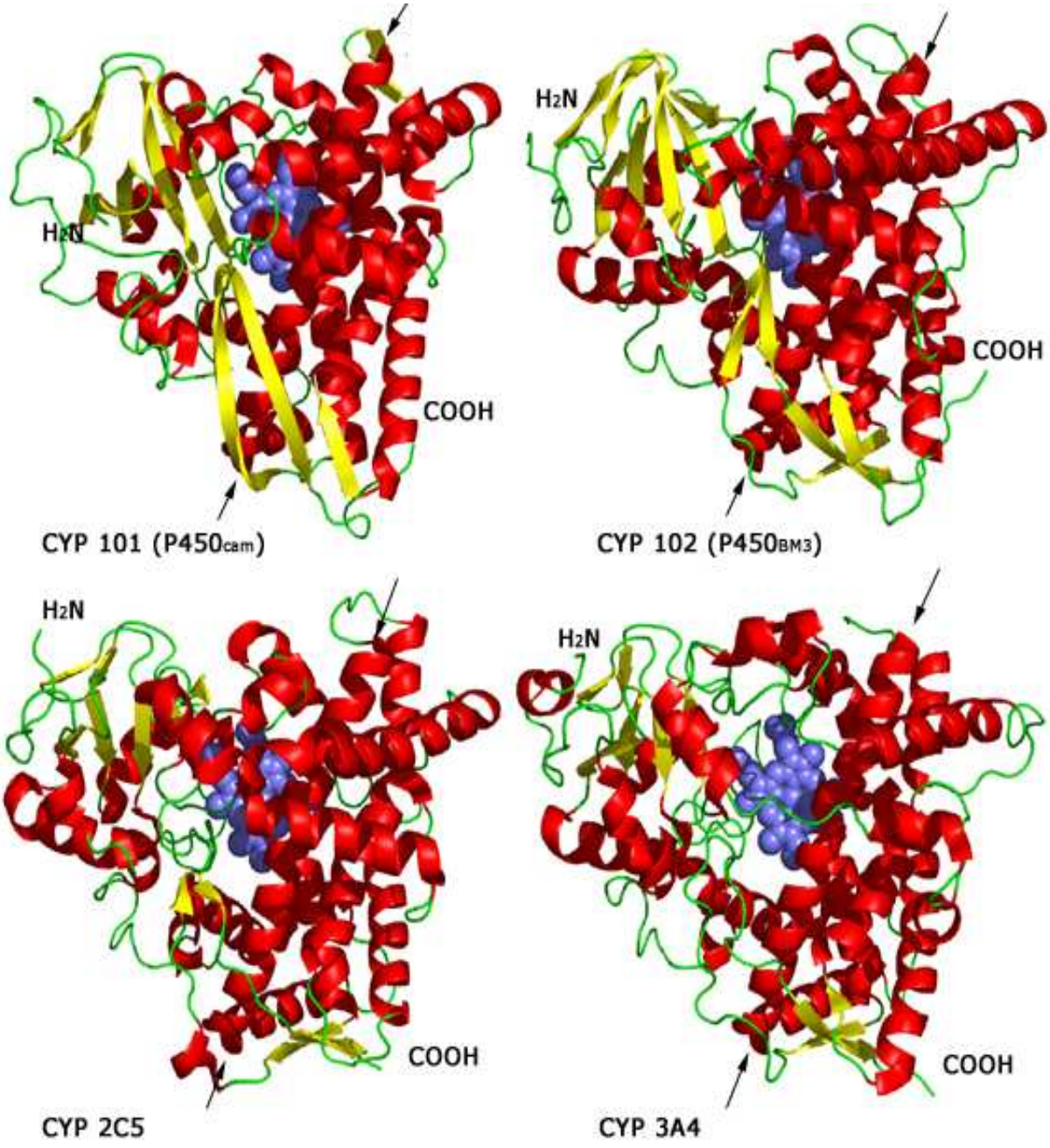


Figure 1-15 Exemples représentatifs de structures tridimensionnelles illustrant le repliement similaire des P450s (images générées avec Pymol avec les structures PDB : 1PHA, 2HPD, 1NR6 et 1TQN). Les hélices sont représentées en rubans rouges, les feuillets en flèches jaunes et l'hème en représentation sphérique bleue. Les petites flèches noires délimitent l'hélice I qui a servi pour aligner les structures dans la présente image. Cette hélice I est présente sur toutes les structures de P450. Elle parcourt la molécule de part et d'autres, ici de haut en bas et de droite vers la gauche. Les 4 structures adoptent une forme relativement conservée, en dépit de quelques différences. La structure du CYP3A4 (1TQN) présente une délétion de quelques résidus dans sa séquence, correspondant à une région qui ne diffracte pas aux RX.

Les parties les plus variables dans les structures quant elles, correspondent à celles qui forment la cavité distale au dessus de l'hème, comprenant le site actif et les canaux potentiels d'accès des substrats comme les hélices B', F, G, et la boucle F-G (ou les hélices F' et G ' chez les P450s membranaires).

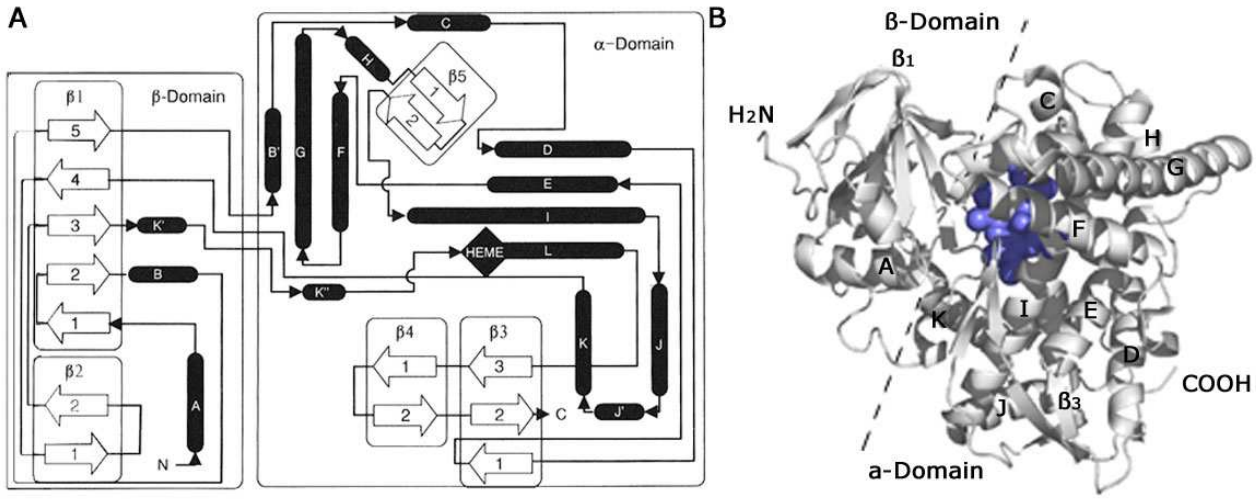


Figure 1-16 (A) Topologie et éléments de structures secondaires du cytochrome P450BM3 reproduit de la p. 157 de Cytochrome P450 (P.R. Ortiz de Montellano, ed.) d'après Graham et Peterson, 1995). Les hélices sont représentées par des barres noires dont la longueur est approximativement proportionnelle à la longueur de l'hélice. Les brins β des feuillets sont représentés par des flèches. L'hème est représenté par un carré en N-terminal de l'hélice L. Avec quelques changements mineurs, cette topographie peut être généralisée à tous les CYPs ; (B) structure 3D du P450BM3 (PDB 2HPD). En raison de l'angle de vue, certaines hélices et certains feuillets n'ont pas pu être annotés. Les hélices α sont annotées en lettre majuscule de A à L et les feuillets β désignés par les chiffres 1 à 5. Sur les deux figures, on observe la présence de deux régions: une riche en feuillets β, annotée β-domain et l'autre riche en hélices α, annotée α-domain.

1.4.2.1 Le site actif de fixation des substrats

Le site actif est constitué dans la majeur partie du temps d'une cavité hydrophobe, généralement bordée par les hélices B', F, G et I, la boucle C-terminale et la boucle située après l'hélice K. Les CYPs membranaires se différencient de leurs homologues bactériens par la structuration en une ou deux hélices de la boucle F-G. Ces hélices, notées F' et G', forment le toit du site actif, comme pour la boucle F-G chez les CYPs cytosolubles. De manière générale, les éléments structuraux bordant le site actif des P450s solubles et membranaires sont les mêmes. Néanmoins, l'arrangement spatial de ces hélices ou boucles est différent, mis à part le positionnement de l'hélice I comme le montre la Figure 1-17.

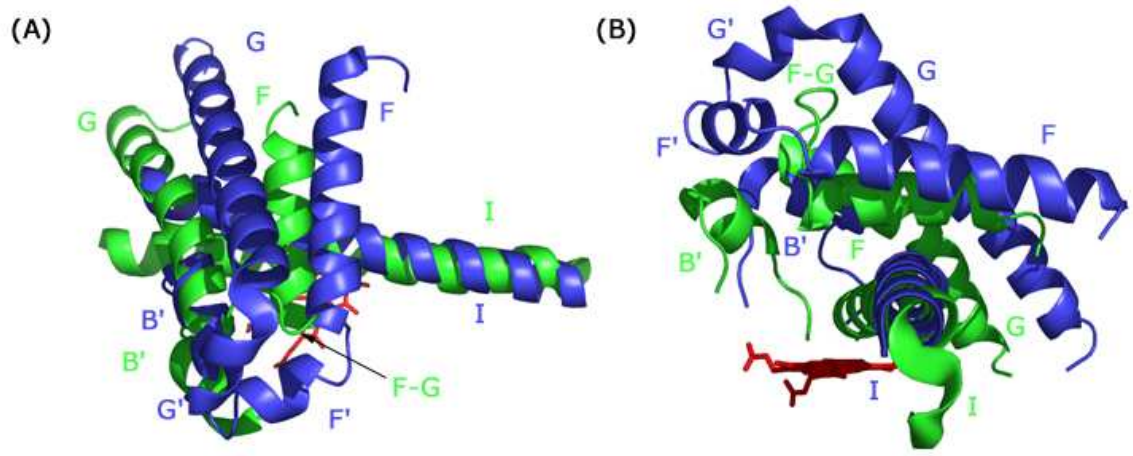


Figure 1-17 Comparaison de structures secondaires bordant les cavités du P450cam (vert) et CYP 2C8 (bleu), en vue de dessus (A) et en vu de profil (B) par rapport à l'hème. Les hélices sont représentées en ruban tandis que l'hème, en rouge est représenté en bâtonnet. Alors que les hélices I coïncident à peu près, on observe un positionnement diffèrent pour les hélices F', F, G, G' et B'

En se basant sur la structure tridimensionnelle du $P450_{cam}$, ainsi que sur les alignements de séquences des CYPs membranaires de la famille 2 sur celle du CYP 101A ($P450_{cam}$) et des informations prédictives de structures secondaires sur les CYP2, O. Gotoh a prédit la position de six régions d'interaction possibles entre le substrat et l'apoprotéine (Gotoh, 1992) (cf. Figure 1-18). Désignés sous le terme de **SRS** (pour « Substrate Recognition Site »), ces zones sont réparties tout au long de la séquence des P450s, recouvrant environ 16% des résidus totaux de la protéine. Ces SRSs correspondent à des zones fortement variables en termes de séquence et ont été repérées grâce à l'observation de substitutions non « synonymes » par l'auteur. Toutes les mutations reportées ou les fragments chimériques ayant significativement affecté les spécificités de reconnaissance du substrat aux enzymes CYP2 parentaux tombent ou chevauchent l'une de ces six régions.

La reconnaissance d'une grande variété de substrats par les P450s pourrait donc être expliquée par le positionnement différent des éléments structuraux bordant le site actif, mais également par la variabilité des séquences au sein des zones en interaction avec les substrats.

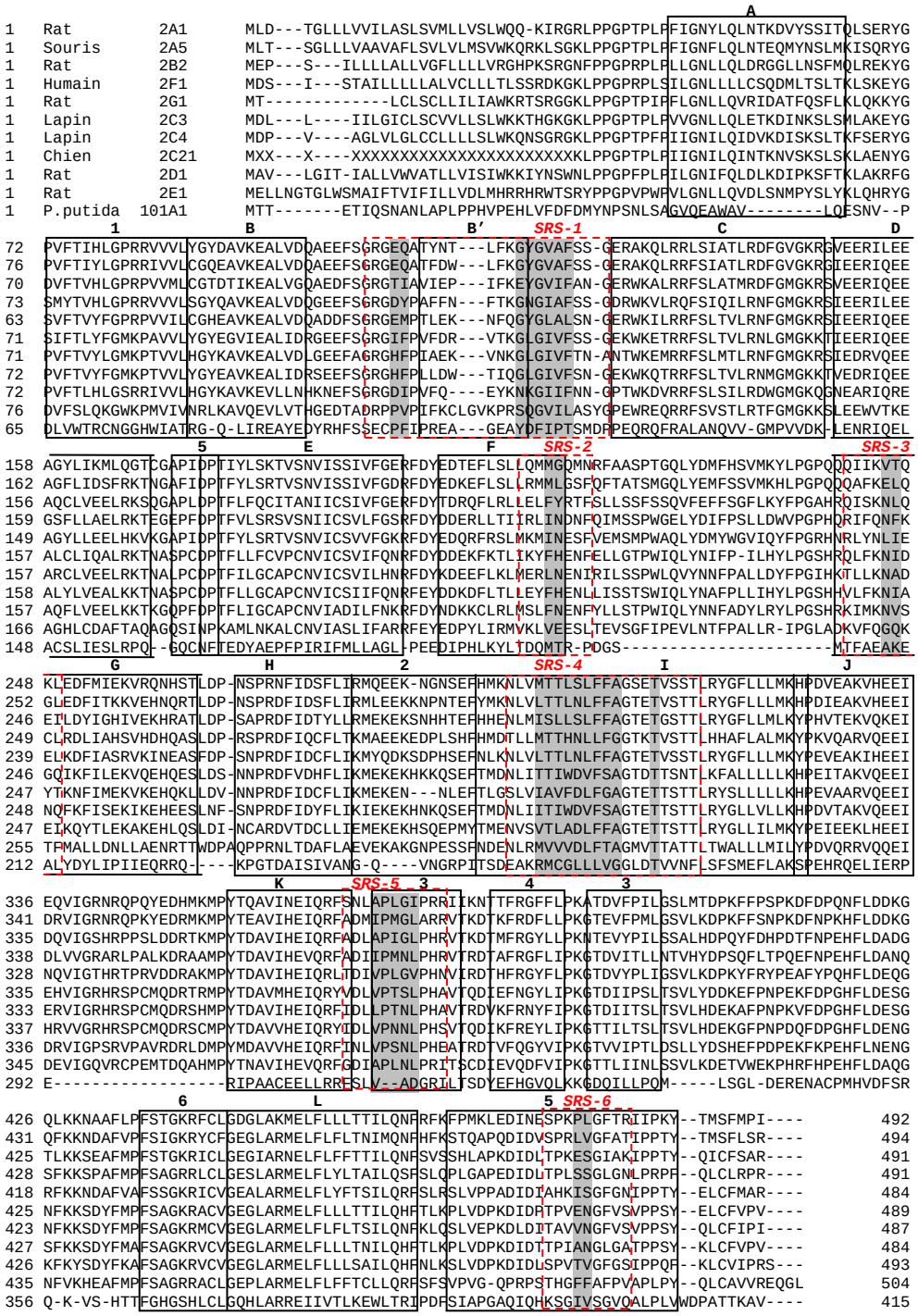


Figure 1-18 Alignement de séquences multiples entre les CYP2 membranaires et le P450cam (source Gotoh, 1992). Les régions correspondant aux hélices et aux feuillets du P450cam sont encapsulés dans un cadre à trait plein. Les zones surlignées en gris correspondent aux résidus de liaison au substrat dans le P450cam (Laughton et *al.*, 1990). Les SRSs sont représentés dans un cadre rouge à trait en pointillé. L'alignement diffère légèrement de la publication de Gotoh en raison des séquences sélectionnées. Il existe une différence de nomenclature entre cet alignement donné par Gotoh et le schéma (A) (celui d'un BM3) de la Figure 1-16. Ainsi, le premier feuillet 5 sur l'alignement, correspond en fait au feuillet 3 de la figure, le premier feuillet 2 à 5, les feuillets entre K et L (3-4-3) correspondent aux feuillets 1-2-1 (avec aussi un feuillet 6 avant le feuillet 1) et le dernier feuillet (5) correspond aux feuillets 3-4-3.

1.4.2.2 Interaction avec la CPR

Avec les structures résolues soit par RX, soit par RMN, de la putidarédoxine de *Pseudomonas putida* (protéine qui se couple au $P450_{cam}$ pour lui fournir des électrons), de nombreux résidus chargés négativement ont été observés en surface de la protéine, suggérant qu'ils pourraient être dirigés vers le CYP. Il en a donc été déduit à l'époque que le $P450_{cam}$ interagissait avec sa protéine de transfert d'électrons par interaction électrostatique, au moyen de ses résidus chargés positivement. Cette hypothèse s'est ensuite vue confirmée lorsque le cristal de $P450_{BM3}$ a pu être étudié. Ce CYP a la particularité d'être lié covalemment à sa CPR, montrant ainsi une surface de contact avec la CPR en position proximale du P450, chargée positivement dans l'ensemble et définie par les hélices B, C et K.

La résolution tridimensionnelle de la totalité de la CPR de rat (Wang et *al.*, 1997), puis la résolution partielle de la CPR humaine (Zhao et *al.*, 1999) ont permis de mieux comprendre les interactions entre ces P450s de classe II et la CPR. De façon analogue aux P450s membranaires, la cristallisation d'une CPR n'est possible qu'après suppression de la partie N-terminale hydrophobe et transmembranaire. En revanche, des études enzymatiques ont montré que chez les mammifères, la CPR dépourvue de sa partie transmembranaire n'est plus en mesure de se coupler avec les P450 membranaires, ce qui démontre l'importance de la membrane dans l'orientation spatiale optimale des deux enzymes entre elles (Paine et *al.*, 2005). Par ailleurs, la surface d'interaction supposée entre le P450 microsomal et la réductase semble être la même que celle décrite pour les P450s solubles. Ce constat ne tient pas compte de la stœchiométrie mesurée, d'un ratio 1/8 pour les P450/réductase décrit précédemment. Certaines hypothèses stipuleraient que les réductases tournent autour d'un P450 à l'image d'un barillet de pistolet. Aucune information ne permet à ce jour de vérifier cette hypothèse.

1.4.3 Structures de complexes P450-substrats. (Exemple du CYP 2C5)

Parmi les structures de CYPs déposées sur la PDB, nombreuses sont celles co-cristallisées de leur substrat. Parmi celles de mammifères, les premières à avoir été réalisées sont celles du CYP2C5/3LVdH, par les équipes d'E.F. Johnson et D. Mansuy. Historiquement, le CYP2C5 (code PDB 1DT6) fut la première structure de mammifère à avoir été réalisée et publiée. Cette dernière a subi quelques mutations et délétions afin d'être cristallisée dans les meilleures conditions. Par la suite, deux autres structures de ce CYP 2C5 ont été réalisées à de meilleurs résolutions, toutes deux co-cristallisées à un substrat dont la taille et la polarité diffèrent : le diclofénac (DIF) et le 4-méthyl-N-méthyl-N(2-phényl-2H-pyrazol-3-yl)benznesulfonamide (DMZ) (Marquest-Soares et *al.*, 2003). Ce fut d'ailleurs les premières structures de complexes P450s de mammifère : celui avec le DMZ est trouvé

sous le code PDB 1N6B et celui avec le DIF, 1R6B. Par superposition différentielle de ces trois structures, il est possible d'observer quelques comportements de la protéine en présence ou absence de son substrat.

Tableau 1.7 Résidus du site actif du CYP2C5 en contact avec le DMZ (Wester et *al.*, 2003a) et le DIF (Wester et *al.*, 2003b)

SRS	***Résidus en contact***		
	Communs aux 2 substrats	***Spécifique du DMZ***	***Spécifique du DIF***
SRS-1	L103, A113, A114	V106	V100
SRS-2	N204, V205	L201, (L208, L213)	–
SRS-3	–	A237, I240	–
SRS-4	D290, G293, A294, T298	S289	–
SRS-5	L359, L363	–	–
SRS-6	F473, V474	–	–

Les résidus du 2C5 en contact avec le DMZ ou le diclofénac sont quasiment les mêmes (cf. Tableau 1.7), et ne présentent pas de différence majeure de polarité. Lorsqu'un substrat se fixe au sein de son site actif, le CYP 2C5 montre une adaptabilité importante des éléments structuraux bordant la cavité du site actif en fonction de la taille et de la polarité du composé (cf. Figure 1-19) :

- Ainsi la présence dans le site actif d'un composé faiblement polaire comme le DMZ correspond à une conformation fermée du P450. La protéine se contracte ainsi autour du substrat, ce qui a pour conséquence de chasser l'eau présente dans la cavité du site actif renforçant alors les interactions favorables avec le substrat. Le déplacement observé d'hélices permettrait donc de boucher les canaux par lesquels pourraient rentrer les molécules d'eau. Conjointement, les deux hélices F et G se rapprochent alors de l'hème par translation perpendiculaire à l'axe de l'hélice I. D'autre part, la boucle B-C se structure pour former une petite hélice B' qui permet d'optimiser les contacts de type hydrophobe avec le substrat. Cette région fait d'ailleurs partie du SRS-1.
- Globalement, les modifications du site actif du CYP 2C5 induites par la fixation du DIF sont analogues à celles observées avec le DMZ (compaction autour du substrat, structuration de la boucle B', fermeture des différents canaux). Cependant, le diclofénac est plus petit que le DMZ et porte une fonction chargée (COOH) qui pointe vers l'extrémité d'un canal. Le reste du canal est comblé alors par un réseau de molécules d'eau, stabilisé par les différents résidus du canal (Figure 1-19 B).

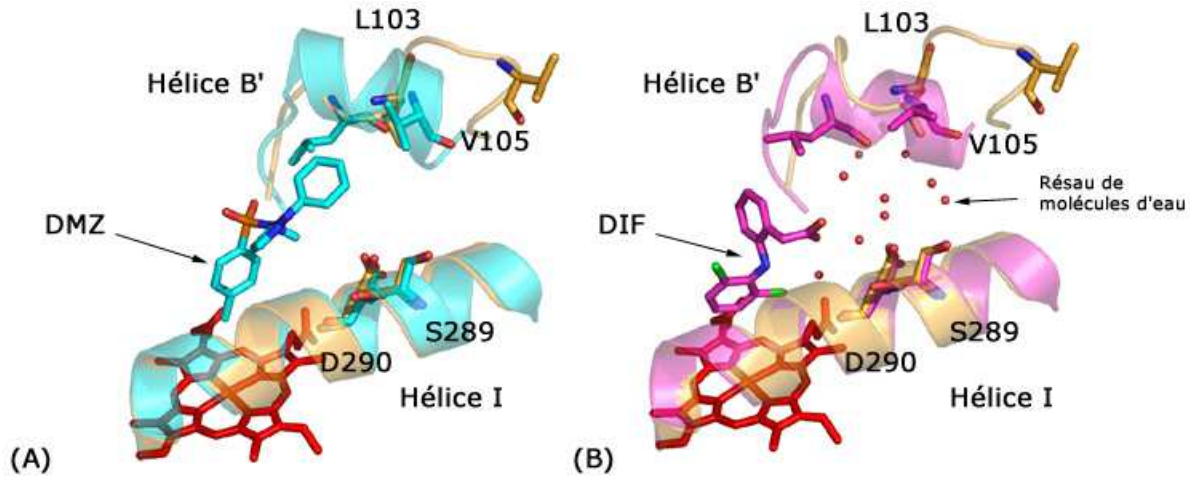


Figure 1-19 Adaptations conformationnelles du site actif du CYP 2C5/3LVdh lors de la fixation du DMZ (A) ou du DIF (B). (d'après Wester et *al.*, 2003a, b). Dans les deux figures, la structure du CYP 2C5/3LVdH sans substrat est représentée en orange. Les portions d'hélices sont représentées en ruban. On observe ici pour les deux cas un structuration de l'hélice B'. Pour le DFI, un réseau de molécules d'eau apparaît pour combler le canal.

1.4.4 Canaux potentiels d'accès au site actif

1.4.4.1 Premiers constats sur les CYPs solubles

Le premier cristal du $P450_{cam}$ complexé avec le camphre a révélé de façon inattendue un site actif profondément enfoui au sein de la protéine, entraînant par la même occasion une série de questions portant sur l'accessibilité des substrats au site actif. En effet, entre les structures du $P450_{cam}$ complexé ou non avec un camphre, aucun canal parfaitement défini n'est constaté. En revanche, les facteurs thermiques (B-factor) des structures X annotés dans le fichier au format PDB (un fichier formaté utilisé pour décrire les protéines publiées dans la PDB), montrent une flexibilité possible des hélices B', F et G. Dans l'article de Poulos (Poulos et *al.*, 1987), les auteurs ont suggéré que l'hélice B', très flexible, permettrait l'entrée du substrat vers le site actif. Plusieurs études menées par d'autres groupes, ont contribué à valider cette hypothèse dont le travail de Dunn et al (Dunn et *al.*, 2002) qui offre une vue d'une structure RX du $P450_{cam}$ co-cristallisé avec un analogue du camphre (cf. Figure 1-20). Dans ce cristal, on peut observer que l'analogue occupe tout un canal d'accès qui se forme par déplacement de l'hélice B' et de la boucle B-C. De nombreux travaux sur les P450s solubles ont par la suite confirmé ce résultat, en apportant des preuves de la présence d'autres canaux possibles dans certains P450s, en fonction du substrat utilisé (Poulos et Johnson, 2005 ; Cojocaru et *al.*, 2007).

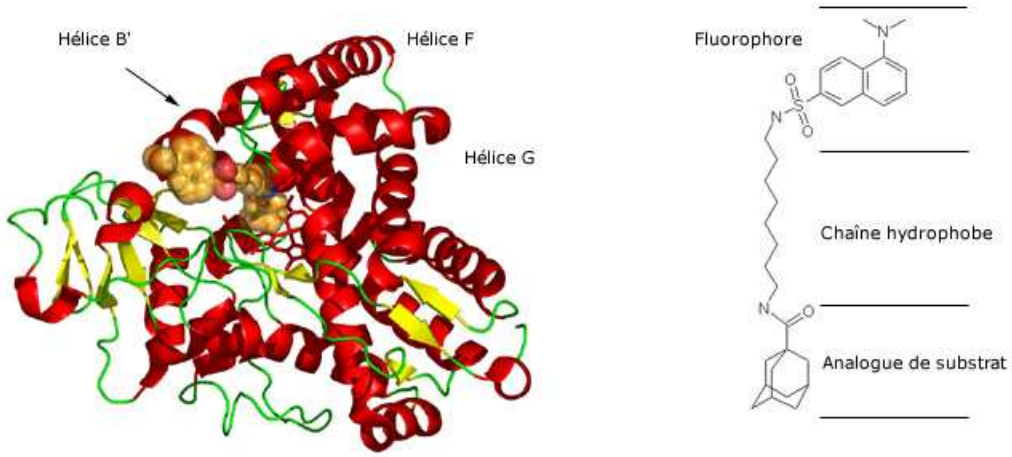


Figure 1-20 P450cam co-cristallisé avec un analogue du camphre (PDB 1LWL) (d'après Dunn et *al.*, 2002). L'analogue du camphre est représenté en sphère orangé au centre du site. Du fait de sa longueur important, il occupe tout un canal d'accès avec la partie fluorescente à l'extérieur de l'enveloppe protéique.

1.4.4.2 Identification des canaux sur les CYPs de mammifères

Contrairement aux CYPs solubles, les CYPs microsomaux ont une boucle F-G beaucoup plus longue qui se structure généralement en deux petites hélices F' et G'. Cette région interagit beaucoup avec la région en feuillets β de la partie N-terminale et probablement avec la membrane, limitant les possibilités d'ouverture évoquées précédemment chez les P450 solubles. Cependant, l'hélice B' reste encore très flexible et peut donc jouer un rôle dans l'ouverture d'un canal d'accès aux substrats. Ainsi, à partir des structures du CYP 2C5/3LVdH précédemment décrites, E.F. Johnson a proposé un canal d'accès aux substrats dont l'entrée serait délimitée par les hélices B', I et G et par la boucle B'-C (Wester et *al.*, 2003a). A l'instar de leur homologue soluble, l'arrivée de nouvelles structures avec et sans ligand ont permis de découvrir d'autres canaux visibles. La structure la plus riche en informations reste celle du CYP 2B4 de lapin sans substrat (Scott et *al.*, 2003). Cette structure présente une large ouverture entre les hélices F et G d'une part et l'hélice B' d'autre part. L'ouverture de ce canal est telle, qu'une histidine d'un autre monomère de CYP 2B4 vient complexer le fer. Deux autres structures du CYP 2B4 ont été ensuite publiées toutes deux complexées avec un substrat de taille différente. L'une d'elle (PDB 1SUO) présente un substrat 4-(4-chlorophenyl)imidazole – CPI – qui met en évidence la très grande flexibilité de la protéine dans la région décrite (Scott et *al.*, 2004 ; Zhao et *al.*, 2006). (cf. Figure 1-21)

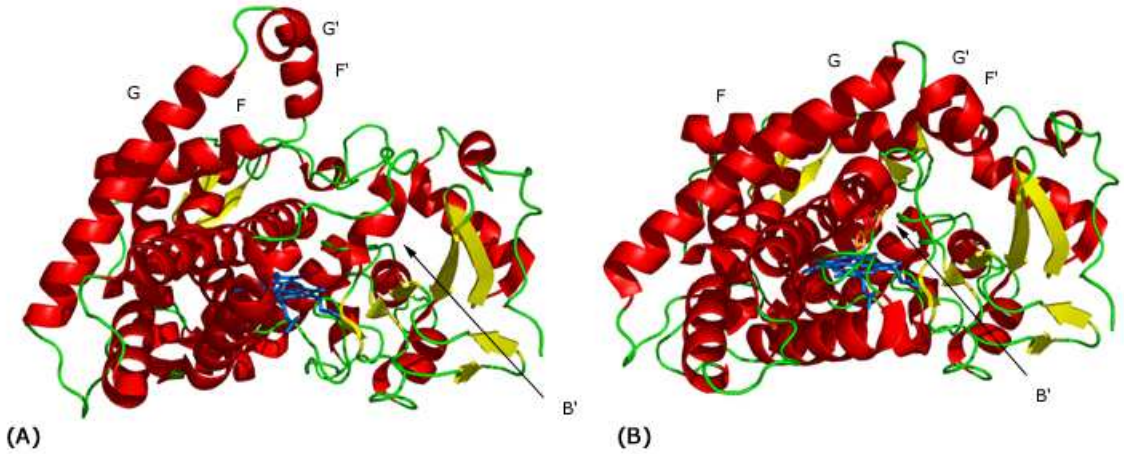


Figure 1-21 Comparaison des structures du CYP 2B4 (A) sans substrat (PDB 1PO5); (B) avec le CPI (PDB 1SUO). Pour les deux structures, l'hème figure en rouge en style bâtonnet, les hélices α sont représentées en ruban rouge, les brins β en flèches jaunes. Le CPI sur la figure (B) est représenté en bâtonnet orange, au dessus de l'hème. Dans la structure sans ligand (A), les hélices F et G sont éloignées du plan de l'hème et la boucle B-C est structurée de façon différente à celle de la seconde structure. D'après Scott et *al.*, 2004 ; Zhao et *al.*, 2006.

Des analyses par dynamique moléculaire (MD) (la technique sera décrite dans le prochain chapitre) d'expulsion de substrat menées par l'équipe de R.C Wade ont permis de compléter ces hypothèses sur l'accès au site actif, en démontrant la présence possible d'autres canaux en fonction des substrats utilisés et montrent que les canaux d'entrée peuvent varier selon le substrat à métaboliser. (Ludemann et *al.*, 2000 ; Winn et *al.*, 2002 ; Wade et *al.*, 2004 ; Schleinkofer et *al.*, 2005). Ces techniques de MD ont permis notamment de faire apparaître des canaux lors d'observation de la protéine en mouvement, canaux qui n'étaient pas percevables sous observation figée de la structure RX.

1.4.4.3 Bilan des canaux d'entrée et de sortie, selon Cojocaru et al.(Cojocaru et al., 2007)

V. Cojocaru a récemment publié un article faisant le bilan de toutes les entrées et sorties possibles, répertoriés sur l'ensemble des P450s disponibles en Mars 2006, soulignant par la même occasion quelques différences entre les mécanismes d'entrées/sorties pour les P450s bactériennes et ceux de mammifères. La recherche des canaux a été opérée à l'aide du logiciel CAVER (Petrek et *al.*, 2006) (http://loschmidt.chemi.muni.cz/caver/). L'ensemble des canaux repérés est décrit dans la Figure 1-22 et le Tableau 1.8.

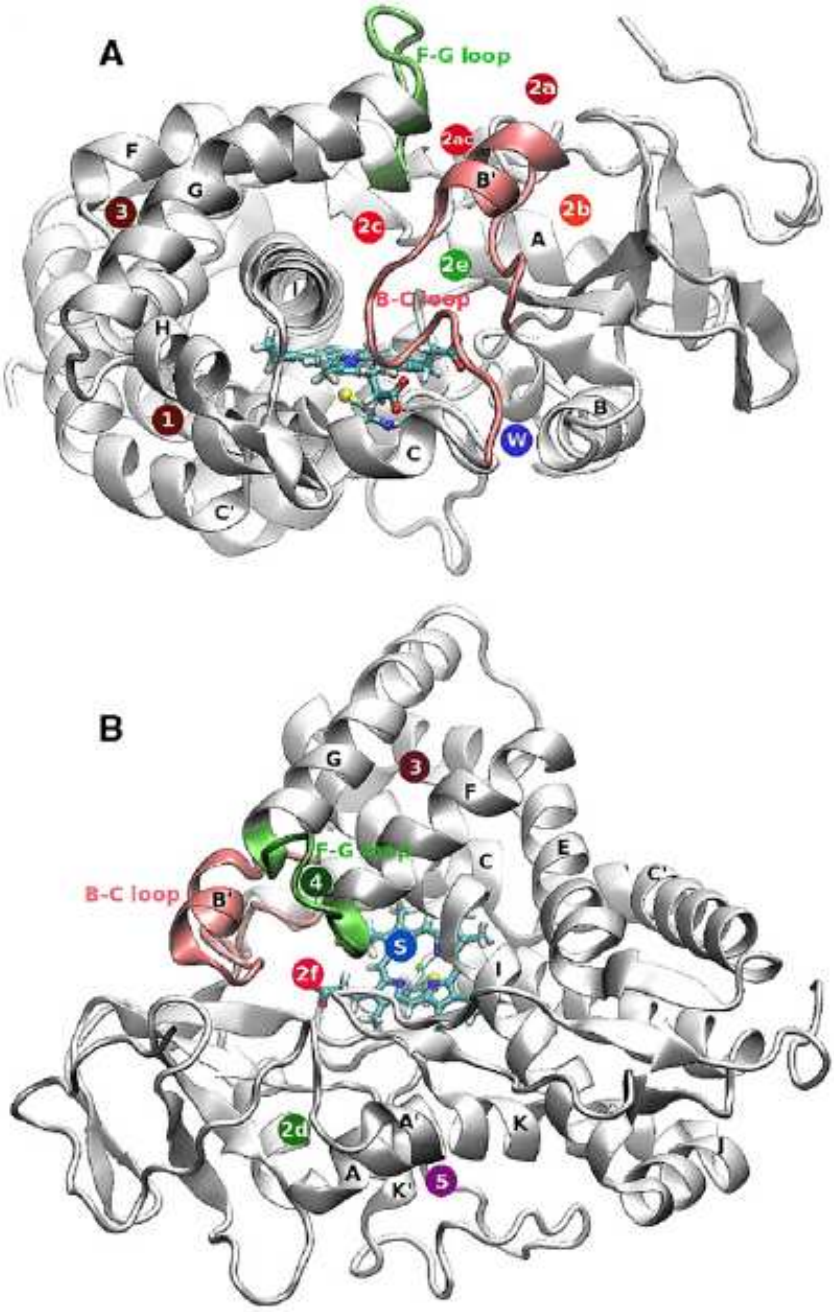


Figure 1-22 Canaux d'accès potentiels des CYPs. (source Cujucaro) Tous les canaux calculés ont été annotés sur la structure du P450cam (PDB 1ATKD). La protéine, en grise, est en représentation cartoon (hélice en ruban et brin en flèches) Les ouvertures de canaux sont montrées à l'aide de cercles numérotés à la position correspondante en surface de la protéine. Les canaux partageant une même couleur sont liés par (i) leur mode d'ouverture (2c, 2ac, 2a, 2f, 2b – en rouge ; 2c, 2d, 4 – en vert ; 1,3 – en marron), (ii) leur fonction [solvant (S), water (W) – en bleu], ou (iii) récemment identifié (5 – violet). Chaque canal est montré avec une nuance de couleur différente. La boucle B-C est coloriée en rose tandis que la boucle F-G est en vert. Deux vues distinctes sont proposées pour pouvoir observer l'ensemble des canaux : (A) une vue de profil en regardant droit ver l'hélice I et (B) une vue de dessus de l'hème, à partir de sa face distale.

Tableau 1.8 Localisation et occurrence des canaux d'accès au site actif des CYPs disponible en Mars 2006 (d'après Cojocaru et *al.*, 2007)

Canal	***Localisation***	***Commentaires***	***Observé dans les P450s***
1	Apparaît entre les hélice C/C' et H ou L, près de la boucle G-H et le feuillet 2	Canal rare, observé en MD. Possibilité de route pour les espèces gazeuses	$P450_{cam}$
2	Toute les sous-classes exceptées 2d et 2f sont bordées par le boucle B-C/hélice B'. Cette région de la protéine est hautement variable à la fois en séquence et en structure entre les différents P450s. Elle est décrite comme importante pour la spécificité du substrat à travers les contacts qu'elle établit avec lui comme il a été vu précédemment.	-	–
2a	Apparaît entre la boucle F-G, l'hélice B'/boucle B-B'/Boucle B-C et le feuillet 1. Toute cette région est extrêmement variable en séquence et en structure chez les P450s. La région C-terminale de l'hélice F forme le SRS-2 et la région N-terminale de l'hélice G forme le SRS-3	Canal commun	CYP119, $P450_{st}$, $P450_{nor}$, CYP2B4, CYP2C9, $P450_{cam}$, $P450_{BM3}$, $P450_{Terp}$, CYP121, CYP152A2, CYP154C1, CYP158A2, $P450_{OxyB}$, $P450_{OxyC}$, $P450_{epoK}$, CYP175A1
2b	Apparaît entre la boucle B-B', les feuillets 1 et 3 (SRS-5 présent dans le feuillet 3)		CYP2B4, CYP2C8, CYP3A4, P450BM3, CYP152A2
2c	Apparaît entre l'hélice Get I et l'hélice B'/boucle B-C. L'hélice I contient le SRS-4	Canal commun	CYP119, $P450_{Nor}$, CYP2B4, CYP2C5, CYP2C8, CYP2C9, CYP154C1, CYP158A2
2ac	Apparaît entre le sommet de la boucle B-C (ou hélice B') et l'hélice G entre les canaux 2A et 2c	Canal commun	CYP119, $P450_{Nor}$, CYP2B4, CYP2C5, CYP2C9, CYP154A1, CYP154C1, CYP158A2 $P450_{OxyB}$, CYP175A1
2e	Apparaît à travers le boucle B-C	Canal assez commun, Il a été montré comme une seconde sortie possible dans la MD du P450eryF	CYP2B4, CYP2C5, CYP2C8, CYP2C9, CYP3A4, CYP51, $P450_{EryF}$, CYP121, CYP152A2, CYP154C1, $P450_{OxyB}$, $P450_{epoK}$
2d	Est spatialement proche du 2a d'où l'appartenance à une sous-classe 2. Apparaît entre la région N-terminal de la protéine et les hélices a/A' et A.	Anciennement observé exclusivement chez $P450_{BM3}$ avec un acide palmitoleïque non chargé.	CYP2B4, $P450_{BM3}$, CYP154C1
2f	Idem que 2d pour la définition. Apparaît entre l'hélice F'/boucle F-G et le feuillet 5	Canal commun	CYP119, P450Nor, CYP2B4, CYP2C5, CYP2C9, P450cam, P450BM3, P450Terp, CYP158A2, P450OxyB, P450OxyC, CYP175A1
3	Apparaît entre les hélice F et G ou au niveau de la boucle E-F	Canal rare, identifié comme un substrat (ou produit) secondaire potentiel en MD	$P450_{cam}$
4	Apparaît au travers de la boucle F-G	Canal rare	CYP119, CYP2B4, CYP2C8
5	Apparaît entre les hélices K et K'	Canal rare	CYP2B4, CYP158A2, CYP175A1
S	Apparaît entre les hélice E,F et I et le feuillet 5	Présent dans la moitié des P450s, possibilité de sortie pour les substrats, mais non confirmé jusqu'à présent	CYP119, P450Nor, CYP2B4, CYP2C5, CYP2C8, CYP2C9, CYP3A4, $P450_{cam}$, $P450_{BM3}$, $P450_{Terp}$, CYP158A2, $P450_{OxyB}$, $P450_{OxyC}$
W	Apparaît à la base de la boucle B-C près du C-terminal de l'hélice B	Entrée de l'eau pour l'apport en oxygène moléculaire.	$P450_{BM3}$

1.5 Propriétés chimiques des P450s

A partir de ces éléments structuraux, il est possible de s'intéresser aux mécanismes mis en jeu lors des biotransformations. Comme il a été déjà souligné, l'élément essentiel qui permet ces réactions est l'hème des P450s, plus particulièrement le fer qu'il porte. Ces réactions ont également besoin d'un apport d'électrons et d'oxygène moléculaire lors de la réaction enzymatique.

1.5.1 Implication du fer de l'hème dans les interactions

L'atome de fer est en mesure de former 6 liaisons de coordination. Dans le cas des CYPs, le fer est pentacoordiné par les 4 azotes pyrroliques de l'hème et par le thiolate de la cystéine. Le sixième ligand peut être soit une molécule d'eau, soit un atome coordinant d'un composé fixé dans le site actif. Ces variations dans l'état de coordination du fer se traduisent par des signatures spectrales caractéristiques dans le domaine de l'UV visible (cf. Figure 1-23).

1.5.1.1 État natif

A l'état natif, le Fe^{III} de l'hème existe en équilibre sous deux états de spin, dépendant de son état de coordination :

- Un état de bas spin S=1/2, dans lequel le fer est hexacoordiné et dans le plan de l'hème. Le sixième ligand est presque toujours une molécule d'eau. Le maximum d'absorption de cette espèce Fe^{III} est situé aux environs de 420 nm.
- Un état de haut spin S=5/2 correspond à un Fe^{III} pentacoordiné situé en dehors du plan de l'hème. Cette espèce a un maximum d'absorption à 390 nm

1.5.1.2 En présence d'un composé

En présence d'un composé exogène qui se fixe dans le site actif, trois cas de figure se présentent :

- Lorsque le composé hydrophobe rentre dans le site actif, il chasse la molécule d'eau coordinant le fer dans l'état de bas spin. Le Fe^{III} se retrouve donc à l'état de haut spin, avec un maximum d'absorption à 390 nm. On appelle cette interaction, une **interaction de type I** dont le spectre différentiel est constituée d'un pic à 390 nm et d'une vallée à 420 nm.
- Si ce composé porte un atome d'oxygène accessible au fer (comme une fonction alcool ou ester) cet oxygène peut complexer le fer pentacoordiné et donner une **interaction de type I inversé**. On retrouve ainsi le Fe^{III} sous forme de bas spin avec un pic d'absorption en UV visible à 420 nm. Le spectre différentiel résultant comporte alors un pic à 420 nm et une vallée

à 390 nm. A noter que dans certains cas, l'arrivée d'un substrat dans le site actif peut favoriser la fixation d'une molécule d'eau comme sixième ligand du fer et donner également un telle interaction.

- Une **interaction de type II** est observée lorsque le composé exogène comporte un atome d'azote ou de soufre pouvant complexer le Fe^{III} et se fixe au sein du site actif. Le Fe^{III} absorbe alors entre 425 et 435 nm. Le spectre différentiel montre ainsi un minimum vers 390-410 nm et un maximum vers 425-435 nm

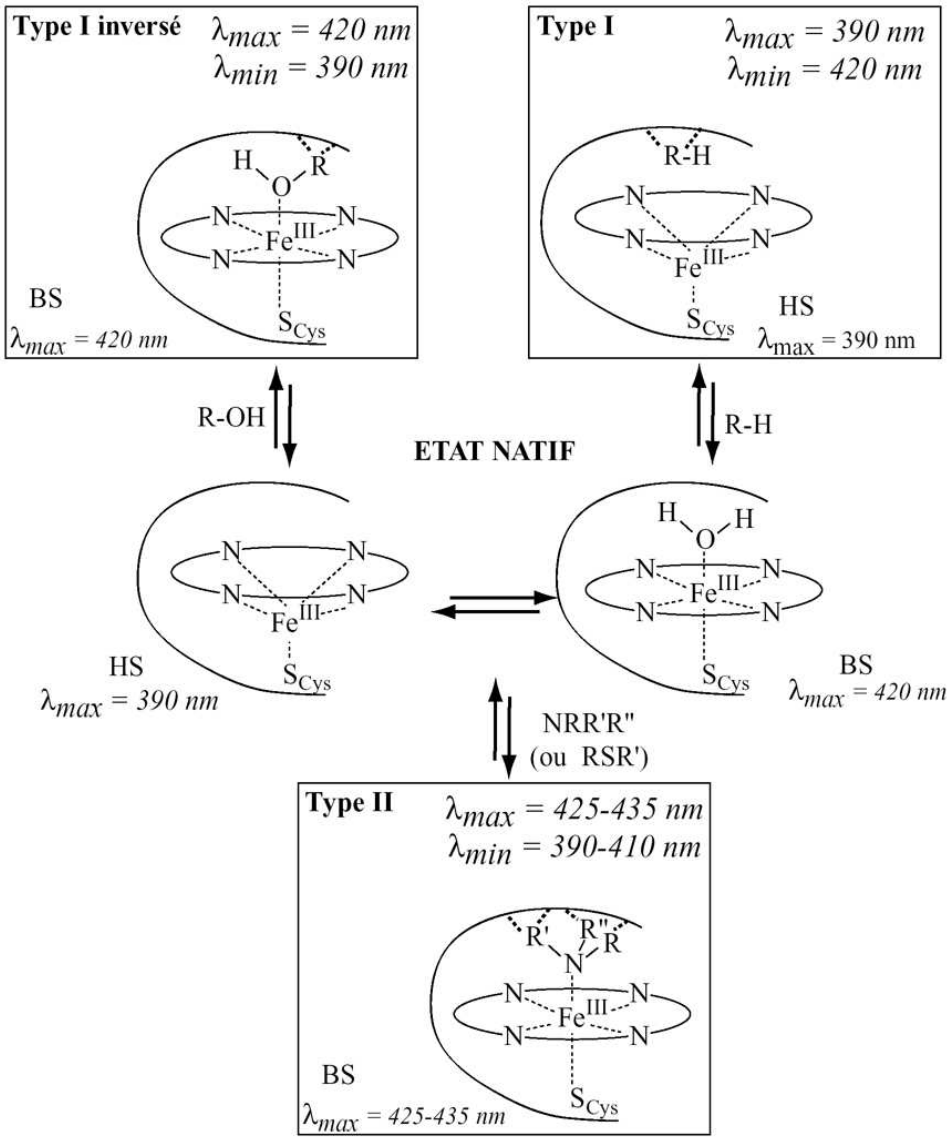


Figure 1-23 Interactions spectrales observées lors de fixation d'un composé dans le site actif (d'après Mansuy et *al.*, 1989) (Source : Thèse de P. Lafite). HS : Haut Spin ; BS : Bas Spin.

1.5.2 Cycle catalytique de la mono-oxygénation des substrats par les CYPs

Le cycle catalytique des CYPs est communément décrit en 7 étapes principales.

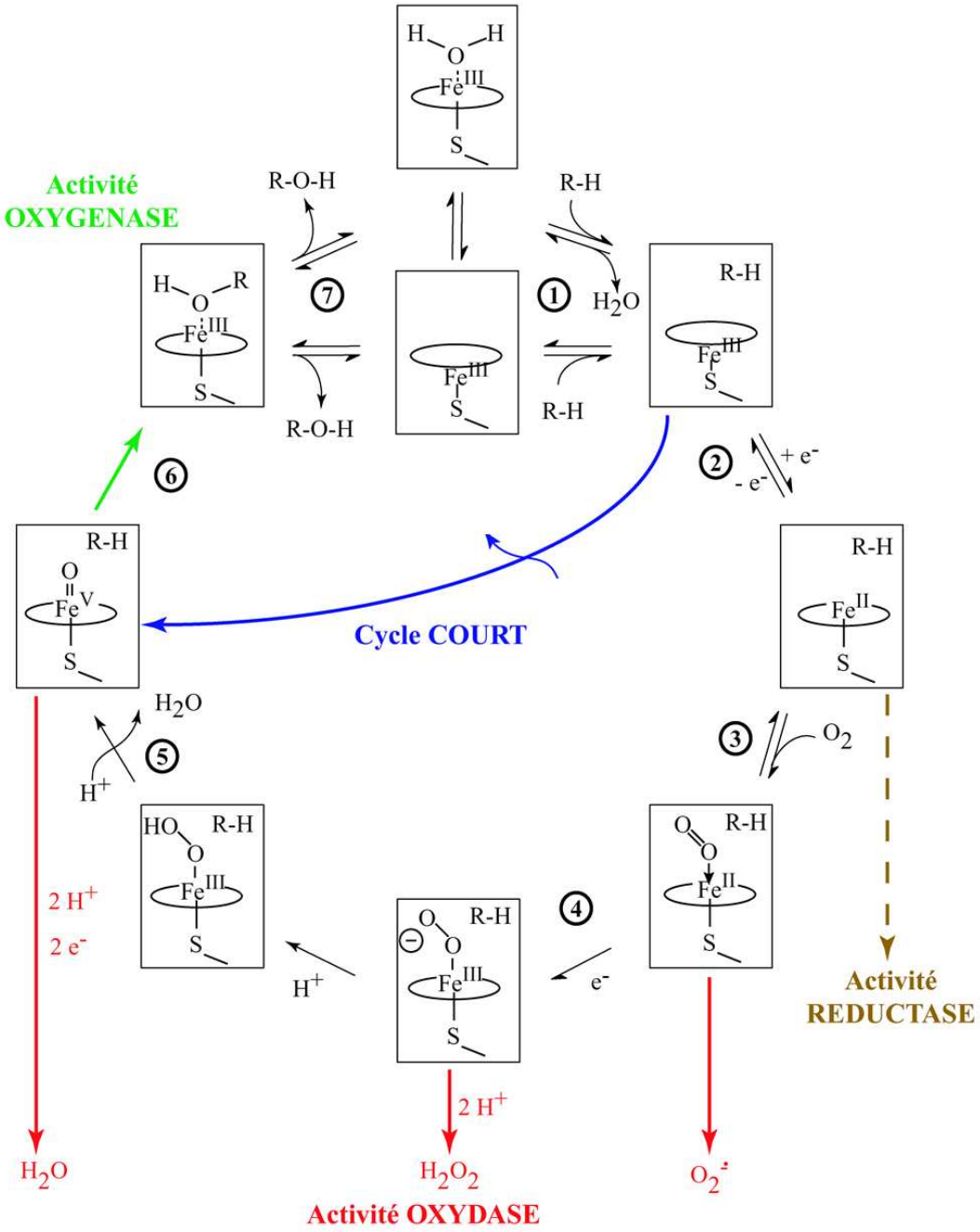


Figure 1-24 Cycle catalytique des CYPs (Makris et *al.*, 2005; Mansuy et Battioni, 2000) (Source : Thèse de P. Lafite) Le cycle court obtenu par addition d'un composé donneur d'atome d'oxygène est indiqué en bleu (aucun exemple n'est donné) ; les voies non-productives ou abortives en rouge et la voie réductase en marron.

1. Le cycle catalytique des CYPs est initié par la fixation du substrat au sein du site actif. Ceci entraîne dans le cas de l'état natif à bas spin le déplacement de la molécule d'eau distale et favorise la position pentacoodinée haut spin du complexe ferrique Fe^{III}.
2. Cette fixation du ligand induit une augmentation du potentiel du couple Fe^{III}/Fe^{II}, permettant ainsi le transfert d'un électron qui peut réduire le fer en complexe ferreux Fe^{II}. Dans certains cas, tels les P450s d'eucaryotes, il semblerait que le transfert d'électron puisse se produire sans fixation d'un composé dans le site actif (Guengerich et Johnson, 1997).
3. La réduction du fer est suivie par la fixation de l'oxygène moléculaire sur le fer, donnant le complexe ferreux $Fe^{II}\leftarrow O_2$.
4. Le transfert d'un second électron vers ce complexe réduit l'oxygène moléculaire en formant l'entité peroxy-ferrique $Fe^{III}–OO^-$ qui par protonation donne ensuite un complexe hydroperoxo $Fe^{III}–OOH$.
5. Une seconde protonation provoque ensuite la rupture hétérolytique de la liaison O–O, libérant une molécule d'eau et le composé fer-oxo formellement écrit $Fe^{V}=O$.
6. Ce composé est hautement oxydant. Il peut transférer son atome d'oxygène au substrat.
7. Le métabolite sort du site actif.

Dans certains cas, l'oxydation du substrat peut être réalisée par l'espèce $Fe^{III}–OOH$. Les travaux de A. Vaz ont ainsi montré que certaines époxydations d'oléfines sont réalisées par cette entité et non par le fer-oxo $Fe^{V}=O$ (Vaz et *al.*, 1998). Par ailleurs, l'utilisation d'agents oxydant donneurs d'atome d'oxygène, tels que l'iodosobenzène, les peracides, les hydroperoxydes, les anions ClO_2^- ou IO_4^-, permettent d'obtenir directement l'espèce réactive fer-oxo à partir de l'état natif. On parle alors de cycle court des CYPs.

1.5.2.1 Voies abortives

Dans le cas idéal, la consommation d'une molécule de NAD(P)H doit s'accompagner de l'oxygénation d'une molécule de substrat, comme dans le cas du $P450_{cam}$ ou de P450s spécifiques des voies de biosynthèses endogènes. Toutefois, dans le cas des P450s impliqués dans la métabolisation des xénobiotiques, ce couplage n'est pas aussi efficace : des voies secondaires de transfert d'électrons dites « abortives » peuvent se produire. Ceci se traduit par retour à l'état de complexe Fe^{III} sans oxydation du substrat. Ce découplage entre le transfert d'électrons et transfert d'atome d'oxygène peut apparaître au niveau de plusieurs étapes du cycle catalytique :

- Le complexe ferreux $Fe^{II}\leftarrow O_2$ peut se décomposer en formant l'ion superoxyde $O_2^{\bullet -}$ et le complexe ferrique Fe^{III}. Cet ion superoxyde peut alors se dismuter pour donner du peroxyde d'hydrogène et de l'oxygène moléculaire :

$$2O_2^{\bullet -} + 2H^+ \longrightarrow H_2O_2 + O_2$$

- L'addition du proton sur l'atome d'oxygène ligand du fer des complexes peroxyferrique Fe^{III}–OO^- ou hydroperoxo Fe^{III}–OOH se traduit par la formation du Fe^{III} et d'une molécule de peroxyde d'hydrogène :

$$Fe^{III} - OO^- + H^+ \longrightarrow Fe^{III} - \underset{H}{O}O^- \xrightarrow{H^+} H_2O_2 + Fe^{III}$$

$$Fe^{III} - OOH + H^+ \longrightarrow Fe^{III} - \underset{H}{O}OH \longrightarrow H_2O_2 + Fe^{III}$$

- L'entité fer-oxo FeV=O peut redonner également l'état natif Fe^{III} après consommation de deux protons et de deux électrons :

$$Fe^{V} = O + 2H^+ + 2e^- \longrightarrow H_2O_2 + Fe^{III}$$

L'ion superoxyde et le peroxyde d'hydrogène libérés, constituent tous deux une source potentielle de stress oxydant. Cette activité oxydase des P450s est généralement plus importante que l'activité catalytique oxygénase présentée précédemment dans le cas des P450s microsomaux, mais ce découplage dépend de paramètres qui influent sur la vitesse de transfert des électrons ou des protons, comme l'accessibilité de l'eau au site actif, un positionnement du substrat trop éloigné du fer, l'absence de sites suffisamment réactifs sur le substrat (Mueller et *al.*, 1995)…

1.5.2.2 Quelques remarques sur l'existence des intermédiaires du cycle catalytique

Le cycle catalytique des cytochromes P450 a été étudié en détail dans le cas de $P450_{cam}$ et de certains de ses mutants, catalysant tous l'hydroxylation régio- et stéréo- sélective du camphre.

- Les premiers intermédiaires du cycle catalytique des P450s (les deux états natifs, Fe^{III}/Camphre et Fe^{II}/Camphre) ainsi que le dernier complexe (Fe^{III}/hydroxycamphre) ont été caractérisés par de nombreuses études cristallographiques et spectroscopiques (Markis et *al.*, 2005). Très peu de différences structurales sont observées entre les deux composés Fe^{III}/Camphre et Fe^{II}/Camphre (Schlichting et *al.*, 2000).
- La structure du complexe Fe^{II}–O_2 a été résolue par cryo-cristallographie à 100K (Schlichting et *al.*, 2000). Celle-ci révèle un déplacement du camphre, une apparition de molécules d'eau et la stabilisation de l'oxygène par des liaisons hydrogène auxquelles participent certains acides aminés du site actif tels que l'Asp251 et la Thr252.

- En dépit du fait que l'existence des complexes peroxo-, hydroxo- et oxo du fer n'a pas pu être prouvée par des expériences de cristallographies, l'équipe de B. Hoffman a pu quand même mettre en évidence par spectroscopie ENDOR et RPE les complexes peroxo- et hydroxperoxy-. Ainsi, ils ont irradié par des rayons γ des échantillons à très basse température (77K et 6K), figeant les structures à chaque étape du cycle, permettant ainsi la visualisation des complexes (Davydov et *al.*, 2001). En revanche, l'espère Fe^{V}=O n'a pas pu être observée : la transformation du complexe Fe^{III}–OOH produit directement l'espèce Fe^{III}/hydroxycamphre sans passer par le fer-oxo.

1.5.2.3 Importance de l'Asp251 et la Thr252 dans le mécanisme d'activation de l'oxygène

La structure tridimensionnelle de l'intermédiaire Fe^{II}←O_2 révèle l'importance de la poche distale dans le transfert des protons et dans l'activation de l'oxygène (Schlichting et *al.*, 2000). Des changements conformationnels du squelette peptidique et des chaînes latérales autour du site actif se produisent lors de la fixation du dioxygène entraînant l'apparition d'un réseau de liaisons hydrogène. Dans ce mécanisme d'activation de l'oxygène, les résidus Asp251 et Thr252 semblent jouer un rôle important :

- Des études isotopiques sur des mutants D251N (Aspartate en position 251 mutée en Asparagine) ont souligné l'importance du rôle structural d'Asp251 dans le réseau de molécules d'eau évoqué précédemment, mais également dans la coupure hétérolytique de la liaison O–O (Deprez et *al.*, 1994 ; Gerber et Sligar, 1994).
- Les études de RPE et ENDOR cryogénique sur le mutant T252A (Thréonine à la position 252 en Alanine) ont montré quant à eux l'existence de l'espèce FeIII–OOH sans observer l'hydroxylation du substrat (Davydov et *al.*, 2001). D'après ce résultat obtenu chez le $P450_{cam}$, on est en mesure d'affirmer que la thréonine 252 est impliquée dans le transfert du second proton du cycle catalytique. Pourtant, il semblerait que le transfert de ce second proton, menant à la coupure hétérolytique de la liaison O–O, soit fortement modulée par la structure du substrat fixé dans le site actif.

Chez les autres P450s, tels que ceux de la famille 2, D.F Lewis a montré qu'une mutation opérée sur la thréonine correspondante à celle du 252 chez $P450_{cam}$ a pour conséquence une réduction important de l'activité. A l'opposé, certains P450s qui ne possèdent pas cette thréonine ont pourtant une activité catalytique (Hiroya et *al.*, 1994 ; Cupp-Vickery et Poulos, 1995).

1.6 Au centre d'un système de détoxication

1.6.1 Place du CYP dans le système de détoxication

Outre son implication dans les voies endogènes, la fonction de biotransformation confère aux CYPs une place privilégiée dans les systèmes de détoxication. En effet, l'une des sources premières d'agression quotidienne que subissent les organismes vivants provient de l'environnement. Ces agressions sont liées de façon directe ou indirecte à des substances d'origine diverse, communément désignées sous le terme de xénobiotique. On retrouve parmi ces substances des produits naturels, médicaments ou encore polluants de l'environnement : toxines végétales et animales, dérivés des combustibles domestiques et industriels, solvants, colorants, additifs alimentaires, pesticides, herbicides, etc.… Ces composés ont pour principales caractéristiques d'être de faible poids moléculaire d'une part et hydrophobes d'autre part. En raison de ces caractéristiques, ils ont une tendance naturelle à s'accumuler dans les phases lipidiques des membranes cellulaires, rendant ainsi leur élimination par voies classiques (urine et bile) très difficile. Ce phénomène résulterait en une mort inéluctable des organismes par effet toxique, si ceux-ci ne s'étaient pas dotés, au cours de l'évolution de systèmes enzymatiques permettant leur élimination.

Ce processus de détoxication se déroule en trois phases (cf. Figure 1-25) qui mettent en jeu différents enzymes de biotransformation (Phase I et II) présentés dans le Tableau 1.9 et des transporteurs (Phase III). L'ensemble forme un système connu sous l'appellation d'Enzymes du Métabolisme des Xénobiotiques (EMX). Ces trois phases sont désignées comme il suit :

- Phase I ou phase de fonctionnalisation
- Phase II ou phase de conjugaison
- Phase III à transporteurs ATP-dépendant

Il peut être toutefois signalé que le métabolisme des xénobiotiques ne conduit pas toujours à la détoxication de ces composés. En effet, dans certains cas, le produit oxydé ou réduit peut manifester un très forte réactivité par attaque électrophile ou nucléophile des macromolécules environnantes (protéines, acides nucléiques).

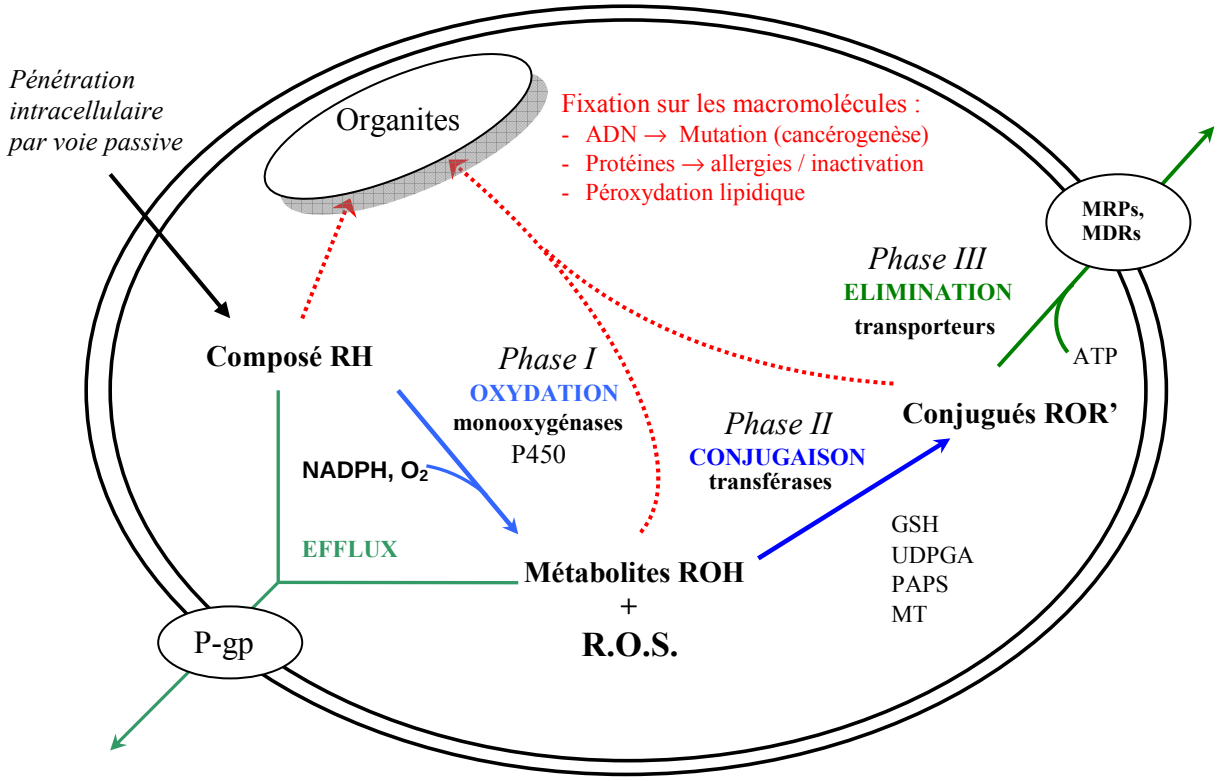


Figure 1-25 Schéma simplifié du métabolisme des composés hydrophobes (Source : Cours de M. Delaforge) La couleur froide représente la voie « normale » du métabolisme, en couleur chaude, lorsqu'il y a des phénomènes de toxicités. Le devenir d'un composé hydrophobe dans la cellule dépend de toutes ces voies de métabolisme. MDR : MultiDrug Resistance ; MRP :Multidrug Resistant Protein ; P-gp : P-glycoprotein

Dans ce processus de détoxication, les CYPs occupent une place prédominante : ils sont les enzymes majoritaire de la phase I (ils oxydent jusqu'à 90% des substrats), jouant ainsi un rôle pharmaco-toxicologique très important. La vitesse d'élimination des xénobiotiques de l'organisme est donc contrôlée par leur vitesse de transformation par les cytochromes P450. Ce métabolisme peut varier en fonction de facteurs génétiques liés au polymorphisme des CYPs par exemple, ou environnementaux tels que l'induction, l'inhibition. Ces variations constituent donc une cause possible d'inefficacité thérapeutique ou même de toxicité (Ingelman-Sundberg, 2002).

Tableau 1.9 Principales réactions de biotransfomation

Réactions	***Enzymes***	***Localisation***	***Substrats***
Réaction de Phase I			
Oxydations	Alcool déshydrogénases	Cytosol	Alcools
	Aldéhyde déshydrogénases	Cytosol	Aldéhydes
	Monoamine oxidases	Mitochondrie	Amines
	Flavine mono-oxygénases	Microsomes	Amines tertiaires
	Cytochrome P450	Microsomes	Variés
Réductions	**Cytochrome P450**	Microsomes	Variés
	Carbonyle réductases	Microsomes	Aldéhydes, Cétones
	Alcool déshydrogénases	Cytosol	Aldéhydes, Cétones
Hydrolyses	Estérases	Cytosol, Microsomes, Mitochondrie	Esters
Peptidases	Variés	Microsomes	Peptides
Réaction de Phase II			
Hydrolyses	Epoxyde hydrolases	Cytosol, Microsomes	Epoxydes
Glucuronidation	UDP Glucuronyl-transférases	Microsomes	Phénols, Thiols, Amines, Acides Carboxyliques
Conjugaison au gluthation	Glutathion-S-transférases	Microsomes	Electrophiles
Sulfatation	Sulfotransférases		Phénols, Thiols, Amines
Méthylation	O-,N-,S-méthyl-transférases	Cytosol, Microsomes	Phénols, Amines
Acétylation	N-acétyl-transférases	Cytosol	Amines
Conjugaison aux acides aminés	Aminoacyl-transférases	Microsomes	Acides Carboxyliques

Comme il a été vu précédemment, l' « organe principal » où s'opèrent les réactions de détoxication est le foie, là où les CYPs sont présents de façon majoritaire. Au sein de cet organe, la contribution relative des différents CYPs au métabolisme des médicaments est à peu près proportionnelle à leur abondance relative, excepté pour les CYPs de la famille 1A et le CYP2D6. Approximativement 75% des médicaments actuellement sur le marché ne sont en fait métabolisés que par ces trois CYPs hépatiques : (a) le CYP 3A4, (b) le CYP 2D6 et (c) le CYP 2C9 (Rendic, 2002). D'autres organes sont également le siège d'un métabolisme secondaire, moins important, car les P450s exprimés dans ces tissus contribuent non seulement à l'élimination, mais influencent également les concentrations tissulaires des agents thérapeutiques. L'intestin est un exemple de ces « organes annexes ». Il serait d'ailleurs le lieu principal du métabolisme extra-hépatique. Certains CYPs sont relativement plus présents dans l'intestin qu'ils ne le sont dans foie : c'est le cas du CYP 2J2 par

exemple. D'autres en revanche sont présents dans le foie, mais pas détectés de l'intestin, tels que les CYP 1A2, 2A6, 2B6, 2C8 et 2E1 (Paine et *al.*, 2006). Des études ont montré qu'en dépit d'une concentration en P450s moins faible dans l'intestin, le métabolisme intestinal pouvait contribuer de façon significative et parfois égale au métabolisme hépatique comme dans le cas de la cyclosporine, du midazolam ou du verpamil (Kolars et *al.*, 1991 ; Paine et *al.*, 1996 ; von Richter et *al.*, 2001). Les tissus du système respiratoire, à la fois exposés aux xénobiotiques inhalés et aux pathogènes transmis par le sang, sont une cible importante en toxicologie environnementale. De nombreux CYPs sont exprimés dans le poumon, dont certains spécifiques du tissu : CYP 2A13, 2S1 ou 4B1 (Ding et Kaminsky, 2003). Le poumon semble de plus être le lieu d'entrée dans l'organisme pour de nombreux procarcinogènes, comme les hydrocarbures aromatiques présents dans la fumée de cigarette, métabolisés par les CYP 1A ou 2A.

1.6.2 Variation inter-individuelle et manifestations pathologiques

L'activité des P450s dans le tissu hépatique est fortement dépendante de deux paramètres, l'un génétique (polymorphisme) et l'autre exogène (induction et inhibition). Cette variabilité peut être la cause d'une accumulation trop importante ou d'une élimination trop rapide du médicament, et par voie de conséquence entraîner les effets indésirables et/ou toxiques.

1.6.2.1 Polymorphisme

Les gènes codant pour les P450s impliqués dans le métabolisme des xénobiotiques peuvent avoir des mutations allèliques qui sont appelées *polymorphismes* si leur fréquence est supérieure à 1% de la population. Ces variations de gènes peuvent conduire à des variations d'expression du CYP correspondant, ainsi qu'à des variations de leur activité catalytique. Ainsi de nombreux polymorphismes de P450s définissent deux phénotypes appelés « métaboliseur lent » et « métaboliseur rapide » pour un médicament donné. Les métaboliseurs rapides éliminent plus vite le médicament et diminuent son efficacité thérapeutique, tandis que les métaboliseurs lents accumulent le médicament qui peut devenir toxique par effet de surdosage. La communauté scientifique et médicale est maintenant bien consciente qu'une thérapie rationnelle peut émerger en prenant en considération ces variations de métabolisme inter-individuelles (Evans et Relling, 1999).

1.6.2.2 Cancérogenèse

La part des cancers imputables aux xénobiotiques (polluants ou constituants « normaux » de l'alimentation) n'est pas bien définie, mais reste non négligeable. Chez l'homme, des corrélations entre les polymorphismes génétiques de certaines isoformes et les risques de cancer suggèrent que les

CYPs joueraient un rôle dans les processus de cancérogenèse. Les isoformes les plus étudiées et impliquées dans ce mécanisme sont les CYP 1A1, 1A2, 4B1 et 2E1. Chez l'homme, une forte activité du CYP 1A1 semble être associée à la survenue de cancers du poumon. L'activité du CYP 2E1 serait impliquée dans le développement de stéatohépatites *via* la production d'un stress oxydant (Weltman et *al.*, 1996). Une étude montre d'ailleurs que l'induction du CYP 2E1 par la N-nitrosodiméthylamine serait associée à l'apparition d'hépatocarcinomes (Tsutsumi et *al.*, 1993). Le CYP 2A6 est responsable de l'activation métabolique de la N–nitrytosamine (contenue dans la fumée de tabac), en métabolite cancérigène pouvant provoquer un cancer du poumon (Kamataki et *al.*, 1999).

1.6.2.3 Malformations congénitales

Le thalidomide et la phénytoïne sont des molécules thérapeutiques tératogènes chez l'homme et sont métabolisés respectivement par les CYP 2D6 et CYP 2C19. La toxicité découlant de l'activité des CYPs peut être un problème lors de l'organogénèse. En effet, la capacité de métabolisation du fœtus est non négligeable. Le CYP 1A1 est exprimé dès le stade de l'organogénèse, avant huit semaines et le CYP 1A2 après quatre mois (Oesterheld, 1998). L'activité *in utero* de ces enzymes pourraient être associée à des malformations et des fausses couches (Hakkola et *al.*, 1998).

1.6.2.4 Réactions immunoallergiques

Parallèlement aux mécanismes de toxicité directe, certains métabolites activés par les CYPs peuvent se révéler toxiques par des mécanismes immunoallergiques (Pessayre, 1995). L'hapténisation des macromolécules du soi par les métabolites réactifs peut stimuler le système immunitaire chez certains individus conduisant à une réponse inadaptée de type auto-immune. Les protéines du soi modifiées (CYPs en particulier) sont considérées comme étrangères au système immunitaire et deviennent alors les cibles d'auto-anticorps. Pour que ceux-ci puissent participer à la destruction immunologique des hépatocytes, les haptènes seront transportés par voie vésiculaire jusqu'à la membrane plasmique. Il en résulte une cytolyse (Pessayre, 1995). Il a été démontré que le foie était principalement touché par ces phénomènes. Au cours de formes sévères d'hépatites immunoallergiques, consécutives à l'absorption de médicaments tels que l'acide tiélinique, l'halothane ou la dihydrlazine, des anticorps dirigés contre les CYPs ont été retrouvés. Il est intéressant de noter que pour ces trois médicaments, les CYPs impliqués dans le métabolisme des xénobiotiques (CYP 2C9 pour l'acide tiénilique, CYP 1A2 pour la dihydralazine et CYP 2E1 pour l'halothane) sont les cibles de ces auto-anticorps.

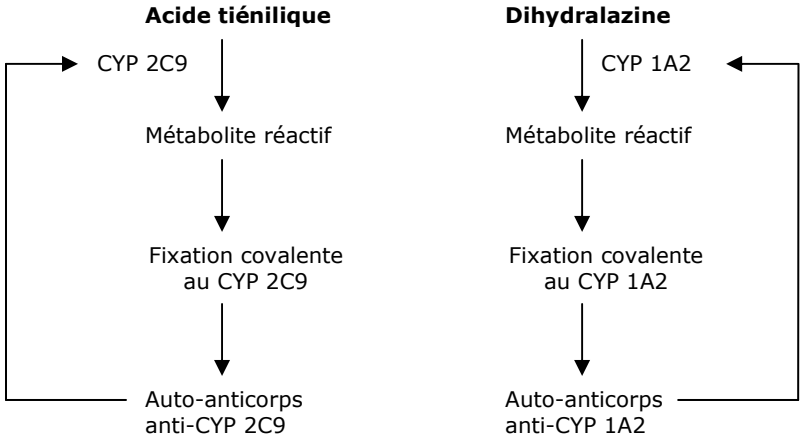


Figure 1-26 Mécanismes de mise en place de maladie auto-imunes après administration de l'acide tiénilique, de la dihydralazine.

1.6.2.5 Interactions médicamenteuses

L'administration concomitante de deux ou plusieurs médicaments peut avoir des conséquences néfastes sur leur pharmacocinétique. Ces variations mettent en jeu les mécanismes d'induction, de répression et d'inhibition. Pour les mécanismes d'inhibition, on distingue les inhibitions de types réversibles – où l'inhibiteur vient se fixer au niveau du site actif en compétition avec le substrat – et les inhibitions de types irréversibles – où ces inhibiteurs oxydés par les P450 restent fixés de façon covalente dans le site actif. On parle alors d'*inhibiteur suicide* –. Un grand nombre d'interactions médicamenteuses ont été mises en évidence. Elles peuvent survenir au cours de chaque étape pharmacocinétique (absorption, distribution, métabolisation, élimination) et plus particulièrement lors de la métabolisation. Au cours de cette phase, tous les CYPs sont impliqués, mais surtout le **CYP 3A4** à qui on doit par exemple le retrait de la terfénadine, anti-histamique qui en présence de kétoconazole (antiparasitaire) ou d'érythromycine (antibiotique) entraînait des complications cardiaques importantes. Des études ont montré que la présence de kétoconazole, inhibiteur direct du CYP3A4, diminuait fortement le métabolisme de la terfénadine, ce qui avait pour effet d'augmenter sa concentration plasmatique et de provoquer des toxicités cardiaques et vasculaires (Wang et *al.*, 2002). Cet exemple illustre bien les conséquences parfois dramatiques des interactions médicamenteuses. On voit donc que l'efficacité ou la toxicité d'un médicament métabolisé par un CYP est affectée par la prise d'un autre médicament. Néanmoins, dans certains cas, les interactions médicamenteuses peuvent être mises à profit en thérapeutique. Le saquinavir est une antiprotéase dont l'efficacité est limitée, *in vivo*, par sa faible biodisponibilité. La prise concomitante de jus de pamplemousse ou de ritonavir augmente nettement sa biodisponibilité et son efficacité (Kupferschmidt et *al.*, 1998). Au contraire, la prise de

certains médicaments est déconseillée avec du jus de pamplemousse. C'est le cas du cisapride (prokinétique) qui après absorption de jus de pamplemousse provoque des torsades de pointe (Morawiecka, 2000). La cyclosporine A est un immunodépresseur très utilisé en thérapeutique, mais aussi très coûteux. Il a été démontré que l'administration simultanée de kétoconazole et de cyclosporine A doublerait la biodisponibilité de la cyclosporine A et ainsi réduisant les doses de 60% à 80 % pour obtenir l'immunosuppression (Gomez et *al.*, 1995).

Le Tableau 1.10 montre plusieurs types d'interactions médicamenteuses faisant intervenir les mécanismes d'induction, de répression et d'inhibition. En pratique clinique, les interactions médicamenteuses mettant en jeu des mécanismes d'inhibition sont plus fréquentes pour celles impliquant des mécanismes d'induction et de répression. Les effets de nombreux médicaments sur l'expression des CYPs restent inconnus. Il est important de les identifier afin de pouvoir prédire les éventuelles interactions médicamenteuses.

1.7 Conclusion

Tout au long de ce chapitre, la nature, la diversité et la complexité des CYPs ont été soulignées. Leurs implications dans le système de détoxication et les conséquences aussi bien bénéfiques que pathologiques ont été également évoquées. Ces informations justifient bien l'intérêt que représente cette superfamille auprès des industries pharmacologiques depuis sa découverte. Jusqu'à présent, la plupart des études portées sur cette superfamille a été réalisée expérimentalement. Parmi elles, l'obtention de cristaux de P450s fut une avancé importante dans la connaissance de ces enzymes : la structure des protéines donnent en effet beaucoup d'information sur leurs fonctionnements. Pourtant, comprendre les mécanismes de reconnaissance des CYPs, en se basant uniquement sur la structure, n'est certainement pas une tâche aisée. C'est pourquoi on sollicite aujourd'hui l'utilisation d'outils informatiques pour traiter et faire une synthèse des informations disponibles sur ces protéines.

Ces outils sont en effet très pratiques, pouvant être utilisés dans la prédiction (modélisation de CYPs de structure inconnue lorsque cela est réalisable), pour proposer des réponses à de nombreuses hypothèses formulées sur le fonctionnement (expérience de docking) ou encore, pour suggérer de nouvelles expériences à réaliser (mutation dirigée). En pharmacologie, les industriels utilisent classiquement ces structures (et/ou modèles) de P450 pour tester leurs molécules chimiques, afin d'en déterminer des pharmacophore (support pour le design de nouveaux médicaments).

Tableau 1.10 Quelques exemples d'interactions médicamenteuses résultant d'une perturbation du métabolisme

Interaction	***Médicament responsable***	***Médicament affecté***	***Effet***	***Conséquences cliniques***	***CYP***	***Références bibliographiques***
Induction	Sécobarbital	Warfarine	–	Pas d'effet anticoagulant : risque de maladie thrombo-embolique	CYP 2C9	Lin et *al.*, 1998
	Ethanol (prise chronique)	paracétamol	–	Risque d'hépatotoxicité	CYP 2E1	Slattery et *al.*, 1996
	Rifampicine Griséofulvine	Contraceptifs oraux	–	Risque de grossesse	CYP 3A4	Barditch–Crovo et *al.*, 1999 ; Weaver et Glasier, 1999
	Rifampicine	Cyclosporine A	–	Risque de rejet de greffe	CYP 3A4	Offermann et *al.*, 1985
	Phénytoïne	Paracétamol	–	Hépatotoxicité	CYP 3A4	Brackett et Bloch , 2000
Répression	IFNα, IFNβ	Théophyline	–	Tachycardie, convulsion	CYP 1A2	ISrael et *al.*, 1993 ; Wills, 1990
Inhibition	Fluconazole	AINS (Ibuprofène, piroxicam, diclofénac)	+	Hémorragies, perforations gastro-intestinales	CYP 2C9	Bertz et Granneman , 1997
	Oméprazole, Lansoprasole	Diazépam	+	Somnolence, coma, dépression respiratoire	CYP 2C19	Gerson et Triadafilopoulos, 2001
	Cimétidine	Imipramine	+	Effet atropiniques, toxicité cartio-vasculaire	CYP 2D6	Bertz et Granneman, 1997
	Kétoconazole, Itraconazole	Triazolam	+	Somnolence, coma, dépression respiratoire	CYP 3A4	Vahre et *al.*, 1994
	Erythromycine, Kéthoconazole	Cisapride, Térfénadine	+	Torsade de pointe	CYP 3A4	Dresser et *al.*, 2000
	Vérapamil, Diltiazem, Erythromycine	Simvastatine, Lovastatine	+	Rhabdomyolyse	CYP 3A4	Dresser et *al.*, 2000 ; Mopusser et *al.*, 2000
	Itraconazole	Félodipine	+	Hypotension, bradycardie	CYP 3A4	Dresser et *al.*, 2000
	Ritonavir	Ergotamine	+	Ergotisme	CYP 3A4	Dresser et *al.*, 2000
	Kétoconazole, Erythromycine	Cycolsporine A	+	Néphrotoxicité	CYP 3A4	Dresser et *al.*, 2000
	Quinine	Codéine	+	Diminution de l'effet antalgique	CYP 2D6	Sindrup et *al.*, 1996

AINS : Anti-inflammatoires Non Stéroïdens
– : Diminution ou perte de l'efficacité
+ : augmentation de la toxicité et/ou de la sévérité des effets secondaires

Ainsi, l'ère du numérique ouvre réellement de nouvelles perspectives dans d'études de ces CYPs. Dans le prochain chapitre, nous verrons comme exploiter les structures de P450s existantes pour la construction de modèles (lorsque cela est possible), et également comment observer et comprendre les mécanismes mis en jeu dans la reconnaissance substrat–protéine au travers de simulation. Bref, le prochain chapitre sera dédié à la présentation des différents outils bioinformatiques déjà disponibles à l'étude d'une protéine.

CHAPITRE 2

Les outils bioinformatiques : vers de nouvelles solutions

« Donnez moi un point fixe et un levier et je soulèverai la Terre »

Archimède (287-212 av JC)

2.1 Brève présentation de la discipline et objectifs

La bioinformatique est un domaine de recherche qui a émergé depuis peu. L'augmentation vertigineuse du nombre de données générées en biologie associée au développement de l'informatique a peu à peu mis en lumière cette nouvelle discipline. La bioinformatique comprend un ensemble de concepts et de techniques nécessaires à l'interprétation de l'information génétique (séquences et expressions) et structurale (repliement 3D et interactions). C'est en quelque sorte le décryptage de la « bio-information ». La bioinformatique est donc une branche théorique de la biologie. Son but est d'effectuer la synthèse de données disponibles (à l'aide de modèles et de théories), d'énoncer des hypothèses généralisatrices (par exemple, comment les protéines se replient ou comment les gènes interagissent) et de formuler des prédictions (par exemple dans notre cas, prédire un mécanisme d'interaction).

Nous allons montrer dans cette thèse comment les outils de la bioinformatique peuvent permettre l'étude des mécanismes sous-jacents des P450s. Les méthodes développées en bioinformatique sont nombreuses et variées : il n'est donc pas question de faire un inventaire de toutes les méthodes existantes, mais plutôt de présenter les différents méthodes ou données qui sont en rapport avec l'objectif qu'on s'est fixé ici. Un plan d'expérience à suivre est ici proposé pour servir de fil conducteur à ce chapitre (cf. Figure 2-1). Ce plan d'expérience consiste en des procédures, des étapes à réaliser pas à pas, analysant des données simples (par exemple les séquences) pour aboutir à un modèle (tel le mécanisme d'action de la protéine). Le modèle peut être le support pour des simulations afin de « mimer » la réalité (par exemple, simulations de dynamique moléculaire en présence ou non d'un substrat). En multipliant les expériences et en modifiant légèrement le support, on espère à terme avoir des idées sur le fonctionnement de la protéine. C'est le principe de la **modélisation moléculaire**.

Dans ce chapitre, nous verrons où et comment chercher l'information nécessaire à notre étude, comment obtenir un modèle (et les principes/méthodes impliqués pour le fabriquer), comment modifier ce modèle et le rendre plus « proche » de la réalité, et comment procéder à des simulations pour observer son comportement dans le temps. Chaque paragraphe de ce chapitre correspond à une des différentes étapes illustrées sur la Figure 2-1.

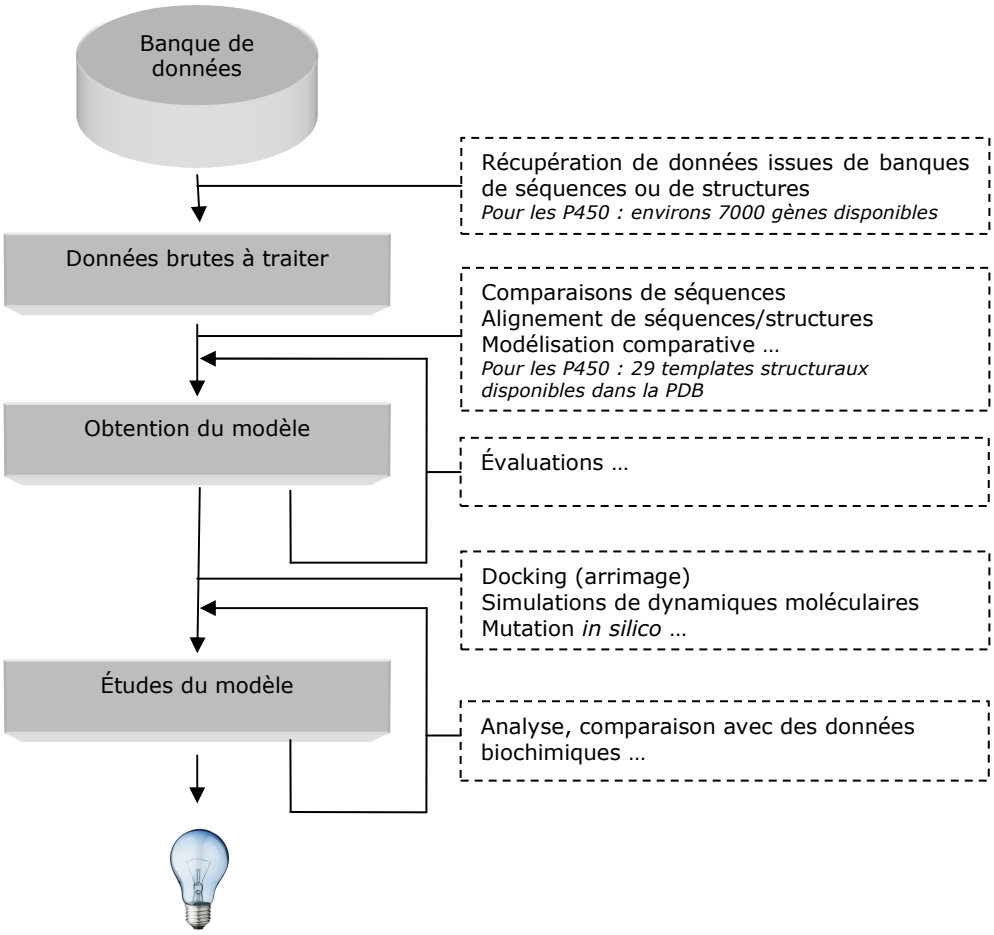


Figure 2-1 Plan d'expérience pour la modélisation moléculaire en générale (correspondant au plan du chapitre).

2.2 Les protéines : description géométrique du squelette peptidique

Dans un premier temps, nous souhaitons aborder ce chapitre par des rappels de notions géométriques et structurales des protéines, sur lesquelles reposent les concepts des logiciels bioinformatiques qui ont été utilisés au cours de la thèse ou qui ont été développés.

2.2.1 Les coordonnées cartésiennes

La description la plus simple du squelette protéique est évidemment la liste des coordonnées cartésiennes de ses atomes ou de ses C_α (carbone central des acides aminés, eux-mêmes composants des protéines, cf. Annexes). Cette description est celle sur laquelle s'appuie la mesure de similarité

structural appelée RMSD (Root Mean Square Deviation) entre les structures superposées (voir section 2.4.3.2).

2.2.2 Les distances internes

La liste des distances internes entre atomes ou C_α du squelette peptidique est un moyen de décrire une conformation de manière indépendante de l'origine des coordonnées. Les distances internes d'une sous-structure donnent donc une information « locale » de la conformation. Celle-ci permet évidemment les comparaisons sans avoir à superposer les coordonnées cartésiennes des structures concernées.

2.2.3 Les angles Phi, psi et Omega

Comme la liaison peptidique *–CO–NH–* est considérée comme plane, le squelette peptidique peut être décrit par les angles ϕ,ψ et ω qui sont les angles dièdres de la chaîne respectivement autour des liaisons *N–Cα*, *Cα–C* et *C–N* (cf. Figure 2-2). L'angle ω est en général *trans* (180°), mais dans les prolines il peut être *cis* dans 6 à 10% des cas. De ce fait, la description du repliement ne prend en compte généralement que le couple (ϕ,ψ) (MacArthur et Thornton, 1996). De plus, le couple ϕ et ψ ne peut pas prendre n'importe quelle combinaison de valeurs (Ramachandran et *al.*, 1963). En outre, les angles ψ et ω sont très corrélés (Esposito et *al.*, 2005).

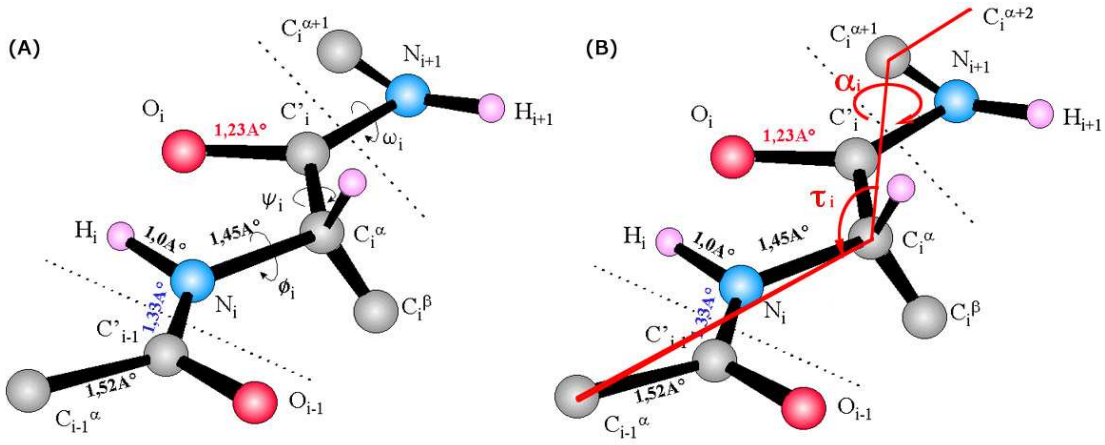


Figure 2-2 Angles ϕ,ψ (A) et angles α,τ (B)

2.2.4 Les angles *Alpha* et *Tau*

L'angle α est l'angle dièdre défini par quatre C_α successifs et l'angle τ est l'angle entre les plans définis par trois C_α successifs (cf. Figure 2-3). L'angle α_i est associé au deuxième des quatre C_α (le $C_{\alpha i}$) et l'angle τ_i est celui formé par les trois premiers C_α.

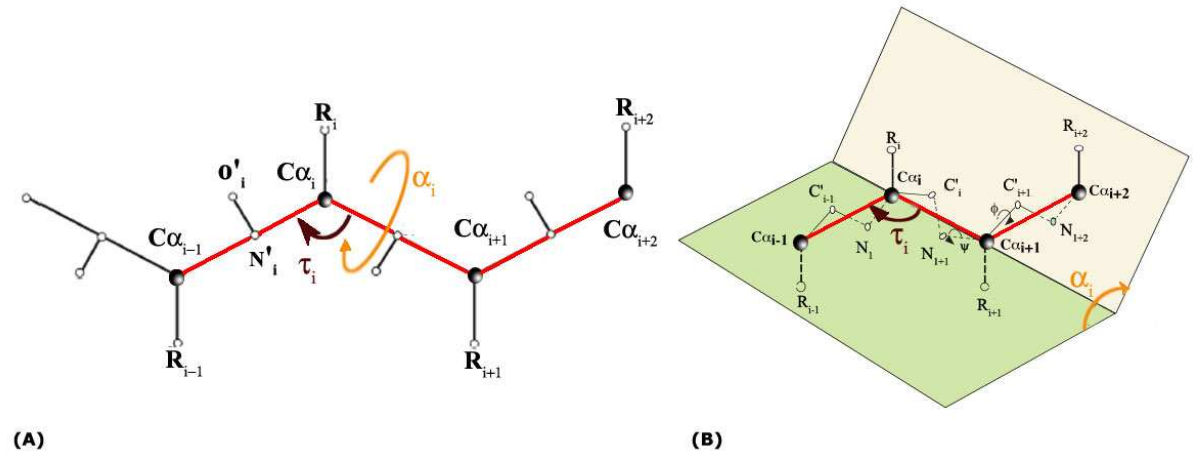


Figure 2-3 Angles Alpha et Tau décrit par Levitt. L'angle α est l'angle dièdre défini par les quatre C_α successifs. L'angle τ est l'angle entre les plans définis par trois C_α successifs. La figure (A) correspond à une simplification de la structure protéique où les groupes d'atomes C, N et H ont été combinés en un atome N' (centres d'interaction) et les atomes O en O' (figure tirées de Levitt, 1976).

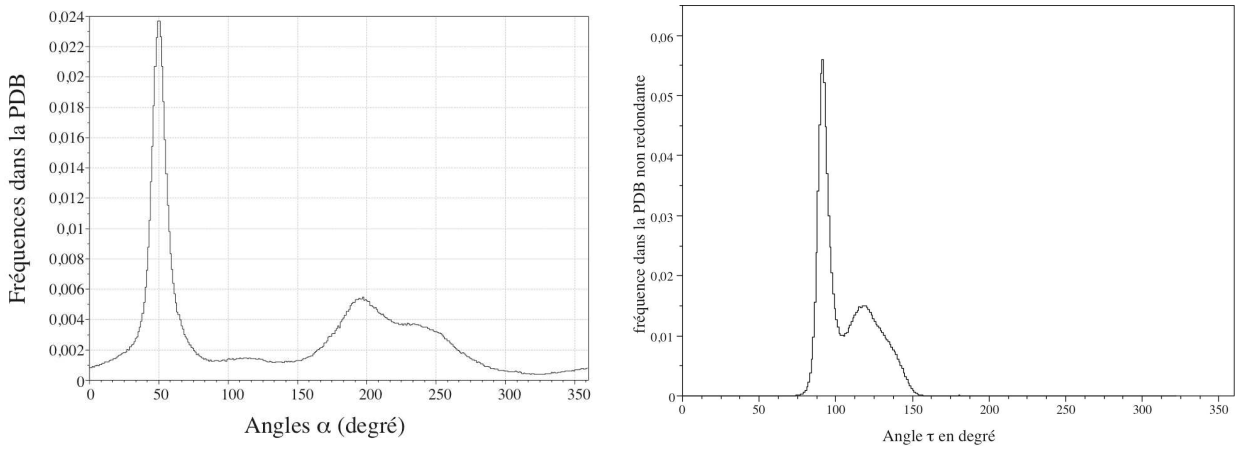
Les angles α et τ ont été décrits par Levitt et *al.* (Levitt et Warshel, 1975 ; Levitt, 1976). Toutefois, ils ont été utilisés précédemment par Flory lors de son étude de la conformation de polypeptides en pelotes statiques (*random-coil*) (Flory, 1969). Les liaisons N– C_α et $C_{\alpha+1}$–C sont parallèles à ± 10° et le groupe peptidique (–$C_\alpha HNH_2$–$COOH$) est plan, il est donc possible de remplacer les angles ϕ et ψ par une seule valeur, l'angle α dont la relation avec ϕ et ψ est la suivante (Levitt, 1976) :

$$\alpha_i = 180° + \phi_{i+1} + \psi_i + 20°(\sin\phi_i + \sin\phi_{i+1})$$

Comme les axes de liaisons N– C_α et $C_{\alpha+1}$–C sont parallèles mais non colinéaires, l'angle τ, formé par 3 C_α successifs, varie avec l'angle α selon la relation suivante :

$$\tau_i = 106° + 13°\cos(\alpha_i - 45°)$$

L'angle τ varie donc peu autour de 106°. Comme celle des angles (ϕ,ψ), la distribution des angles α n'est pas uniforme (Oldfield et Hubbard, 1994). Les éléments de structures secondaires sont décrits par une suite répétitive d'angles α : une hélice α est une suite d'angles α proches de 50° tandis qu'un brin β est une série d'angles α fluctuant autour de 200° (entre 180° et 240°, Figure 2-4).



(A) (B)

Figure 2-4 Distribution des angles α dans la PDB (A) et τ dans la PDB non redondante (B) (source thèse de M. Carpentier, 2005)

2.3 Banques de séquences et de structures : les données sources

Nous décrirons dans ce paragraphe deux catégories de banques de données qui ont été utilisées dans notre « plan d'expérience » (adapté aux P450s) : les banques de séquences et les banques de structures de protéines.

2.3.1 Banque de séquences biologiques

Les premières banques de séquences sont apparues au début des années 80, sur l'initiative de plusieurs équipes. Très rapidement, avec l'augmentation de l'efficacité du séquençage, la collecte et la gestion des données ont nécessité une organisation plus conséquente. C'est ainsi que se sont apparues les premières banques de séquences nucléiques telles que l'**EMBL** pour l'Europe – base financée par l'EMBO (European Molecular Biology Organisation) et maintenue par une équipe située actuellement à Cambridge au sein de l'EBI (Europeean Bioinformatics Institue) – ou encore **GenBank** pour les États-Unis – financée par le NIH (National Institue of Health) diffusé par le NCBI (National Center of Biotechnology Information) –. Une collaboration entre ces deux banques a été initiée relativement tôt, s'est étendue en 1987 avec la participation de la DDBJ (DNA Data Bank) du Japon, pour donner naissance en 1990 à un format unique de description des séquences dans les banques de données nucléiques (Le format de ce fichier peut être consulté sur le serveur de l'EMBL à l'adresse suivante : http://www.ebi.ac.uk/embl/Documentation/FT_definitions/feature_table.html)

Parallèlement, deux banques principales de séquences protéiques ont été créées. La première se nomme la Protein Identification Ressource (**PIR-NBRF**) (George et *al.*, 1986), fruit de l'association de données issues du MIPS (Martinsried Institute for Protein Sequences), de la base japonaise JIPID (Japan International Protein Information Database) et des données de la NBRF (National Biomedical Research Fondation). La deuxième, gérée par l'EBI et le SIR (Swiss Institute of Bioinformatics), se nomme la spTrEMBL et a été constituée à Genève à partir de 1986. Elle comprend une banque de données soigneusement annotée, la **Swiss-Prot** (Bairoch et Boeckmann, 1991), et une banque plus large annotée automatiquement, la **TrEMBL** (Translated EMBL nucleotide sequence library) (Apweiler et *al.*, 1997). Aujourd'hui, l'EBI, le SIR et le PIR ont uni leurs forces pour créer la base de données **UNIPROT** (Apweiler et *al.*, 2004). La version non redondante de cette base de données, UniRef100, contient environ 5 millions et demi de séquences. Si on analyse un extrait soigneusement annoté de cette banque, UniProtKB Swiss-Prot, on constate qu'environ la moitié des séquences sont d'origine bactérienne (cf. Figure 2-5)

À noter enfin que cette richesse en information est due à la présence de génomes entièrement séquencés. Au début, il s'agissait de petits organismes comme le bactériophage Lambda, de 40 kb (Sanger et *al.*, 1982) ou encore la bactérie *Haemophilus influenzae* de 1,83 Mb (Fleischmann et *al.*, 1995), premier organisme vivant dont le génome a été entièrement séquencé. Progressivement les séquences complètes de génomes de plus en plus grands sont arrivées avec en point d'orgue en 2001, la séquence du génome humain 3400 Mb (Lander et *al.*, 2001 ; Venter et *al.*, 2001).

Tableau 2.1 Statistiues de la base de données UniProt (release 12.0 du 24-Jui-2007)

Database	***Entries***
UniProtKB	4 949 164
UniProtKB/Swiss-Prot section	276 256
UniProtKB/TrEMBL section	4 672 908
UniRef100	4 910 948
UniRef90	3 145 989
UniRef50	1 555 946
UniParc	15 345 639

UniprotKB comprend l'ensemble de séquences annotées, les UniRef sont des bases de données non redondantes et UniParc correspond aux archives. Dans les banques UniRef90 et UniRef50, aucune paire de séquence dans les bases respectives n'a une identité mutuelle en séquence de > 90% ou > 50%. La base UniRef100 les séquences identiques et les fragments de ces séquences comme une même entrée, avec des liens vers l'ID de la protéine, la séquence, la bibliographie associée, etc.

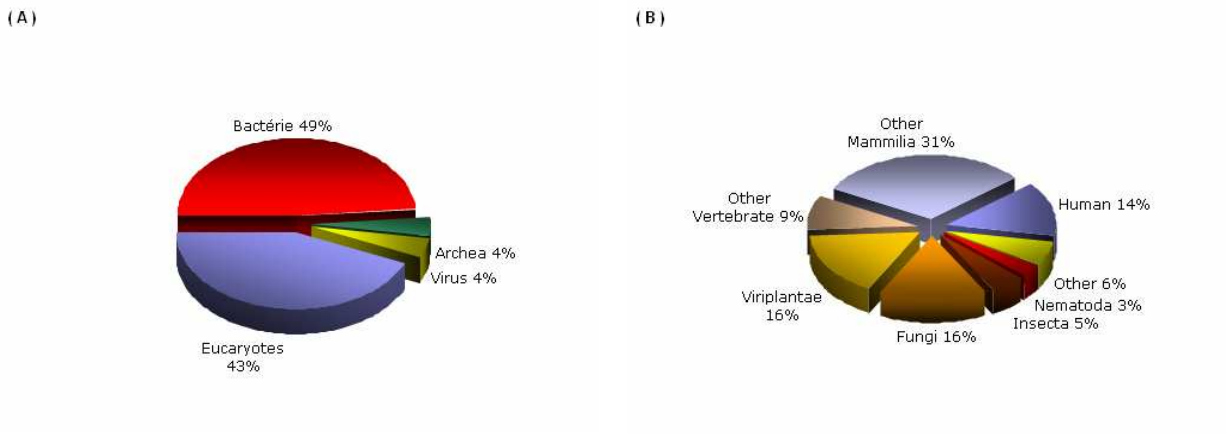


Figure 2-5 Distribution taxonomique des 283 454 séquences (A) et des 122 974 séquences eucaryotes (B) de la version 54.2 du 11-sept-07 d'UniProtKb/Swiss-Prot

Pour ce travail de thèse, les séquences de P450s utilisées proviennent essentiellement des bases de données Swiss-Prot et TrEMBL. Dans ces dernières, les cytochromes P450s représentent environ plus de 6481 entrées dans l'UniProtKB release 12.2 ($Swiss\text{-}Prot_{(909)}$ et $TrEmbl_{(5572)}$).

2.3.2 Banque de structures tridimensionnelle de protéines

Les structures de protéines sont généralement toutes répertoriées dans la Protein Data Bank (PDB) (http://www.rcsb.org/pdb). Ce qui a été précédemment dit (dans le Chapitre 1) pour l'évolution d'apparition des structures de P450 reflète l'évolution de la PDB en général. Cette banque connaît une croissance impressionnante depuis sa création. Toutefois, cette banque est extrêmement redondante, beaucoup de structures provenant de protéines très proches (grand nombre de mutants, de co-cristaux avec des substrats variés). À titre d'exemple, plus de 600 structures de lysozyme sont présentes dans la PDB. Les statistiques de la banque montre néanmoins que le nombre de structures 3D déterminées est encore bien faible comparé au nombre de séquences disponibles.

Tableau 2.2 Dénombrement des structures déposées dans la PDB au 15 septembre 2007

Exp. method	***Molecule type***				
	Proteins	***Nucleic Acids***	***Proteins/NA complexes***	***other***	***Total***
X-Ray	36223	983	1684	24	**38914**
NMR	5665	781	134	7	**6587**
Electon Microscopy	105	10	38	0	**153**
Other	80	4	4	2	**90**
Total	**42073**	**1778**	**1860**	**33**	**45744**

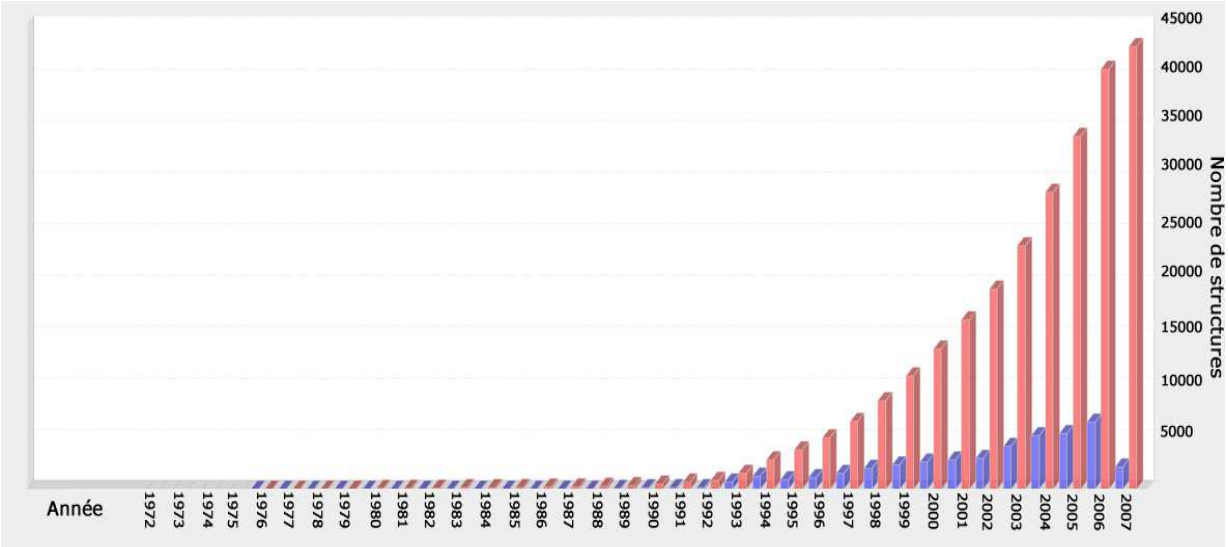


Figure 2-6 Croissance annuelle du nombre de structures déposées dans la PDB. En bleu, est représentée le nombre de structures annuelles et en rouge le nombre cumulé des structures. (Source : http://www.rcsb.org/)

Un site dérivé de la PDB, PDBsum (http://www.ebi.ac.uk/thornton-srv/databases/pdbsum) a été mis en ligne pour une navigation plus agréable dans les structures protéiques et l'information associée aux structures.

Dans la PDB, l'ensemble des structures de P450s a été déterminé par cristallographie RX, (voir Figure 1-13 et Figure 1-14 page 39). On peut remarquer que le nombre de structures résolues de P450 suit la tendance fortement croissante de la PDB elle-même.

2.3.3 Banque de classification de structures protéiques

Il existe des banques pour identifier et classer – soit de façon hiérarchique soit de façon non hiérarchique – les familles de repliements déjà connus. Ces banques utilisent plusieurs approches soit basées sur des calculs de « distance » entre repliements, soit essentiellement construites sur un examen visuel des structures 3D disponibles.

2.3.3.1 FSSP

La base de données FSSP de L. Holm et C. Sander (Holm et *al.*, 1992) a été construite à partir d'un algorithme de mesure quantitative de la compacité des structures. Ce critère permet de diviser une structure en domaines de plus en plus petits. La récurrence des domaines obtenus est ensuite analysée afin de connaître la taille des domaines pouvant être retrouvés en grand nombre dans des protéines différentes. C'est le serveur DALI (Holm et Sander, 1997) qui permet d'interroger la base

FSSP. Ce serveur permet non seulement de consulter la classification la plus récente de la PDB90 qui contient les structures présentant entre elles moins de 90 % d'identité en séquence, mais aussi de rechercher les structures les plus proches d'une structure que l'on soumet.

Lorsqu'une structure de P450 est soumise à ce serveur par exemple, un alignement structural de structures proches de celle de la requête est alors affiché, comprenant des références et des liens vers l'accession PDB de chaque structure.

2.3.3.2 SCOP

SCOP (Structural Classification of Protein) (Murzin et *al.*, 1995) découpe les protéines en domaines (régions ayant un cœur hydrophobe et peu d'interaction avec les autres protéines) pour les classer. La classification de SCOP s'effectue sur quatre niveaux, du plus général au plus fin :

1. *class* : la composition en structures secondaires est similaire (tout α ; tout β ; α et β mêlés ; α et β organisés en deux régions séparées) ;
2. *fold* : la composition en structures secondaires (hélices α et feuillets β), leur arrangement spatial et leurs connexions topologiques sont similaires ;
3. *superfamily* : l'identité de séquence peut être faible mais les structures et les fonctions suggèrent une origine évolutive commune ;
4. *family* : les structures protéiques ont au moins 30% d'identité de séquence ou bien possèdent des fonctions et des structures très similaires

Ainsi, la banque SCOP classifie « manuellement » selon les organisations hiérarchiques les 26000 structures de la version Oct2004 de la PDB en 945 repliements, 1539 super familles et 2845 familles. Dans cette banque, les CYPs sont retrouvés dans la classe des « ***tout α*** » et le repliement des « ***Cytochrome P450*** ». Ce dernier ne contient qu'une seule superfamille, celle des Cytochrome P450s, elle-même ne contentant qu'une seule famille, celles des Cytochromes P450s. En résumé, la superfamille des P450s représente une seule et unique branche dans la hiérarchie.

2.3.3.3 CATH

CATH (Class, Architecture, Topology (fold family) and Homologous superfamily) (Orengo et *al.*, 1997) est aussi base de données construite automatiquement et vérifiée manuellement à partir de la PDB. Seuls les cristaux résolus à plus de 4 Å dans la PDB sont considérés dans CATH. La classification hiérarchique des structures proposées dans cette base de données est illustrée sur la Figure 2-7.

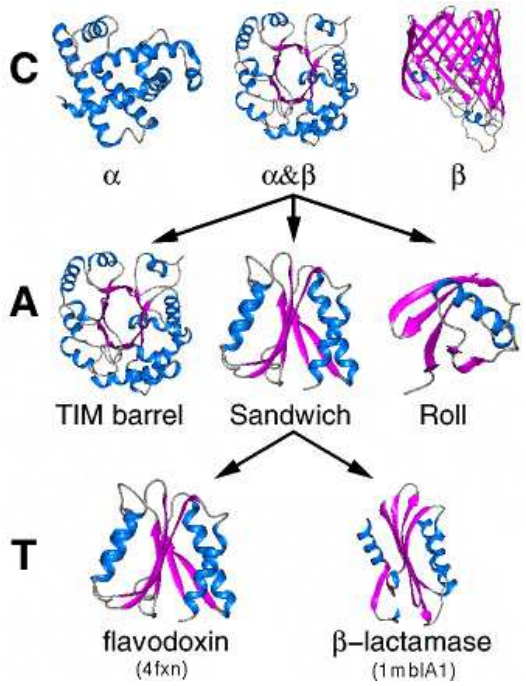
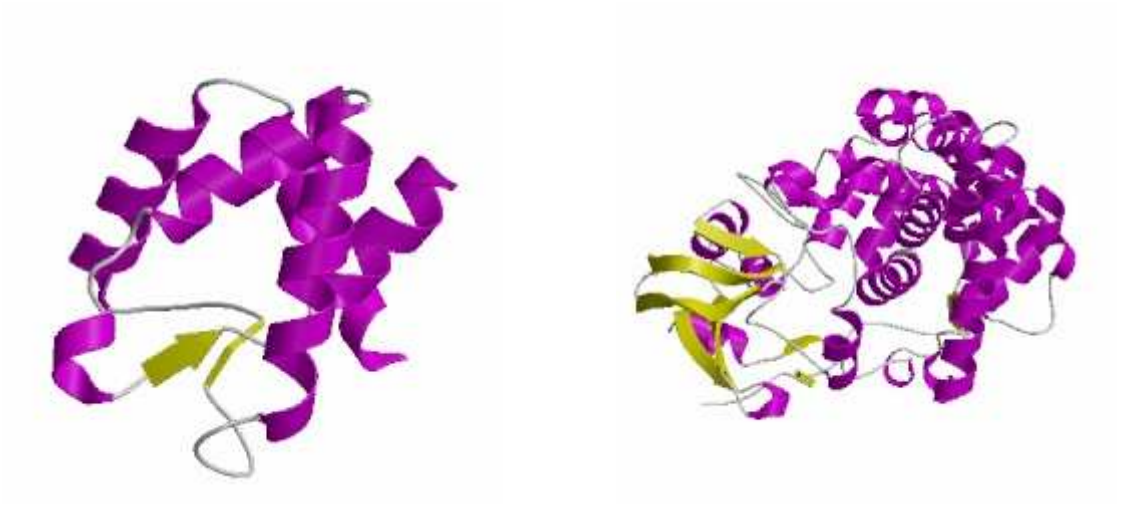


Figure 2-7 Classification des rempliements dans la base de données CATH (source : http://cathwww.biochem.ucl.ac.uk/cgi-bin/cath/GotoCath.pl?link=cath_info.html)

Cette base comporte également quatre niveaux principaux par familles structurales, et cinq autres niveaux supplémentaires établis sur les similarités des séquences protéiques. Ces niveaux de classification sont les suivants :

1. ***C****lass* : les structures sont regroupées selon leur composition en structures secondaires et les contacts entre celles-ci (4 classes : principalement α, principalement β, mixte, quelques structures secondaires).
2. ***A****rchitecture* : l'organisation générale des structures secondaires est la même pour les structures d'un même groupe.
3. ***T****opology* : les structures ayant un même repliement en termes de nombre, ordre et connexions de structures secondaires sont regroupées.
4. ***H****omologous superfamily* : les structures d'un même groupe ont des structures et des fonctions très similaires, suggérant un ancêtre commun.
5. *Niveaux supplémentaires* : S, O, L, I selon l'identité de séquences respectivement > 35%, > 60%, >95% et de 100% (ce dernier regroupe en fait les protéines qui ont été résolues plusieurs fois, par exemple complexées ou non avec un ligand). Le dernier niveau D sert vérifier l'unicité de chaque entrée.

Dans cette banque, et à l'image de SCOP, les P450s figurent dans la classe des « principalement α » (Class 1), dans l'architecture des « paquets orthogonaux » (Architecture 1.10), dans la topologie des « Cytochromes P450s » (Topology 1.10.630) et enfin dans les superfamilles homologues des « Cytochromes P450s » (Homologous superfamily 1.10.630.10). Les CYPs dans la nomenclature CATH peuvent descendre jusqu'aux sous-niveaux S, O, L I puis D. À noter qu'il existe une autre entrée au niveau topologique (niveau 3), relative aux P450s : Cytochrome P450-Terp, domain 2 (Topology 1.10.230). La différence topologique observée est présentée à la Figure 2-8. Le recherche de la structure 1cpt du $P450_{Terp}$ est pourtant retrouvée au niveau des « Cytochromes P450s » (Homologous superfamily 1.10.630.10) : on peut donc se demander d'où peut provenir cette autre entrée relative aux P450s…



(A) (B)

Figure 2-8 Différence topologique observée dans CATH pour « Cytochrome P450s » et « Cytochrome P450-Terp, domain 2 »

2.3.4 Autres banques secondaires

Les banques de données de séquences telles que GenBank/EMBL/DDBJ pour les acides nucléiques, Swiss-Prot, TrEMBL, UNIPROT pour les protéines et la PDB pour les structures, sont des banques de données généralistes : elles contiennent toutes les données sources suffisantes pour les travaux d'analyses et de traitements en bioinformatique. Devant la croissance quasi exponentielle des données et l'hétérogénéité des séquences contenues dans les principales bases de séquences généralistes, d'autres bases spécialisées sont apparues. Elles se sont constituées autour de thématiques biologiques ou tout simplement en vue de réunir les séquences d'une même espèce et d'en enrichir les annotations pour diminuer, ou lever les ambiguïtés laissées par les grandes banques publiques. A titre

d'exemple, on peut citer la banque d'alignement de structures HOMSTRAD, les banques de domaines de protéine PFAM, ProDom, ou SMART, les banques de régions ou de motifs conservés Blocks, PRINTS mais également des banques de motifs/signatures spécifiques comme Prosite.

2.3.4.1 Prosite

Prosite (Hulo et *al.*, 2006) est une banque originale et peut être considérée comme un dictionnaire qui recense des motifs (ou signatures) protéiques ayant une signification biologique. Elle est établie en regroupant, quand cela est possible, les protéines contenues dans Swiss-Prot par famille comme par exemple les kinases ou les protéases. On recherche ensuite, au sein de ces groupes, des motifs consensus susceptibles de les caractériser spécifiquement. La construction de cette base repose sur quatre critères essentiels : i) collecter le plus possible de motifs significatifs, ii) avoir des motifs hautement spécifiques pour caractériser au mieux une famille de protéines, iii) donner une documentation complète sur chacun des motifs répertoriés, et iv) faire une révision périodique des motifs pour s'assurer de leur validité par rapport aux dernières expérimentations. L'essentiel de l'expertise est basé sur un réseau de correspondants spécialistes des sujets traités. La base est organisée en deux parties. La première contient l'identification et la description de chaque motif. La deuxième contient l'information qui documente chaque motif (Bairoch, 1993 ; Bairoch et Bucher, 1994).

Dans le cas des P450s, la fiche documentaire a pour le numéro d'accession *PDOC00081*, et pour titre « *Cytochrome P450 cysteine heme–iron signature* ». Cette fiche répertorie les informations générales des P450s telles que leur description, quelques références ainsi que l'entrée du motif PROSITE des P450s proprement dite. Celle-ci dispose également d'un nom et d'un numéro d'accession (*PS00086*) ainsi que la règle associée pour détecter les P450s sur UniProtKB/Swiss-Prot et la PDB (grâce aux références croisées). Ce motif, correspond à peu près à la signature caractéristique des P450s vu dans le premier chapitre (Figure 1-4, page 16) :

[FW]-[SGNH]-x-[GD]-{F}-[RKHPT]-{P}-C-[LIVMFAP]-[GAD].

Dans ce motif de 10 résidus, les crochets « *[ ]* » correspondent aux résidus possibles à la position donnée, le caractère *x*, à n'importe quel résidu, et les accolades «*{ }* », à un résidu facultatif à une position donnée. Cette signature (Cys-Pocket) est retrouvée sur 782 séquences des 283 454 séquences d'UniProtKB/Swiss-Prot (version 54.2) ainsi que tous les P450s présent sur la PDB. À noter que parmi ces 782 séquences, seuls 753 correspondent réellement à des P450s et 29 correspondent à des protéines non P450s. Par ailleurs, 42 séquences de P450s ne présentent pas ce motif et ne sont pas détectées par Prosite (source : http://www.expasy.org/prosite/PS00086)

2.3.4.2 Blocks

La base BLOCKS (Henikoff et *al.*, 1999) est également basée sur un système qui détecte et assemble les régions conservées de protéines apparentées. La détection des blocs similaires de séquences est effectuée sur des alignements multiples de séquences. Un bloc ne contient pas d'insertion ni de délétion. L'ensemble de tous ces blocs forme la base. C'est ainsi que Henikoff et Henikoff (1991) ont défini 1764 blocs à partir des 437 groupes de protéines recensés durant l'établissement de PROSITE. Les motifs représentés par la base BLOCK sont généralement plus courts que ceux donnés par la base PROSITE mais les différences fondamentales entre ces deux banques résident dans la représentation des données. Les motifs de PROSITE sont définis sous forme de chaînes de caractères prenant en compte des insertions et des ambiguïtés sur les acides aminés conservés alors que les motifs de la base BLOCK sont représentés par blocs d'alignements multiples. Dans la banque de données *Blocks*, les blocks sont générés automatiquement à l'aide du programme PROTOMAT (Henikoff et Henikoff, 1991), à partir des entrées de la base InterPro et en utilisant des séquences issues de Swiss-Prot et TrEMBL, en association avec des références croisées à PROSITE et/ou PRINTS et /ou SMART et/ou PFAM et/ou ProDom (toutes ces bases se référencent entre elles).

2.3.4.3 Pfam

Pfam (Bateman et *al.*, 2004) est une banque de domaines de familles protéiques. Elle contient une large collection d'alignements de séquences multiples ainsi que les profils associés de modèles de Markov cachés (HMM pour Hidden Markov Model) qui permettent de retrouver ces domaines dans de nouvelles séquences. Pour chaque famille, Pfam fournit également une annotation fonctionnelle, les références bibliographiques ainsi que des liens vers d'autres banques de données. Chaque famille dans Pfam est représentée par deux alignements de séquences multiples : (i) un alignement dit *seed* (graine) contenant un petit nombre de membres de la famille et (ii) un alignement complet qui contient tous les membres qui peuvent être détectés dans la base. Tous les alignements sont effectués à partir de séquences provenant de la pfamseq (une banque construite à partir de la SwissProt et de la TrEmbl), un ensemble de protéines non redondantes issues de la Swiss-Prot et SP-TrEMBL. Le profil HMM est alors construit à partir de l'alignement *seed* en utilisant le package HMMER (http://hmmer.wustl.edu/) qui permet une recherche de nouvelles séquences sur cette même banque pfamseq. Selon un seuil défini, les séquences trouvées sont alors incorporées à l'alignement complet. À ce jour, dans la version 22.0 de Pfam (juillet 2007), on répertorie 9318 entrées (ou familles). L'entrée correspondante aux P450s est la *PF00067*. Il s'agit d'un domaine d'une vingtaine de résidus, placé coté C-terminal. Cette entrée est définie à partir d'un alignement *seed* comprenant 51 séquences, l'alignement complet quant à lui comporte 6906 séquences de P450s.

2.3.4.4 HOMSTRAD

HOMSTRAD –pour Homologous Structure Alignment Database– (Mizuguchi et *al.*, 1998) est une banque d'alignement de structures de familles homologues (1032 familles / 3454 structures alignées en 2005). L'homologie, ordinairement utilisé dans le sens d'origine évolutionnaire commune, est ici inférée par un degré d'identité suffisamment élevé. Les classifications de SCOP, Pfam, PROSITE et SMART sont combinées avec les résultats de recherche de similarités de PSI-BLAST et de FUGUE – une méthode de *threading* liée à HOMSTRAD – pour définir les familles, qui sont représentées de manière à montrer l'environnement structural local de chaque résidu. Les alignements structuraux sont réalisés automatiquement par les logiciels MNYFIT (Sutcliffe et *al*., 1987), STAMP (Russell & Barton, 1992) et COMPARER (Sali & Blundell, 1990; Zhu et *al*., 1992) puis vérifiés par une expertise humaine.

HOMSTRAD est la banque utilisée dans le logiciel SYBYL (Tripos Inc, St Luis, USA) pour construire des modèles par homologie.

2.3.5 Récupération des données de banques

La récupération d'information telle que la séquence ou la structure d'une protéine est très aisée dès lors que l'on dispose de son identité ou de son numéro d'accession. Elle se fait à l'aide d'une simple requête à partir des logiciels dédiés comme GCG (Devereux et *al.*, 1984) ou SRS (Etzold et Argos, 1993). En revanche, lorsqu'on désire récupérer les séquences « voisines » ou les structures proches de notre protéine d'intérêt, des techniques d'alignements sont nécessaires. Elles permettent de récupérer uniquement les séquences dont le score d'alignement excède un certain seuil. Un alignement repose sur une notion de ressemblance entre séquences qu'il convient définir. Ainsi, par exemple, on peut mesurer le pourcentage d'identité global entre deux séquences après alignement. On peut aussi utiliser le nombre et la qualité de sous-alignements locaux entre deux séquences. Les méthodes de recherches sont donc basées sur des alignements et des scores attribués à ces alignements.

Communément, pour récupérer les séquences « voisines » d'une protéine étudiée, les outils les plus utilisés sont **FASTA** (Pearson et Lipman, 1988), **BLAST** (Altschul et *al.*, 1990) ou **PSI-BLAST** (Altschul et *al.*, 1997), qui sont des heuristiques d'alignement. À noter également que chaque banque de données précédemment décrite dispose dans la plupart des cas de son propre « moteur » de recherche. Au cours de ma thèse, je me suis principalement servi de BLASTP et PSI-BLAST pour récupérer les séquences d'intérêt.

Il existe parallèlement des logiciels analogues à BLAST pour la recherche de structures « voisines » d'une protéine. Certains ont été abordés plus haut comme le serveur DALI de la base FSSP. Il existe également de nombreux outils comme par exemple le très récent logiciel **YAKUSA** (http://www.rpbs.jussieu.fr/Yakusa/) mis au point par M. Carpentier (Carpentier et *al.*, 2005) de l'Atelier de BioInformatique (ABI) avec lequel je travaille en étroite collaboration. Ce logiciel utilise le concept des angles α (coordonnées internes) pour les aligner les structures. Les angles α sont également utilisés par l'un des outils que j'ai utilisé pour comparer les structures de P450s (GOK).

2.4 Traitement des données sources

L'étape qui suit la récupération des données sources est celle de comparaison entre les données récupérées. Nous décrirons dans le paragraphe qui suit, les diverses méthodes d'alignements de séquences et de structures.

2.4.1 Comparaison de séquence à séquences : matrices et algorithmes

La recherche de similitude entre séquences est un élément fondamental qui constitue souvent la première étape des analyses de séquences. Elle permet de révéler des régions proches dans leur séquence primaire en considérant le minimum de changements en insertion, suppression, ou substitution qui séparent deux séquences. Cette méthode est largement utilisée dans les recherches de motifs sur une séquence, dans la caractérisation de régions communes ou similaires entre deux ou plusieurs séquences, dans la comparaison d'une séquence avec l'ensemble ou sous-ensemble des séquences d'une base de données, ou bien encore dans l'établissement d'un alignement multiple sur lequel sont basées les analyses d'évolution moléculaire. Pour aligner deux séquences entre elles, il est possible de chercher à optimiser le pourcentage de positions identiques dans l'alignement. Néanmoins, cette mesure ne tient pas compte de la fréquence relative des différents acides aminés qui composent les protéines. Or, cette fréquence est très variable. Ainsi, dans un alignement de séquence, la conservation d'un acide aminé « rare » (Tryptophane, Cystéine) n'a pas la même significativité statistique que la conservation d'un acide aminé « abondant » (Alanine, Valine). De plus, certains acides aminés présentent des structures et des fonctions chimiques voisines. Le remplacement d'un acide aminé par un autre acide aminé de fonction voisine peut être considéré comme une forme de conservation. Afin de tenir compte des caractéristiques propres à chaque acide aminé, des matrices de scores, appelée matrice de similarité, ont été développées.

2.4.1.1 Les matrices de scores

Pour quantifier la similitude entre séquences, un score est calculé pour chaque paire d'acides aminés alignés. Ce score peut être un score de distance (à minimiser) ou de similarité (à maximiser). Ces matrices de scores servent donc à réaliser une pondération de remplacement d'un acide aminé par un autre selon divers critères. Il s'agit donc d'attribuer des scores différents aux ~200 « mutations » symétriques possibles (20x19/2) entre acides aminés. Plusieurs matrices ont été développées, basées sur les caractéristiques physico-chimiques des acides aminés, mais surtout à partir d'analyse statistiques des substitutions au sein d'un ensemble de familles de séquences alignées. Ces dernières, telles que les matrices PAM ou BLOSUM, sont couramment utilisées.

PAM. M. Dayhoff et ses collègues ont établi une méthode permettant d'estimer la probabilité qu'un acide aminé *i* soit remplacé par un acide aminé *j* à une position donnée au cours de l'évolution, sans que la fonction de la protéine ne soit altérée (Dayhoff et *al.*, 1972, 1978 ; Kosiol et Goldmann, 2005). Cette estimation repose sur l'analyse d'alignements de séquences proches au sein desquelles il est peu probable qu'une mutation de A vers B résulte de mutation successives A→X→Y→B. Elle a été obtenue à partir de l'analyse de 71 alignements globaux de famille de protéines (~1300 séquences) très semblables (à 85% d'identité en séquences). Les matrices de probabilités de mutation issues de ces analyses dépendent donc des différences de séquences, fréquences et mutations observées au sein de ces 1300 séquences pour 100 sites estimés sur un temps d'évolution particulier (1 mutation sur 100 sites). On parle alors d'une 1PAM (1 Percent Accepted Mutations) matrice. Si la matrice est multipliée par elle-même un certain nombre de fois, une matrice XPAM est obtenue, et donne des probabilités de substitution pour des distances d'évolution plus grande. Pour être plus facilement utilisable dans les programmes de comparaison de séquences, chaque matrice XPAM est transformée en une matrice de similitudes PAM-X appelée matrice de mutation de Dayhoff. Cette transformation revient à diviser la probabilité de substitution observée pour une paire, par les fréquences des deux acides aminés la composant. On obtient une chance (« *odd* »), c'est-à-dire un rapport de probabilité (la probabilité d'obtenir une substitution observée dans des alignements de familles, divisée par la probabilité d'obtenir cette substitution par hasard (due seulement à la composition des séquences). Afin d'additionner les scores (au lieu de multiplier les probabilités) on prend le logarithme de cette « chance » (« *log-odd* »). C'est ce nombre (ramené à un entier) qui constitue le score d'alignement d'une paire d'acides aminés. Des études de simulation ont montré que la PAM-250 semble optimale pour distinguer des protéines apparentées de celles possédant des similarités dues au hasard (Schwartz et Dayhoff, 1979). C'est pourquoi, la matrice PAM-250 est devenue la matrice de mutation standard de Dayhoff. Cette matrice est basée sur un échantillon assez large et représente assez bien les probabilités

de substitution d'un acide aminé en un autre suivant que cette mutation engendre ou pas des changements dans la structure ou la fonctionnalité des protéines. Néanmoins, elle présente un certain nombre d'inconvénients. Principalement, elle considère que les points de mutation sont équiprobables au sein d'une même protéine (George et *al.*, 1990). Or, on sait que ceci n'est pas vrai et qu'une protéine peut présenter plusieurs niveaux de variabilité le long de la séquence. De plus, l'ensemble des protéines utilisé en 1978 n'est pas entièrement représentatif des différentes classes de protéines connues. Ainsi l'échantillon de 1978 était composé essentiellement de petites molécules solubles très différentes des protéines membranaires ou virales que l'on peut étudier aujourd'hui. Ce constat a conduit à une réactualisation de la matrice (Jones et *al.*, 1992) en considérant 16 130 séquences issues de la version 15 de Swiss-Prot, ce qui correspond à 2 621 familles de protéines. Cette étude a permis de prendre davantage en compte les substitutions qui étaient mal représentées en 1978. Une autre critique –mais qu'on peut adresser à l'ensemble des matrices de scores– est que l'on ne prend pas en compte l'environnement (voisinage) de la paire considérée.

BLOSUM. On doit cette matrice à Henikoff et Henikoff (Henikoff et Henikoff, 1992). Une approche différente a été réalisée ici pour mettre en évidence le caractère de substitution des acides aminés. Alors que les matrices PAM dérivent d'alignements globaux (cf. la recherche d'alignements optimaux page 85) de protéines très semblables, ici le degré de substitution des acides aminés a été mesuré en observant des blocs d'acides aminés issus de protéines plus éloignées. Chaque bloc est obtenu par l'alignement multiple sans insertion-délétion de courtes régions très conservées (cf. la base BLOCK page 80). L'hypothèse sous-jacente est que ces blocs correspondent à des éléments bien conservés au sein des structures tridimensionnelles des protéines correspondantes. Au sein de chaque alignement, les séquences sont ensuite regroupées en familles partageant plus d'un certain pourcentage seuil d'identité. Les fréquences de substitution pour toutes les paires d'acide aminé sont alors calculées et permettent d'obtenir une matrice logarithmique de probabilité dénommée BLOSUM (BLOcks Substitution Matrix). Différentes matrices ont été calculées en faisant varier le pourcentage d'identité seuil. A titre d'exemple, la matrice BLOSUM62 (cf. Figure 2-9) a été obtenue en utilisant un seuil de 62% d'identité : si deux séquences d'un même alignement initial partagent plus de 62% d'identité, elles ne sont alors représentées qu'une seule fois dans l'alignement servant à construire la matrice BLOSUM62.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0	-4
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1	-4
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
N	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1	-4
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3	-2	-4
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1	-4
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1	-4
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1	-4
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1	-4
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1	-4
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1	-4
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1	-1	-4
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3	-1	-4
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1	-2	-4
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0	0	-4
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1	0	-4
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3	-2	-4
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2	-1	-4
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2	-1	-4
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1	-1	-4
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1	-4
*****	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	1

Figure 2-9 Matrice de substitution BLOSUM62. Les scores positifs indiquent les acides aminés considérés comme similaire. Les scores de la diagonale correspondent aux valeurs attribuées en cas d'identité. On compte 24 acides aminés au lieu de 20 : X et * correspondent à des acides minés indéterminés, B correspond à D ou N, et Z peut remplacer E ou Q.

Il est à noter que par la suite, d'autres matrices ont vu le jour, dérivées d'alignements structuraux (Risler et *al.*, 1988 ; Overington et *al.*, 1990 ; Bowie et *al.*, 1991).

2.4.1.2 La prise en compte des insertions dans les alignements par paires

La comparaison locale de deux séquences, qu'elles soient nucléiques ou protéiques, repose sur une hypothèse d'évolution par mutations ponctuelles, à savoir, l'un des trois cas de figure qui suit : (i) le remplacement d'un acide aminé ou d'un nucléotide par un autre, (ii) une délétion (iii) ou une insertion. Afin d'évaluer la qualité d'un alignement, un score élémentaire issu de la matrice de substitution est calculé à chaque position alignée et une pénalité est également calculée en cas d'insertion ou de délétion (« indel »). La valeur attribuée aux pénalités d'indel est généralement calculée avec une loi affine associant une pénalité d'ouverture de l'indel à une pénalité d'extension augmentant linéairement avec la longueur de l'indel. L'avantage de ce calcul est sa simplicité mais la modélisation de la pénalité à attribuer aux insertions fait encore l'objet d'optimisation, en particulier dans le cas de l'alignement de séquences très divergentes.

2.4.1.3 Les algorithmes d'alignement

Il existe un certain nombre de façon d'aligner deux séquences entre elles en tenant compte des insertions et/ou délétions. Ainsi, pour deux séquences de longueur M et N, le nombre d'alignements possibles est de $n=2^M2^N$. Par exemple, rien que pour deux séquences de 20 résidus, il y aurait $2^{(40)}$

façons différente d'aligner, soit à peut près 137 milliards de combinaisons possibles. Il existe une solution pour trouver l'alignement optimal : la programmation dynamique.

Programmation dynamique. Les méthodes de programmation dynamique sont des approches algorithmiques développées par le mathématicien Richard Bellman visant à réduire la complexité de certains problèmes combinatoires par recherche d'optimum de sous problèmes. Appliquées au problème des alignements de séquences, les méthodes programmation dynamique construisent un alignement optimal de sous-séquences de plus en plus longues en utilisant les scores obtenus pour les sous-séquences. Les méthodes de programmation dynamique les plus couramment utilisées pour les alignements de séquences sont les algorithmes de Needleman et Wunsch (1970) et Smith et Waterman (1981).

Needleman et Wunsch (NWS). L'algorithme de Needleman et Wunsch a été développé pour réaliser l'alignement global de deux séquences protéiques (Needleman et Wunsch, 1970). Cet algorithme s'organise en deux étapes. Une matrice d'alignement dont les deux dimensions correspondent aux deux séquences à aligner, est remplie à chaque position par le score maximal d'un alignement qui se terminerait à cet élément (règle de récurrence simple présenté à la Figure 2-10) selon la formule suivante :

$$S(i,j) = \max \begin{cases} S(i-1,j-1) + S_{SubMatrix(a_i,a_j)} \\ S(i,j-k) + k \times W_{insert} \\ S(i-l,j) + l \times W_{insert} \end{cases}$$

Où $S(i,j)$ est le score somme de la case d'indice i et j, $S_{SubMatrix(ai,aj)}$ le score de la matrice de similarité (score PAM ou BLOSUM par exemple) entre l'acide aminé a_i et a_j, et W_{insert} la pénalitée d'insertion d'un gap sur la séquence. À chaque case de la matrice, le chemin d'où l'on provient (substitution, délétion ou insertion) est noté.

Enfin, la matrice d'alignements, parcourue en sens inverse, à partir du bout des deux séquences en remontant le chemin noté, permet d'identifier l'alignement global optimal entre les deux séquences (flèches vertes dans la Figure 2-11). De nombreux programmes utilisent cet algorithme d'alignement. Le programme ALIGN (Dayhoff et al., 1979) en est une application directe avec l'utilisation de pénalités à deux paramètres (dépendant et indépendant de la longueur).

	0	H	E	A	G	A	W	G	H	E	E
0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	-8	-2									
A	-16	-10									
W	-24	-18									
H	-32										
E	-40										
A	-48										
E	-56										

Figure 2-10 Construction de la matrice des scores d'alignement entre les deux séquences « HEAGAWGHEE » et « PAWHEAE » calculée à partir d'une matrice de similarité basée sur les scores de substitution BLOSUM62 et un coût d'insertions (Winsert) constant de -8. La séquence horizontale est indicée en *i* et la séquence verticale en *j*. Pour chaque cellule, le score maximal obtenu à partir des trois flèches de couleurs est attribué à cette cellule. Dans l'exemple, la flèche rouge correspond à une absence d'insertion et le score associé se calcule par $S(i,j) = S(i-1,j-1) + S_{BLOSUM62(W,H)} = -16 - 2 = -18$, la flèche verte à une insertion dans la séquence 1 dont le score associé est $S(i,j) = S(i,j-k) + k*Winsert = -10 - 8 = -18$ avec (k=1), et la flèche bleue, à une insertion dans la séquence 2 ayant un score associé de $S(i,j) = S(i-l,j) + l*Winsert = -24 - 8 = -32$ (l=1). Dans la Figure 2-11 ci-dessous, seuls les parcours passant par les flèches bleues et vertes sont conservés car ils sont associés aux trajectoires ayant fourni les scores maxima.

	0	H	E	A	G	A	W	G	H	E	E
0	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	-8	-2	-9	-17	-25	-33	-41	-49	-57	-65	-73
A	-16	-10	-3	-5	-13	-21	-29	-37	-45	-53	-61
W	-24	-18	-11	-6	-7	-15	-10	-18	-26	-34	-42
H	-32	-16	-18	-13	-8	-9	-17	-12	-10	-18	-26
E	-40	-24	-11	-19	-15	-9	-12	-19	-12	-5	-13
A	-48	-32	-19	-7	-15	-11	-12	-12	-20	-13	-6
E	-56	-40	-27	-15	-9	-16	-14	-14	-12	-15	-8

HEAGAWGHE-E
-PA--W-HEAE

Figure 2-11 Lecture de la matrice de scores d'alignement afin d'identifier l'alignement optimal par l'algorithme NWS. Les flèches indiquent à partir de quelle cellule le score d'alignement a été obtenu à l'étape de construction de la matrice (Figure 2-10). Les flèches vertes indiquent le parcours produisant le score optimal parmi les flèches rouges illustrant les sous-parcours optimaux. Les flèches bleues indiquent les parcours produisant les mêmes scores que le parcours optimal. L'alignement optimal (parcours vert) est présenté en également.

Smith-Waterman. L'algorithme de Smith et Waterman (Smith et Waterman, 1981) permet d'identifier le(s) meilleur(s) alignement(s) local(aux) deux séquences. La procédure est directement inspirée de celle de Needleman et Wunsch. La principale différence vient du fait que n'importe quelle case de la matrice d'alignement peut être considérée comme un point de départ pour une trajectoire maximisant un score d'alignement. On arrive à ce comportement en réinitialisant le score à 0 dès qu'il devient inférieur à 0. Ainsi, la case en question peut être considérée comme un nouveau point de départ d'un petit alignement de type NWS (qui s'arrêtera dès que le score diminue). Une fois la matrice d'alignement remplie, il suffit de noter la case de score le plus élevé et de remonter jusqu'au prochain zéro pour obtenir le meilleur alignement local (on peu répéter l'opération pour obtenir les autres meilleurs alignements locaux)

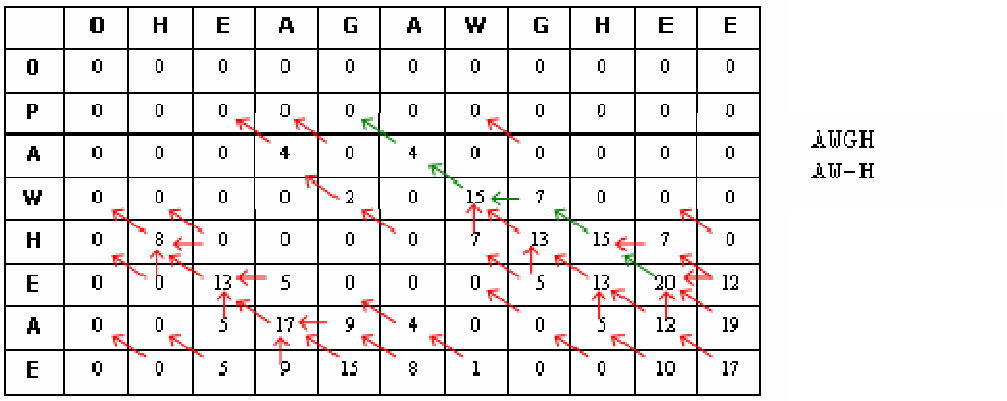


Figure 2-12 Lecture de la matrice de scores d'alignement afin d'effectuer l'alignement optimal par l'algorithme de Smith et Waterman. Les flèches indiquent à partir de quelle cellule le score d'alignement maximal a été obtenu à l'étape de construction de la matrice. Les flèches vertes indiquent le parcours produisant le score optimal parmi les flèches rouges illustrant les sous-parcours optimaux. L'alignement optimal correspondant au parcours vert est présenté.

Développement des méthodes heuristiques. Les méthodes de programmation dynamique permettent l'obtention d'un alignement optimal entre deux séquences, mais elles présentent en contrepartie un coût en temps de calcul et en mémoire important. Ces méthodes sont donc adaptées pour aligner un nombre limité de séquences, mais sont moins utilisées pour comparer une séquence avec un grand ensemble de séquences (bases de données)[3]. Le but de ces méthodes heuristiques est de calculer à moindre coût des alignements pas nécessairement optimaux mais de qualité suffisante pour établir des relations de similarités entre les séquences. Ces méthodes heuristiques ont été déjà évoquées précédemment : il s'agit des algorithmes FASTA et BLAST.

[3] Du moins, jusqu'à présent, car vu l'augmentation des puissances de calcul, les logiciels tels que BLITZ (Sturrock et Collins, 1994) ou SSEARCH (Pearson, 1991) permettent de le faire à présent.

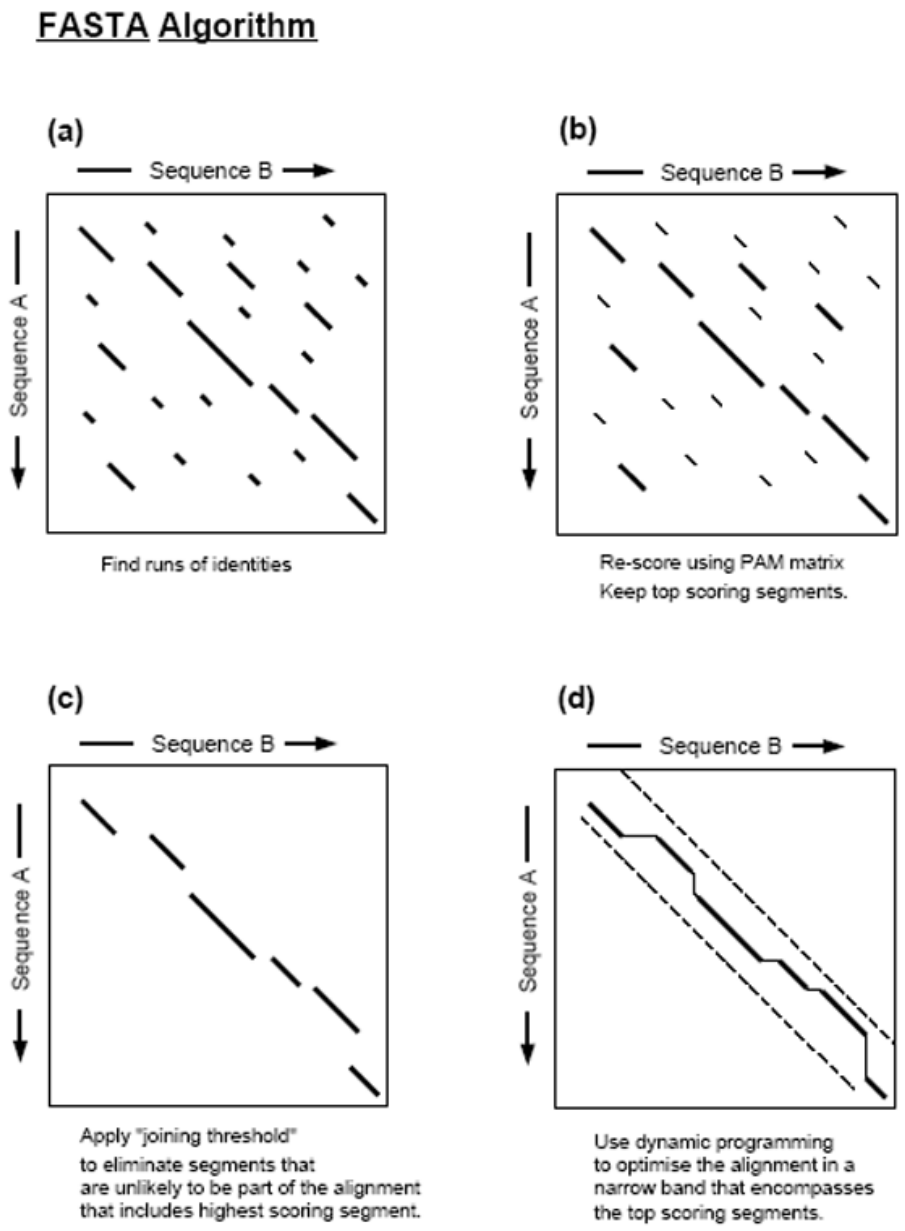


Figure 2-13 Algorithme FASTA. (a) Comparaison de la position et de la nature de segments de longueur L entre deux séquences afin de repérer les régions les plus denses en identités partagées. (b) Évaluation des 10 régions présentant les plus hauts scores de similarité calculés à partir des matrices de substitution. Cette étape correspond à une recherche de similitude sans insertions. Le score *init1* est attribué à la région ayant le plus fort score parmi les 10 analysées (*initn* est le score incluant les autres régions). (c) Jonction des régions précédentes, s'il en existe au moins deux et si chacune d'elles possède un score supérieur à un seuil donné. Ce seuil correspond à un score moyen attendu pour des régions non apparentées. Les régions initiales sont réunies à chaque fois que leur score diminué d'une pénalité de jonction est supérieur ou égal au score *init1*. Cette étape permet d'éliminer les segments peu probables parmi ceux définis à l'étape précédente. (d) Alignement optimal (par programmation dynamique) des deux séquences en considérant uniquement les régions définies à l'étape précédente avec calcul d'un score (opt). Cette étape n'est accomplie que pour les « meilleurs » séquence de la banque (Source Barton, 1996)

FASTA est un logiciel d'alignement de séquences aussi bien protéiques que nucléiques. Il a été pour la première fois décrit par D.J Lipman et W.R Pearson (Pearson et Lipman, 1988) sous le nom de FASTP car il n'était élaboré que pour la recherche de similarité dans les séquences protéiques. FASTA est apparu ensuite comme une amélioration de FASTP, incluant la recherche de similarité entre séquence nucléique contre séquence nucléique, et séquence de protéine traduite contre séquence nucléique. Cette nouvelle version de FASTA incorpore une évaluation de la signifiance statistique des alignements obtenus. Le logiciel FASTA présenté précédemment procède en quatre étapes telles présentées sur la Figure 2-13. Le programme calcule ensuite un z-score qui correspond à un score diminué de la moyenne des scores des séquences de la banque et normalisé par l'écart type des scores.

BLAST (pour Basic Local Alignement Searc Tool) développé par S.F Altschul (Altschul et al., 1990), est lui aussi un outil pour comparer les séquences biologiques primaires. A ce jour, il est certainement l'outil le plus utilisé en bioinformatique, probablement parce qu'il repose sur une grande efficacité algorithmique associée à une fiabilité statistique suffisante. Ces deux propriétés font de lui l'un des programmes de recherche les plus rapide à ce jour, permettant ainsi la recherche sur des grandes banques de données. Il permet ainsi de répondre à ce genre de questions qu'un chercheur a l'habitude de se poser : « D'où provient cet ADN que je viens juste de séquencer ? » ou encore « Quels autres gènes codent pour des protéines qui présentent des structures ou motifs tels que celui dont je viens de déterminer ? » etc. … BLAST s'est vu décliné en plus de 7 versions différentes, comme BLASTN qui compare deux séquences nucléiques, BLASTP, deux séquences protéiques, BLASTX une protéine traduite à partir d'une séquence nucléique selon les 6 cadres de lecture… L'une de ces versions, appelé PSI-BLAST présente des caractéristiques fort intéressantes car l'algorithme est itératif et permet un affinement de la recherche basée sur les similarités déjà trouvées. Tout comme FASTA, BLAST est une heuristique d'alignement local. Son fonctionnement peut être décrit en trois étapes. D'abord, il cherche des coïncidences de mots de longueur W (usuellement égale à W=3 pour les séquences protéiques) qui se ressemblent entre la séquence requête et les séquences de la banque. Pour ce faire, BLAST calcule un dictionnaire des mots équivalents à ceux de la séquence requête et ne conserve que ceux dont le score d'alignement obtenu à l'aide d'une matrice de substitution, dépasse un certain seuil (seuil "T"). Une fois ces mots rangés, BLAST parcours ensuite les séquences de la banque, et s'arrête à chaque mot d'une séquence de la banque ayant un correspondant dans le dictionnaire de mots similaires tiré de la séquence. BLAST essaie alors d'étendre l'alignement local (sans insérer de gap) de part et d'autre des ces "amorces" – constituées des 2 mots coïncidant – pour obtenir un alignement local optimal (ou HSP pour "High-scoring Segment Pairs") dont le score est au moins à S ou une *E-value* inférieur à un seuil spécifié.

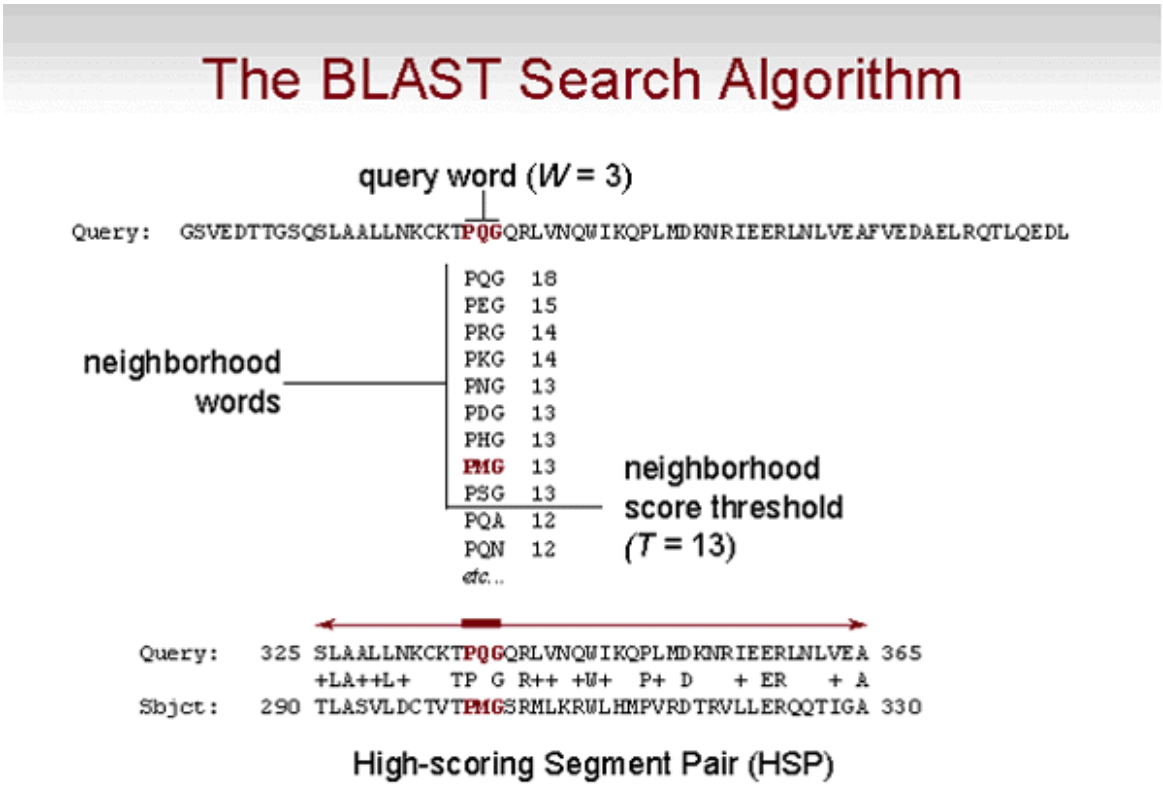


Figure 2-14 Algorithme du logiciel BLAST (Altschul et *al.*, 1990) (source de la figure : http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/BLAST_algorithm.html)

Pour évaluer la validité statistique d'un alignement, le programme BLAST s'appuie sur une expression analytique qui permet de calculer la probabilité d'obtenir l'alignement par hasard à partir du score de l'alignement en question. La distribution des scores d'alignement obtenus par la comparaison d'une séquence avec l'ensemble des séquences d'une base de données est supposée suivre une loi de distribution dite des valeurs extrêmes. Fonction du score, la *E-value* (ou nombre d'alignement attendu par hasard ayant un tel score) peut s'écrire de la façon suivante :

$$E\text{-}value = K \times L_1 \times L_2 \times e^{-\lambda \times Score}$$

Dans cette expression L_1 et L_2 sont des longueurs des deux séquences à aligner, K et λ des paramètres de normalisation ajustés en fonction de la matrice de substitution utilisée, de la taille de la base de données, de la taille des séquences. Plus la *E-value* est faible, plus l'alignement est supposé significatif.

2.4.2 Les méthodes d'alignements multiples

La recherche par paires entre une séquence d'intérêt et les séquences des bases de données a permis d'identifier un ensemble de séquences (supposées homologues). L'étape qui suit a pour objectif d'agencer en colonne les acides aminés qui possèdent la même histoire évolutive : c'est l'étape d'alignement multiple. L'obtention d'alignements multiples constitue une étape essentielle de l'analyse bioinformatique : elle permet de mettre en évidence les positions importantes pour la structure et/ou la fonction. Dans le contexte de la prédiction de structure, l'optimisation des alignements multiples revêt donc une importance cruciale.

2.4.2.1 Méthodes optimales

L'algorithme de programmation dynamique (NWS) décrit précédemment pour deux séquences est généralisable à l'alignement de *N* séquences (Kruskal et Sankoff, 1983). Le souci qu'on rencontre lors d'utilisation de cet algorithme est sa complexité en temps et en mémoire (de l'ordre de $O(N^k)$). Quelques astuces ont été mises au point pour accélérer le temps de calcul, comme dans la méthode **MSA** où la matrice n-dimensionnelle est « tronquée de ses coins », mais les calculs demeurent de infaisables sur plus d'une dizaine de séquences.

2.4.2.2 Méthodes heuristiques

De la même manière que pour les alignements deux à deux, des méthodes heuristiques ont été développées pour pallier à la complexité de l'algorithme. Une façon de résoudre le problème de l'alignement de *N* séquences consiste à utiliser une procédure progressive d'alignement par paires qui peut être esquissée comme suit : (i) Alignement par programmation dynamique des deux première séquences, (ii) Alignement de la 3^e séquence avec l'alignement précédent, (iii) Alignement de la séquence *N* avec l'alignement obtenu sur les *N-1* premières séquences. Ce type de méthode présente toutefois une part de difficulté : il faudra toujours définir l'ordre dans lequel les séquences doivent être prises en compte car l'alignement final en dépend. Plusieurs solutions ont été envisagées pour contourner ce dernier problème. Il est possible dans un premier temps de limiter la difficulté à l'alignement de trois courtes séquences (Murata et *al.*, 1985) ou de séquences relativement proches (Bains, 1986). On peut également chercher à définir d'abord « l'ordre de passage » des séquences à traiter au moyen d'arbres de distances (Higgins et *al.*, 1992 ; Higgins et Sharp, 1988 ; Sankoff et *al.*, 1976) ou en utilisant les scores obtenus par alignement par paire des différentes séquences pour ensuite les aligner en sous groupes, puis en groupes (Corpet, 1988 ; Taylor,1987). Un nombre important de logiciels et de programmes exploitent ces différentes approches : il serait long de tous les

présenter. Ici, on se propose de décrire ClustalW, l'un des plus simples, rapides largement utilisé (et aussi cours de cette thèse).

CLUSTALW. De tous les logiciels, ClustalW (Thompson et *al.*, 1994) est sans conteste l'un des plus populaire dans la production d'alignements multiples à partir de séquences récupérés à partir d'un BLAST par exemple. Il procède en plusieurs étapes. La première étape calcule un arbre « dendogramme » qui sert de guide pour l'alignement multiple. Pour cela, un alignement de toutes les séquences deux à deux est effectué par un algorithme relativement frustre mais extrêmement rapide (*n.(n-1)* comparaisons) qui donne un score de distance pour chaque couple. Un arbre est alors construit par Neighbor-Joining à partir de la matrice des distances. Les deux séquences les plus proches sur l'arbre sont alignées. Un « profil » est alors construit à partir de cet alignement. Chaque position de ce profil représente la « moyenne » des deux séquences de la paire. La séquence suivante (ou profil de séquences déjà alignées) la plus proche (par rapport à la topologie de l'arbre) est alors alignée sur le profil de la paire. On peut noter que les gaps aux positions terminales des séquences ne coûtent rien. Les gaps seront introduits soit dans la nouvelle séquence (ou le nouveau profil), soit dans le profil. Ce point représente une des principales limites du programme car il lui est impossible d'effectuer un gap ou de le recalculer sur seulement une portion des séquences déjà alignées.

2.4.2.3 Alignements multiples et profils.

PSI-BLAST. Les procédés heuristiques précédemment décrits génèrent généralement un grand nombre d'insertions qui rend les alignements multiples difficiles à exploiter. Ces procédés constituent néanmoins une première approximation efficace. PSI-BLAST a été créé pour rechercher des homologies éloignées, à faible identité de séquence. Sa première itération est un simple BLASTP qui va donner les voisins proches de la séquence protéique recherchée. À partir des résultats obtenus, la distribution des acides aminés et des insertions dans chaque colonne de l'alignement multiple permet d'extraire une fréquence d'occurrence pour chaque position qui peut être traduite en termes de scores de probabilité sous forme d'une matrice appelée PSSM (Position Specific Score Matrix ou « profil »). Un exemple de PSSM généré par le logiciel PSI-BLAST est illustré sur la Figure 2-15. C'est ce profil qui est alors utilisé en remplacement de séquence dans une seconde itération pour rechercher de nouvelles séquences.

POS	PROBE	CONSENSUS		PROFILE																			
				A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	+/-
1	E G V L		V	3	-2	3	4	0	4	-1	3	-1	4	4	1	1	1	-2	1	2	6	-6	-2	9
2	L L S P		L	2	-2	-2	-1	3	0	-1	3	-1	6	5	-1	3	0	-1	3	1	4	1	-1	9
3	V V V V		V	2	2	-2	-2	2	2	-3	11	-2	8	6	-2	1	-2	-2	0	2	15	-9	-1	9
4	K E A T		A	6	-2	5	6	-5	4	1	0	5	-2	0	3	3	3	1	3	6	0	-6	-4	9
5	A P L P		P	6	-1	0	1	-2	2	0	1	0	2	2	0	8	2	0	2	2	3	-5	-4	9
6	G G G G		G	7	1	7	5	-6	15	-1	-3	0	-4	-3	4	3	2	-3	6	4	2	-11	-7	9
7	S S Q E		D	4	-1	7	7	-6	7	2	-2	2	-3	-2	4	3	6	1	6	2	-1	-6	-5	9
8	S S T P		S	4	4	2	2	-4	4	-1	0	2	-3	-2	2	7	0	1	10	6	0	-2	-4	9
9	V L V A		V	5	0	-1	-1	3	1	-2	7	-2	7	6	-1	1	-1	-3	0	2	10	-5	-1	9
10	K R R S		R	0	-1	1	1	-5	0	2	-2	8	-3	1	3	3	3	10	5	1	-2	7	-5	9
11	M L I I		I	0	-2	-3	-2	7	-3	-3	11	-1	11	10	-2	-2	-1	-2	-2	1	9	-3	1	9
12	S S T S		S	4	6	2	2	-3	5	-1	0	2	-3	-2	3	4	-1	1	12	6	0	0	-4	9
13	C C C C		C	3	15	-5	-5	-1	2	-1	3	-5	-8	-6	-3	1	-6	-3	7	3	3	-13	10	9
14	K S Q R		K	1	-2	3	3	-6	1	3	-2	7	-3	0	3	3	5	7	4	1	-2	2	-5	9
15	A A G S		A	10	3	4	3	-5	8	-1	-1	1	-2	-1	3	4	1	-2	7	4	2	-6	-4	9
16	T S D S		S	4	3	5	4	-5	6	0	0	2	-3	-2	4	3	1	1	9	6	0	-3	-4	9
17	G G S Q		G	5	1	6	5	-6	9	1	-2	1	-3	-2	4	3	4	0	6	3	0	-6	-6	9
18	Y F L S		F	-1	2	-4	-3	9	-3	0	4	-3	6	3	-1	-3	-3	-3	1	-1	2	7	7	9
19	T T R L		T	1	-2	0	1	0	0	0	2	2	2	3	1	1	1	3	1	7	2	1	-2	9
20	F F . L		F	-2	-3	-6	-4	10	-4	-1	6	-4	9	6	-3	-4	-4	-3	-2	-1	3	7	8	4
21	S S . D		S	3	2	5	4	-4	5	0	-1	2	-3	-2	4	3	1	1	8	2	-1	-2	-3	4
22	S . . S		S	2	3	1	1	-2	3	-1	0	1	-2	-1	2	2	0	1	8	2	0	1	-2	4
23	. . . G		G	2	0	2	1	-2	4	0	0	0	-1	-1	1	1	1	-1	2	1	1	-3	-2	4
24	. . . D		D	1	-1	4	3	-2	2	1	0	1	-1	-1	2	1	2	0	1	1	0	-3	-1	4
25	. . . G		G	2	0	2	1	-2	4	0	0	0	-1	-1	1	1	1	-1	2	1	1	-3	-2	4
26	. A G N		A	6	0	4	3	-4	6	1	-1	1	-2	-1	5	2	2	-1	3	3	1	-5	-3	4
27	Y N Y T		Y	0	5	0	-1	5	-1	2	1	-1	0	-1	4	-3	-2	-2	0	3	0	3	6	4
28	E D D Y		D	2	-2	9	8	-3	3	4	-1	1	-3	-2	5	-1	4	-1	1	1	-1	-6	0	9
29	L M A L		L	3	-5	-3	-1	6	-1	-2	6	-1	10	10	-2	0	0	-2	-1	0	6	-1	0	9
30	Y N A W		N	4	1	3	2	0	2	3	-1	1	-1	-1	8	0	1	-1	2	1	-1	-1	2	9
.	.													.										
.	.													.										
.	.													.										
48	S G N S		S	4	3	5	3	-4	7	0	-2	2	-4	-3	6	3	1	0	10	3	0	-2	-4	9
49	S S N Y		S	2	5	2	1	1	2	1	0	1	-2	-2	5	1	-1	0	8	1	-1	3	1	9

Figure 2-15 Représentation d'un profil paramétrant un alignement de 4 séquences, présenté en vertical à gauche. Pour Chaque position un score de substitution différent est calculé pour chaque acide aminé et pour les insertions (Gribskov et *al.*, 1987).

Plusieurs itérations permettent d'affiner progressivement le profil et d'augmenter considérablement la sensibilité de la méthode de détection. Ces itérations peuvent être effectuées jusqu'à la convergence, c'est-à-dire qu'aucune nouvelle séquence n'est détectée. L'avantage de cette méthode est qu'elle « gomme » les particularités de la séquence requête grâce aux séquences voisines (supposées être de sa famille). Le principal reproche qu'on peut donner à cette méthode vient du fait qu'un faux positif puisse apparaître au cours des itérations et qu'il soit pris en compte pour construire la PSSM : ce faux positif serait alors susceptible de biaiser l'ensemble du profil aux itérations ultérieurs. Quelques tentatives d'optimisations ont été proposées pour réduire le danger potentiel de l'intégration de ces faux-positifs (Schaffer et *al.*, 2001). Par ailleurs, l'approche itérative telle qu'elle vient d'être décrite présente un risque : lorsque le nombre de séquences est faible, les probabilités d'occurrence de certains acides aminés peuvent être estimées, faussant ainsi le profil associé à la famille de séquences homologues. Par exemple, l'observation d'une position exclusivement occupée par une isoleucine devrait laisser la possibilité que d'autres hydrophobes tels que la leucine ou la valine soient également probables. Pour améliorer les profils, il est possible d'enrichir les fréquences

observées dans l'alignement par la connaissance que l'on a, a priori, des relations entre acides aminés. Ces fréquences d'occurrence observées peuvent être corrigées par la méthode de « pseudo-count » comme c'est le cas dans PSI-BLAST. Dans cette approche, on ajoute une contribution variable des scores des matrices BLOSUM et PAM aux fréquences observées. L'importance de ces scores de « connaissance *a priori* » est pondérée en fonction de la richesse d'informations déjà contenues dans l'alignement multiple.

Les approches HMM. La notion de profil a été généralisée par le développement du formalisme des Chaînes de Markov Cachées (HMM) (Eddy, 1998). Ce formalisme associé aux HMM fournit un ensemble d'outils statistiques très performants pour manipuler et évaluer la vraisemblance d'un alignement. Les méthodes HMM sont très utilisées dans le traitement des séquences à grande échelle, dans la constitution et l'interrogation des bases de données de domaines telles que PFAM et SMART, mais également pour la détection et l'alignement des homologues lointains. Sans rentrer dans les détails, le formalisme HMM permet de générer un modèle statistique d'un alignement multiple dans lequel l'apparition des acides aminés dans l'alignement suit un processus stochastique de Markov (la probabilité de l'état n dépend uniquement de l'état *n-1*). Un alignement multiple peut être ainsi modélisé par une chaîne d'éléments qui possèdent 3 états (M pour une position alignée, I pour une insertion, D pour une délétion) avec des probabilités d'émission et de transition attribuées entre chacun des états (Figure 2-16). Les modèles HMM fournissent une flexibilité accrue par rapport aux profils en autorisant les états de délétions en plus des états d'insertions, et surtout en modélisant la relation de voisinage (ignorée dans le cas des PSSM). Les probabilités qui sous-tendent un alignement sont inconnues (variables « cachées ») et le premier objectif est de les estimer à partir des fréquences d'occurrence observées à chaque position. Comme pour les profils présentés précédemment, cette information est enrichie par la connaissance *a priori* des probabilités d'occurrence des acides aminés en fonction des contextes.

Le modèle HMM ainsi paramétré peut être utilisé pour reconnaître les séquences susceptibles d'être reliées aux séquences de l'alignement tel qu'il est décrit dans la Figure 2-16. L'algorithme de Viterbi, de façon similaire aux algorithmes de programmation dynamique, permet d'identifier la trajectoire la plus probable au sein du modèle HMM. Formellement, une séquence n'est pas alignée sur un HMM. Ce qu'on mesure, c'est la probabilité qu'un HMM donné puisse générer la séquence alignée de façon optimale. La base Pfam dont je me sers pour identifier des P450s, est créée à partir d'HHMER, un logiciel utilisant les HMM.

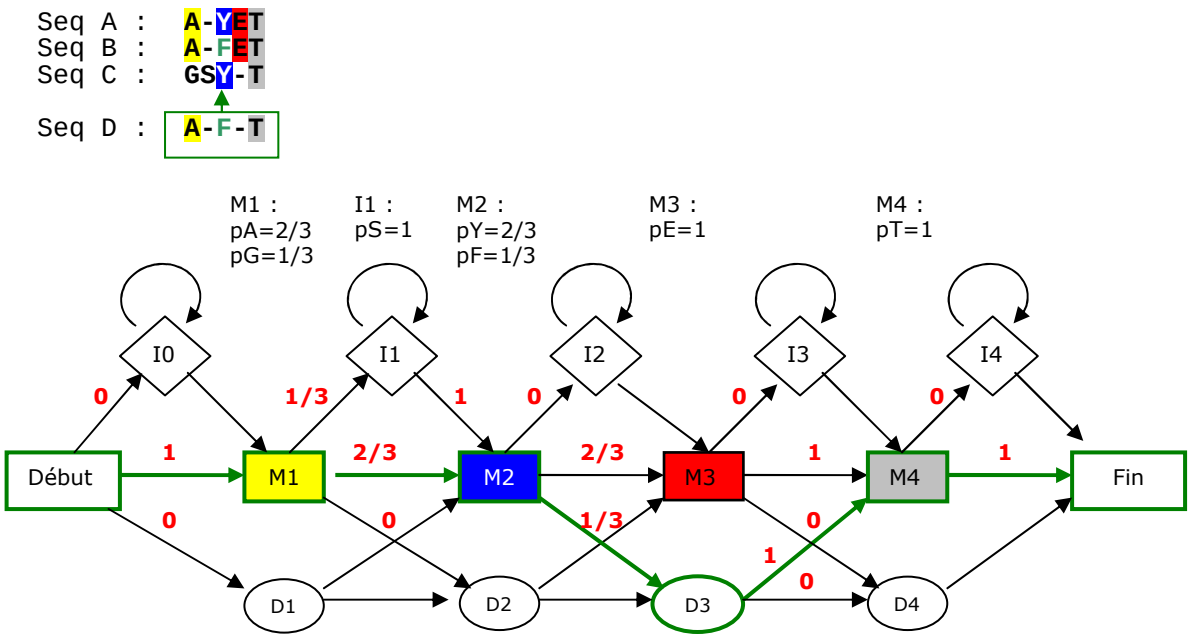


Figure 2-16 Architecture simplifié d'un modèle HMM (HMMER plan 7) et exemple de paramétrage. L'alignement des trois séquences A,B et C peut être représenté par une chaîne à 4 états. Les probabilités de transitions entre ces états déduites de l'alignement multiple sont indiquées en rouge au dessus de chaque flèche. Les probabilités d'émission de chaque acide aminé au sein de chaque état sont indiquées au dessus du modèle. L'indentification de la trajectoire pour le HMM génère la séquence D alignée de façon optimale par l'algorithme de Viterbi. La trajectoire permettant de maximiser des probabilités lors du parcours du HMM est indiquée en vert. L'alignement optimal pour la séquence D est indiqué en haut. (Source : thèse de V. Meyer)

2.4.3 Comparaison de deux structures

Après avoir abordé les comparaisons des séquences primaires issues des bases de données, il est intéressant de s'attarder à la comparaison de structure. Les techniques de comparaison de structures protéiques sont essentielles dans beaucoup de domaines de recherche, notamment dans la prédiction de structure d'une protéine et dans la compréhension de l'évolution des structures protéique. Initialement, les première méthodes de comparaison servaient à comparer une structure avec elle-même : il s'agissait tout simplement de comparaison de postions atomiques. Très vite, ces méthodes se sont montrées limitées : lors de comparaison de structures différentes, un problème de choix de correspondance entre les éléments à comparer se posait : comme deux protéines ne sont pas composées de la même séquence en acides aminés, il n'est pas possible de comparer tous les atomes, chaînes latérales comprises. Dans un premier temps, il a fallu ainsi se restreindre aux atomes du squelette peptidique. Toutefois, le rôle des chaînes latérales (dans le cas de reconnaissance ou fixation de ligand) ne pouvant être négligé, l'incorporation de ces atomes dans la comparaison devaient être également prise en compte. Par ailleurs, l'organisation des structures secondaires semblait être

conservée dans les protéines – même lointaines – d'une même famille. Ne comparer que celles-ci serait donc plus pertinent, au moins pour des protéines assez différentes. Au final, trois niveaux de représentation ont pu être recensés : une représentation « tout atome », une représentation restreinte aux atomes du squelette peptidique et une représentation en termes de structures secondaires. A ces trois représentations, une dernière peut être ajoutée : il s'agit de la représentation qui tient compte la forme générale de la molécule.

De façon similaire à la comparaison des séquences, l'hypothèse sous-jacente à la comparaison de structures est généralement l'homologie supposée de ces structures (ou d'une de leurs sous-structures). Ainsi, on cherche à mettre en correspondance la position des acides aminés tout en tenant compte de leur séquence et de sa divergence. Dans le cas où l'on veut mesurer la convergence éventuelle des structures ou des sites, la comparaison ne doit pas prendre en considération la séquence des résidus.

Quelles que soient les méthodes utilisées, le principe dans la comparaison de structure demeure inchangé : il s'agit de mettre en correspondance un élément d'une structure avec un seul élément de l'autre structure. L'objectif recherché est le calcul d'une mesure quantitative de similarité entre deux structures protéiques et/ou de générer un alignement structural, souvent converti ensuite en alignement de séquences. Quatre points sont importants lors d'une comparaison structurale. Ceux-ci ont été définis par Holm et Sander (Holm et Sander, 1996).

- **La représentation des structures** : les structures protéiques sont toujours simplifiées mais les caractéristiques conservées doivent être suffisantes pour la comparaison. Exemples : les $C\alpha$ sont décrits par leur coordonnées cartésiennes ou par leurs distances internes ; les Structures Secondaires sont décrites par des vecteurs ; *etc*…
- **La mesure de similarité ou de dissimilarité** : il faut pouvoir déterminer si un sous-alignement est meilleur qu'un autre pendant le processus d'alignement. Cette mesure est bien sûr totalement dépendante de la représentation ;
- **L'algorithme de comparaison** : un algorithme général bien connu peut être adapté au problème (recherche des cliques maximales dans un graphe, Monte Carlo…) ou un algorithme *ad hoc* peut être développé ;
- **Les post-traitements** : par exemple le calcul d'un score exprimant la significativité des résultats (il peut être empirique ou statistique, humain ou automatique).

Tout comme pour les séquences, les techniques et méthodes associées à ces comparaisons de structures sont extrêmement riches et diversifiées. Dans les sections qui suivent, ne seront présentées que succinctement les méthodes les plus usuelles et surtout celles qui ont servi pour cette thèse.

2.4.3.1 Description au niveau atomique

Les méthodes utilisant cette représentation permettent rarement de comparer des protéines entières. La majorité d'entre elles sont utilisées pour l'amarrage de molécules (*docking*). Elles se regroupent usuellement en deux catégories : les méthodes comparant des atomes ou des groupes d'atomes et les méthodes comparant des surfaces. La première catégorie est plus spécifique dans la comparaison de protéines entières tandis que la seconde est plus utilisée pour comparer des sites spécifiques. Dans ces catégories, il existe des méthodes de graphes, des méthodes de constructions de motifs structuraux, des méthodes par hachage géométrique mais également les méthodes fondées sur la forme, utilisées en morphométrie où l'on cherche à déceler des homothéties. Toutes ces méthodes permettent selon des principes et algorithmes différents de comparer deux structures entre elles, en prenant en compte tous les atomes.

Étant donné qu'aucune de ces méthodes n'a été utilisée dans mon travail de thèse, il n'est pas nécessaire de s'attarder d'avantage sur ces méthodes au niveau atomique.

2.4.3.2 Description au niveau du squelette peptidique

Les méthodes utilisées à ce niveau, déjà plus communes que celles décrites précédemment, présentent deux manières d'aborder cette description. Dans la première, le squelette peptidique est décrit par les coordonnées cartésiennes, dans la majeure partie du temps limitée à celles des $C\alpha$. Pour comparer deux structures à l'aide de cette description, il faut effectuer une transformation rigide d'une structure sur l'autre. Cette description est appelée « externe » par opposition aux descriptions internes où les deux structures peuvent être comparées directement. Les descriptions internes peuvent être des distances internes, des angles dièdres (ϕ,ψ) ou (α,τ) ou encore d'autres repères définis par des éléments des structures.

Mesures de similarité : les RMSD. Quelque soit les descriptions – externes ou internes – des mesures de similarités ont été mise en place afin de comparer deux structures. La plus usitée de toutes, est certainement le RMSD (*Root Mean Square Deviation*) sur les coordonnées. Il s'agit de la racine carrée de la moyenne des carrées des distances entre les atomes mis en correspondance dans les deux structures – décrits par leurs coordonnées cartésiennes –. Ce $RMSD_c$ est donc :

$$RMSD_c = \sqrt{\frac{\sum_{i=1}^{N} D(a_i, b'_i)^2}{N}}$$

où N est le nombre d'atomes mis en correspondance : dans la structure B, l'atome b'i est mis en correspondance avec l'élément a_i de la structure A, et $D(a_i,b'_i)$ est la distance entre les atomes a_i et b'_i après superposition optimale de tous les atomes mis en correspondance (ensembles *M(A)* et *M(B)*). Une superposition optimale de *M(A)* et *M(B)* est donc une transformation rigide *T* (une translation-rotation) telle que le $RMSD_c$ est minim*al*. Il existe de nombreuses méthodes pour trouver cette transformation optimale, comme celles faisant appel au formalisme fondé sur les quaternions (Kearsley, 1989) ou encore diagonalisation de matrices (Kabsch, 1976, 1978) itérations successives (Sippl et Stegbuchner, 1991) ou encore minimisation (McLachlan, 1979,1982). D'autres RMSD tels que le $RMSD_d$ et l'*URMS* et les *RMSD* pour les angles, existent mais ne seront pas présentées car elles n'ont pas été utilisées durant ce travail de thèse.

Coordonnées cartésiennes des $C\alpha$. Les premières méthodes de comparaison de structures protéiques utilisent la description des coordonnées cartésiennes, sont généralement restreintes à la comparaison des $C\alpha$. Dans ces méthodes, qui sont plutôt des méthodes de comparaison globale, l'objectif est de minimiser la valeur du $RMSD_c$ avec un maximum de $C\alpha$ en correspondance possible. Deux types de méthodes exploitant ces propriétés se distinguent : (i) les méthodes itératives de superposition - alignement et (ii) les méthodes basées sur des fragments structuraux similaires. Dans un cas, il s'agit de processus itératifs où les meilleures correspondances d'un ensemble de paires *P(A,B)* entre les deux structures A et B sont recherchées. Ces processus se font en deux étapes, la première par correspondance des $C\alpha$, cherche une superposition optimale des deux ensembles de points de *P(A,B)*, et la seconde, par programmation dynamique, détermine une nouvelle et meilleure correspondance en prenant en compte la superposition précédemment effectuée. Les programmes tels que SHEBA (Jung et Lee, 2000) TMalign (Zhang et Skolnick, 2005) ou encore MINAREA (Falicov et Cohen, 1996) illustrent cette catégorie de méthodes. Dans l'autre cas, il s'agit de travailler avec des fragments de taille donnée pour caractériser les paires de protéines similaires. Toutefois, ces méthodes-ci ne permettent pas l'alignement des deux structures entre elles. Ces méthodes basées sur les fragments structuraux similaires, sont décomposées en trois étape, comprenant dans la première étape une recherche de fragments similaires (AFP) dans les deux structures avec un $RMSD_c$ faible, puis dans une seconde étape une recherche des meilleures séries de AFP, et enfin dans la dernière étape qui est similaire aux méthodes de superposition – alignement, un affinement de l'alignement ou

de la correspondance au niveaux des résidus. On retiendra ici les programmes tels que WHAT IF (Vriend, 1990 ; Vriend et Sander, 1991) ou encore Flexprot (Shatsky et *al.*, 2002, 2004).

Coordonnées internes : les distances « internes ». L'utilisation des coordonnées internes dispense de l'étape de superposition. Les méthodes exploitant ces coordonnées entre atomes d'une même structure, ne prennent en compte que les $C\alpha$ (ou quelques autres atomes) pour comparer les structures au niveau peptidique. Il y a $N^2/2$ descripteurs pour chaque structure (N étant le nombre d'atomes) au lieu des $3 \times N$ paramètres de la description en coordonnées cartésiennes. Ces $N^2/2$ descripteurs sont souvent présentés sous la forme d'une matrice (symétrique) dite de distances internes. Les matrices de contact sont aussi parfois utilisées : elles sont remplies par des 1 si la paire d'atomes satisfait à certaines conditions (par exemple si la distance interne est en dessous d'un seuil) 0 sinon. Les méthodes utilisant les coordonnées internes sont très coûteuses en temps de calcul, aussi, à l'image de celles vues pour les séquences, diverses heuristiques ont été utilisées. Ces différentes méthodes développées se classent entre trois grandes catégories : méthodes utilisant la programmation dynamique, méthode d'assemblage d'AFP, méthodes utilisant les graphes. Dans le cas des méthodes utilisant la programmation dynamique, le programme SSAP (*Structure and Sequence Alignment Program*) – ou ses dérivés – (Taylor et Orengo, 1989a, 1989b) en est le seul représentant. Son dérivé SAP (*Structure Alignment Program*) (Taylor, 1999) est d'ailleurs utilisé pour établir la classification CATH (cf. section 2.3.3.3, page 76). Il procède par une méthode dite de double programmation dynamique avec un score basé sur les distances internes et se servant également d'autres descripteurs comme l'information de séquence, l'accessibilité au solvant etc.… Pour ce qui concerne les méthodes d'assemblage d'AFP, le principe repose sur la recherche de petits fragments similaires (AFP) puis de la meilleure série d'AFP par assemblage d'AFP entre eux. Les deux programmes les plus connus, DALI (cf. section 2.3.3.1 page 75) et CE (*Combinatorial Extension* Shindyalov et Bourne, 1998) utilisent ce principe. Enfin, dans le cas des méthodes utilisant les graphes, les distances internes sont considérées comme des « relations » entre atomes : chaque structure est représentée par un graphe ayant pour sommet les atomes et les distances pour arêtes pondérées. Par ailleurs, dans cette méthode, la contrainte de séquentialité a permis de réduire le problème de recherche des sous-graphes communs isomorphes. L'alignement est ensuite affiné en ajoutant les $C\alpha$ voisins des régions communes.

Coordonnées internes : les angles. Les angles utilisés pour la comparaison de structures protéiques sont les angles (ϕ,ψ) ou (α,τ) ou encore des dérivés de ceux-ci. A ce jour, il existe assez peu de méthodes de comparaison structurales qui utilisent cette description angulaires des structures. Ces méthodes reposant sur les angles ont pourtant été utilisées très tôt pour comparer les structures

entre elles. En dépit de leur rapidité, l'alignement fourni ne se révélait pas toujours très précis. Ce n'est que récemment que les angles ont été mieux exploités dans la comparaison de structure. **GOK** (Jean et al, 1997), le logiciel dont je me suis servi pour rechercher les similitudes communes entre les structures de P450s utilise les angles (α,τ) pour représenter la trajectoire du squelette protéique et comparer les structures des protéines entre elles. Il permet de déterminer les sous-trajectoires communes les plus longues, entre deux ou plusieurs structures. Son fonctionnement sera détaillé ultérieurement (cf. section 2.4.4.3, page 106).

2.4.3.3 Description en éléments de structures secondaires

Les structures secondaires sont des éléments structuraux réguliers, souvent considérés comme fragments structuraux les mieux conservés dans des structures similaires ou homologues. De plus, en raison de son niveau de représentation qui engendre moins d'éléments (~qu'une dizaine de SSE pour une protéine de 300 résidus), les comparaisons de structures sont plus rapides que les précédentes. Les méthodes de comparaison travaillant à ce niveau sont donc bien adaptées pour la recherche de similarités structurales dans les banques de structures. En contrepartie, les méthodes exploitant les SSE sont biens moins précises que celles vues précédemment, certaines régions de protéines pouvant être ignorées. Les structures secondaires prises en comptes par ces méthodes sont généralement uniquement des hélices α et des feuillets β. Il existe plusieurs méthodes d'attribution des éléments de structure secondaire. La plus courante est probablement DSSP (Kabsch et Sander, 1983).

Méthodes avec SSE représentés par des vecteurs. Les méthodes de comparaison utilisant les SSE sont très nombreuses. Pour la plupart, les SSE y sont représentées par des vecteurs colinéaires à leur axe principal qui n'est autre qu'une droite pour laquelle l'inertie des éléments (souvent uniquement des $C\alpha$) du SSE est minimale. Le vecteur dont les extrémités correspondent aux projections des atomes aux extrémités des SSE est orienté de l'extrémité N-terminal vers l'extrémité C-terminale. En conséquent, une structure est représentée par un ensemble de vecteur, réduisant le problème à la comparaison de deux ensembles de vecteurs (l'ordre des SSE pouvant être prise en compte ou non lors de la comparaison). Les algorithmes les plus utilisés pour la comparaison des ensembles de vecteurs proviennent de la théorie des graphes. Tous disposent d'une procédure assez similaire qui se déroulent en quatre étapes : (i) dans un premier temps, les SSE sont attribués et les paramètres des vecteurs calculés, (ii) puis un graphe est construit pour chaque structure ayant aux sommets les SSE et pour arêtes des informations sur l'organisation spatiale relative des vecteurs ; (iii) les graphes sont alors comparés ente eux avec une recherche de sous-graphes similaires communs et enfin en dernière étape (qui peut être facultative) un retour au niveau peptidique avec alignement

résidu par résidu. Parmi les programmes qui utilisent ces méthodes de graphes, on citera à titre d'exemple POSSUM (Protein Online Substructure Searching) (Mitchell et *al.*, 1990 ; Artymiuk et *al.*, 1990), PROFET (Grindley et *al.*,1993), GRATH(Harrison et *al.*, 2003), VAST (Gibrat et *al.*, 1996 ; Madej et *al.*, 1995) ou encore le très récent SSM (Krissinel et Henrick, 2004). Les méthodes de graphes ne sont pas exclusives dans l'utilisation des vecteurs de SSE. On y trouve également des méthodes par programmation dynamique utilisées dans le programme LOCK (Singh et Brutlag, 1997 ; Shapio et Brutlag, 2004) par exemple ou encore PrISM (Yang et Honig, 1999, 2000).

Méthodes avec SSE représentés par d'autres caractéristiques. Outre la représentation par vecteur, les SSE peuvent être dans certains cas représentés par d'autres caractéristiques. Dans la méthode **MATRAS** (Kawabata, 2003) la comparaison se fait par programmation dynamique à partir de scores de similarités basés sur un modèle d'évolution. Cette méthode a été largement utilisée au cours de la thèse pour comparer ses résultats avec le logiciel GOK : une description plus détaillée est donc nécessaire pour comprendre son fonctionnement. Dans cette méthode, trois scores de similarité structurale sont calculés de manière analogue au modèle de Dayhoff (matrice PAM) pour la substitution des acides aminés. Ces scores sont calculés entre paires de protéines homologues d'une banque non-redondante à 95% d'identité extraite de SCOP, regroupées et alignées grâce à leurs séquences. Une formule générale de log-odds est utilisée pour le calcul des trois scores :

$$S(i,j) = \log \frac{P(i \to j)}{P(j)}$$

Où i et j sont les états des caractéristiques structurales (telles que la distance inter résidus par exemple), $P(j)$ la probabilité que l'état j apparaisse par hasard, et $P(i \to j)$ la probabilité que l'état i passe à l'état j durant l'évolution, suivant une transition du modèle de Markov. Les trois scores de similarité structurale sont les suivants :

- un score S_{SSE}, de paires de SSE, où trois états sont les combinaisons d'hélices α et de feuillets β. Les SSE sont représentés comme des vecteurs. Quatre fonctions de similarité géométrique sont estimées pour chaque type de paire (θ_1, θ_2, d et ϕ, voir la Figure 2-17) ainsi que deux fonctions de similarité de longueur (L_1 et L_2). Le score S_{SSE} est la somme de ces six fonctions estimées sur l'échantillon des protéines homologues ;
- un score environnemental S_{env} où les états sont les deux structure secondaire (hélices α et feuillets β) ainsi que 3 types de boucles définis selon leurs angles (ϕ,ψ). Chacune des catégories peut être enfouie ou exposée, ce qui donne 10 états ;

- une score de distances S_{dis} où les états sont les distances internes entre $C\beta$ discrétisées avec un pas de 1Å. Les bornes sont [0,50Å], il y a donc 50 états. Le score S_{dis} entre une distance Di_x – entre le i^e^ et le x^e^ résidu d'une protéine – et la distance Dj_y – entre le j^e^ et le y^e^ résidu d'une autre protéine – est évalué par les transitions observées entre ces distances pour tous les intervalles k en résidus (k=|i–x|=|j-y|, l'intervalle est identique sur les deux structures).

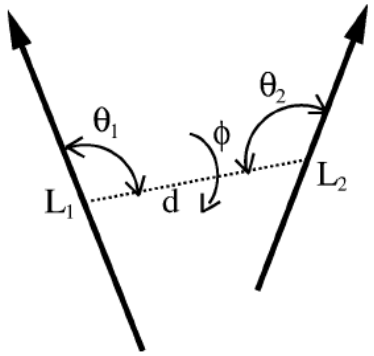


Figure 2-17 Les quatre paramètres géométriques associées aux pairs de vecteurs de SSE de MATRAS : la distance minimale entre les deux SSE d, les deux angles (plans) θ1 et θ2 et l'angle dièdre ϕ entre les plans formés par chacun des deux vecteurs et la droite minimale en pointillée. Les deux longueurs sont L_1, L_2 exprimées en résidus. Figure extraite de (Kawabata, 2000).

L'alignement se déroule alors en trois étapes :

1. premier alignement grossier des SSE en utilisant les scores S_{SSE} ; les meilleures suites de SSE sont trouvées par *branch and bound* ;
2. un second alignement des SSE est obtenu par programmation dynamique (alignement local avec pénalités d'ouverture et d'extension de gap) en utilisant le score environnemental S_{env} et en affectant un bonus pour l'alignement des SSE déjà en correspondance avec l'étape précédente ;
3. alignement des résidus : l'alignement précédant étant utilisé comme initialisation, plusieurs alignements successifs par programmation dynamique sont effectués en utilisant le score de distance S_{dis}. Cette étape est répétée jusqu'à convergence.

Un Z-score est calculé avec le score final de l'alignement, la moyenne et l'écart type étant calculés en comparant la structure requête avec toute la banque.

Méthodes utilisant les alphabets structuraux. Dans ces méthodes, les structures protéiques sont converties en suites de symboles qui correspondent à des classes de blocs structuraux. Les blocs structuraux sont le plus souvent de taille fixe, comportent quelques résidus (moins de 10) et sont

chevauchant. Ils sont classés en comparant toutes les sous-structures d'une taille donnée d'une banque non redondante de structure. Ces classes de blocs partitionnent le mieux possible l'espace de structures. Un symbole est très souvent affecté à chaque classe de blocs, c'est pourquoi on parle d'alphabets structuraux. Dans la méthode nommée SA-Search (Guyon et al., 2004), un modèle de loi normale multivariée a permis de discrétiser le squelette peptidique en classes de blocs structuraux de longueur 4 résidus (un critère d'information est utilisé pour déterminer le nombre de classes optimales). Les blocs structuraux – symboles d'un alphabet structural – sont décrits par 4 distances internes (voir Figure 2-18).

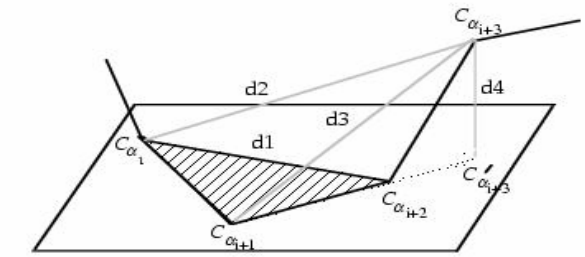


Figure 2-18 Chaque fragment de 4 résidus es représenté par un vecteur de taille 4 qui contient 3 distances entre $C\alpha$ non consécutifs (d1=d($C\alpha_i$ - $C\alpha_{i+2}$), d1=d($C\alpha_i$ - $C\alpha_{i+3}$), d1=d($C\alpha_{i+1}$ - $C\alpha_{i+3}$)) et la projection orthogonale d4 de $C\alpha_{i+3}$ sur le plan formé par les premiers $C\alpha$. Figure extraite du site http://bioserv.rpbs.jussieu.fr/Help/SAHelp.html

Cet alphabet structural comporte 27 symboles et permet une description précise de l'ensemble des structures (Camproux et *al.*, 2004). Après cette étape, l'enchaînement des blocs structuraux est évalué par un modèle de Markov entraîné sur une banque non redondante. Pour comparer les structures, celles-ci sont d'abord converties en suites de symboles de l'alphabet structural avec l'algorithme de Viterbi (Baum et al, 1970) qui permet de déterminer la meilleure série de symboles en prenant en compte la dépendance Markovienne. La comparaison des deux structures est ensuite faite soit par programmation dynamique (Smith et Watermann), soit par construction d'arbre de suffixes pour rechercher des motifs exacts. Ces deux méthodes sont assez rapides pour rechercher les similarités structurales dans une banque.

2.4.4 Comparaison multiple de structures

Les méthodes de comparaison multiple de structure s'appuient souvent sur des méthodes de recherche de motifs. L'objectif est de déterminer les éléments communs à plusieurs structures, ces éléments formant alors le(s) motif(s) caractéristique(s) du groupe de structures. De telles méthodes

sont assez répandues au niveau des séquences, mais il est plus difficile de les appliquer au niveau des structures. Les méthodes d'alignement multiple de structure ont été initialement conçues pour la RMN, où il fallait comparer les modèles obtenus entre eux afin de déterminer un cœur structural rigide d'une structure. Toutefois, lorsque des structures différentes sont comparées, comme pour les comparaisons de paires de structures, la définition de correspondances entre les éléments est nécessaire.

Un objectif des méthodes de comparaison multiple de structure est de déterminer des régions ou blocs structuraux conservés au sein d'une même famille de structures homologues. De tels blocs structuraux sont utilisés notamment pour la modélisation par homologie ou par reconnaissance de repliement. Plusieurs serveurs sont consacrés aux alignements structuraux multiples ; certains ont été présentés précédemment. La caractérisation de ces blocs structuraux conservés peut apporter aussi des informations sur les mécanismes d'évolution des structures protéiques ainsi que sur le repliement.

Les méthodes de comparaison multiple peuvent être regroupées en deux catégories : les méthodes procédant à des comparaisons de paires de structures et les méthodes réellement multiples procédant sans comparaison préalable des paires de structures.

2.4.4.1 Méthodes avec comparaison des paires de structures

De nombreuses méthodes abordées dans la section précédente ont été adaptées à la comparaison structurale multiple. Elles peuvent être subdivisées en trois sous-catégories : les méthodes avec une structure pivot (moyenne ou réelle), les méthodes d'alignement hiérarchique (avec construction d'un dendogramme) et les méthodes utilisant les graphes. Des méthodes de superposition multiple ont aussi été développées mais nécessitent une correspondance ou un alignement préalable. De manière comparable aux séquences biologiques, il faut noter que toutes les paires de structure ne sont pas forcément alignées de manière optimale dans un alignement optimal multiple : ces méthodes s'appuyant sur les alignements optimaux par paires, sont des heuristiques.

Méthodes d'alignement hiérarchique et méthode de structures moyennes. Dans ces méthodes, les structures sont représentées au niveau des résidus et la procédure générale s'effectue en trois étapes : (i) alignement de toutes les paires de structure ($N(N-1)/2$ alignements) ;(ii) construction d'un dendogramme à partir des scores calculés ; (iii) alignement de toutes les structures dans l'ordre indiqué par le dendogramme (en commençant par la paire de structures les plus similaires). On trouve parmi ces méthodes les programmes COMPARER (Sali et Blundell, 1990) ou encore STAMP (Russell et Barton, 1992) qui ont servi pour établir la base HOMSTRAD. Dans ces deux cas, l'alignement d'une structure avec un ensemble de structures déjà alignées se fait par programmation dynamique. MATRAS utilise également ce principe pour l'alignement multiple. La similarité entre toutes les

paires de structures est calculée par alignement, comme décrite précédemment, et un arbre est ensuite construit par la méthode UPGMA. L'alignement entre deux groupes suit celui entre les deux structures appartenant à chacun des groupes qui alignent le mieux.

2.4.4.2 Méthodes avec structure pivot ou moyenne

Dans ce genre de méthode, une structure réelle ou calculée va servir de référence pour aligner les autres structures. Plusieurs méthodes sont proposées ici, comme celle de Gerstein et Altman (Gerstein et Altman, 1995) qui ont développé une méthode pour trouver un cœur structural d'une famille de N structures. Pour ce faire, ils n'utilisent pas de blocs structuraux mais de résidus en équivalence (pouvant être discontinus) sur l'ensemble de chaque structure. Un alignement multiple (ou cœur) initial étant donné, deux étapes suivantes sont alors répétées : (i) calcul de la position moyenne de tous les éléments de ce cœur et superposition de toutes les structures par rapport à cette structure moyenne ; (ii) calcul de la variation des positions de toutes les structures par rapport à la structure moyenne, pour ôter la position ayant la plus forte variation. Les deux étapes sont répétées jusqu'à ce qu'il n'y ait plus de résidu dans le cœur. L'analyse statistique des résultats de cette procédure permet d'identifier le cœur.

Une autre méthode décrite par Gerstein et Levitt (Gerstein et Levitt, 1996, 1998) utilise la méthode itérative classique de superposition-alignement par programmation dynamique pour aligner les paires de structures autour d'une structure « médiane » choisie, celle qui est en moyenne la plus proche des autres structures. Après superposition, de nouveaux alignements entre la structure médiane et les autres structures sont calculés par programmation dynamique, le score dépendant de la distance externe entre les C_α. Ces alignements sont ensuite combinés pour former l'alignement multiple et les structures sont à nouveau superposées selon ces nouvelles correspondances entre C_α.

Il existe bon nombre d'autres méthodes, mais les deux exemples présentés ci-dessus illustrent bien les méthodes avec structure moyenne (pour la première méthode) ou pivot (pour la seconde méthode).

2.4.4.3 Méthodes sans alignement des paires

Ces méthodes sont similaires dans leur concept à celles qui ont déjà été vues : il y a les méthodes utilisant les graphes, les méthodes de hachage géométrique et les méthodes de recherche de **motifs répétés**. Parmi ces derniers, trois logiciels, dont deux importants dans mon travail de thèse, peuvent être cités : GOK, YAKUSA et GAKUSA. Dans ces trois outils, la conformation des squelettes

peptidiques est décrite par une suite d'angles (ϕ,ψ) ou (α,τ). Les valeurs des angles pouvant être échantillonnées suivant un pas –ou maille– δ, la conformation du squelette peut alors être décrite par une chaîne de « caractères » d'un alphabet fini comptant $360/\delta$ symboles. La recherche de similarités structurales entre structures se ramène alors à la recherche de motifs répétés dans des chaînes de symboles constituées de la concaténation des structures.

2.4.5 Recherche de similarités structurales locales développé à l'ABI

2.4.5.1 Logiciel GOK

GOK utilise une méthode nommée KMRC est inspiré de l'algorithme de recherche de motifs répétés exacts de Karp, Miller et Rosenberg, nommé algorithme de KMR (Karp et al, 1972). Afin de comprendre les expressions qui seront utilisées ultérieurement, il est nécessaire de définir préalablement les termes utilisés

Nomenclature. L'alphabet sera nommé Σ et le « texte » dans lequel les motifs sont cherchés *s*. Le symbole en position *i* dans le texte sera noté *s*[i]. Un motif de longueur k sera désigné par ***k*-motif**, et un ***k*-motif répété**, un motif répété au moins une fois dans au moins *q* structures, *q* étant le **quorum**[4]. Dans le cas de motifs exacts, une **instance** d'un motif représente ce motif. Par contre, dans le cas de motifs approchés, une instance ne représente pas le motif car elle est « stricte ». Une distinction se fera alors entre motif et instance d'un motif. Les **occurrences** *p* d'un motif sont les positions de ses instances. L'ensemble des occurrences d'un motif sera nommé **extension**.

Description. La recherche de motifs répétés est un problème connu et très étudié. Beaucoup de solutions ont été proposées dont certaines permettent de trouver des motifs approchés (Crochemore et Rytter, 1994). Dans l'algorithme de KMR, la construction des motifs répétés exacts est progressive : les motifs de taille *k* permettent de construire les motifs de taille 2*k*. En effet, si un motif de taille *k* a pour extension $\{i,j\}$ et qu'un second motif répété a pour extension $\{i+k,j+k\}$, il est possible de combiner ces deux motifs pour former le motif de taille 2*k* présent en *i* et *j*. La recherche des motifs répétés est donc faite en largeur d'abord.

Cependant, les motifs construits à chaque étape ne respectent pas obligatoirement la condition de quorum, ils ne sont pas toujours répétés. Soient par exemple deux *k*-motifs, un dont l'extension est $\{i,j\}$ et l'autre dont l'extension est $\{i+k,l\}$ avec $l \neq j + k$. La combinaison de ces deux *k*-motifs aura pour

[4] Dans tous les exemples, le quorum est de 2.

extension $\{i\}$, le nouveau motif n'existant pas en j. Ce nouveau $2k$-motif n'est pas un motif répété, il doit donc être éliminé. Il n'est pas possible de savoir quel motif sera répété avant de les avoir construits. Une étape de filtrage des motifs répétés – *i.e.* ne respectant pas le quorum – est donc nécessaire après chaque étape de construction.

Il sera bien sûr dommage de ne construire que des motifs de taille multiple de deux mais il n'est pas obligatoire que deux motifs soient contigus pour être combinés : ils peuvent être chevauchants et donner naissance à des motifs de taille intermédiaire (entre k et $2k$). En théorie, il est aussi possible de combiner des motifs de taille variable. Cependant, il est plus simple qu'à une étape donnée, tous les motifs aient la même taille. Un des intérêts de cet algorithme est qu'à chaque étape, seuls les motifs existants réellement son créés et traités.

La méthode proposé par H Soldano et coll. (Soldano et *al.*, 1995) est une généralisation de cet algorithme permettant de trouver des motifs approchés. Pour trouver des motifs approchés, certains symboles sont considérés comme similaires. Soit par exemple l'alphabet $\Sigma = \{a,b,c\}$, a est similaire à b et b est similaire à c, mais a n'est pas similaire à c. L'instance de 2-motif « *aa* » est similaire à « *ab* » et « *ab* » est similaire à « *ac* » mais « *ac* » n'est pas similaire à « *aa* ». Ces similarités peuvent être représentées par un pavage de l'espace des symboles : l'espace de l'alphabet Σ est découpé en **pavés** recouvrant G_i tels que l'ensemble des pavés $G = \{G_1,G_2,\ldots G_{|G|}\}$ avec $G_i \subseteq \Sigma$ pour $1 \leq i \leq |G|$ et aucun $1 \leq i,j \leq |G|$ avec $i \neq j$ tel que $Gi \subseteq Gj$. Ces pavés peuvent donc être chevauchants mais ne peuvent pas se recouvrir totalement. Les symboles dans un même pavé sont similaires. Dans notre exemple, l'alphabet est découpé en deux pavés : $G_0 = \{a,b\}$ et $G_1 = \{b,c\}$. L'alphabet utilisé pour décrire les motifs sera pour les motifs approchés cet ensemble de pavés G_i, aussi nommé **alphabet dégénéré**. L'algorithme de construction des motifs suit le même principe que l'algorithme pour les motifs exacts : les motifs de taille k permettent de construire les motifs de taille $2k$. Il faut cependant noter que – puisqu'un symbole peut appartenir à plusieurs pavés – plusieurs motifs différents peuvent exister en une même position. Toujours avec le même exemple, l'instance « *ab* » est similaire à « *aa* » et à « *ac* » mais ces deux instances sont différentes. L'instance « *ab* » appartient donc à deux 2-motifs, qui sont les motifs G_0G_0 et G_0G_1 lorsqu'ils sont écrits avec l'alphabet dégénéré.

Un effet de cette dégénérescence est que beaucoup plus de motifs sont générés à chaque étape et certains des motifs ont une extension incluse dans l'extension d'un autre motif. Prenons par exmple le

texte *s* « *abba* », le même alphabet et les mêmes pavés que précédemment. Le symbole *b* appartient à plusieurs pavés, le texte peut don être représenté de la manière suivante :

positions	1	2	3	4
symboles	*a*	*b*	*b*	*a*
pavés	G_0	G_0 G_1	G_0 G_1	G_0

Quatre 2-motifs sont présents : G_0G_0 dont l'extension est {1,2,3}, G_1G_0 dont l'extension est {2,3}, G_0G_1 dont l'extension est {1,2}, et G_1G_1 dont l'extension est {2}. Ce dernier motif ne respecte pas le quorum, il est donc éliminé. Par ailleurs, les deux motifs G_0G_1 et G_1G_0 ont chacun une extension contenant deux occurrences, mais ces extensions sont chacune incluses dans l'extension du premier motif G_0G_0. Ces deux motifs sont donc éliminés et seul le premier motif, G_0G_0 est conservé. Ce motif est dit **maxim*al.*** Les motifs maximaux seuls sont suffisants pour construire tous les motifs maximaux de taille supérieure.

Cette méthode a été appliquée aux structures protéiques. Les symboles sont des angles discrétisés selon une « maille » δ donnée. Les symboles sont alors des entiers représentant un intervalle d'angles. Cependant, si $\delta = 10°$, un angle de 8° est représenté par le symbole « 0 » car il fait partie de l'intervalle [0°,10°[, il est alors considéré comme différent de l'angle 11° car ce dernier est représenté par un symbole « 1 » (intervalle [10°,20°[). Or, ces deux angles devraient être similaires. La similarité pour les angles fait intervenir un paramètre entier κ appelé « marge ». Si $\kappa = 1$, le symbole 1 est similaire aux symboles 0 et 2. Deux angles sont donc considérés comme similaires si leur différence angulaire est inférieur à $(2 \times \kappa + 1) \times \delta$.

La complexité de cet algorithme est : $O(n.k_{max}.g^{2kmax}.logk_{max})$ avec n la longueur totale de la structure, k_{max} la longueur maximale des motifs répétés, g la mesure de la dégénérescence ou nombre maximal de pavés auxquels un symbole de l'alphabet peut appartenir. Dans l'exemple précédent, la dégénérescence est de 3. Par exemple, le symbole 2 appartient aux pavés G_0 = {0,1,2}, G_1 = {1,2,3}, G_2 = {2,3,4}

Cette méthode a été appliquée à la recherche de blocs structuraux similaires dans les cytochromes P450 par P. Jean et coll. (1997) en vue de la modélisation de la structure du $P450_{eryF}$.

2.4.5.2 Logiciel GAKUSA

GAKUSA est un autre logiciel, à l'instar de GOK, représentant la structure d'une protéine par sa description en terme d'angles de coordonnées internes. Contrairement à GOK, GAKUSA repose sur un algorithme de « Gibbs Sampling » (Lawrence et al, 1993). Cet algorithme permet de générer des blocs d'alignement multiple sans passer par l'étape d'alignement par paires. Supposons qu'un bloc structural similaire de taille *l* est recherché dans les *m* structures recodées en angles α discrétisés (symboles). L'algorithme suivant est alors répété plusieurs fois :

- Un segment de taille *l* est d'abord choisi au hasard dans chacune des *m* structures ;
- puis, les deux étapes suivantes sont répétées jusqu'à convergence :
  1. une structure est exclue et les probabilités de chaque angle α_i à chaque position *j* dans les segments des *m-1* autres structures sont calculées selon la formule :

$$q_{i,j} = \frac{c_{i,j} + b_j}{m - 1 + \beta}$$

Avec $c_{i,j}$ le nombre d'angles α_i en position *j* des segments, b_i les *pseudo-comptes* pour l'angle α_i, et *m-1* le nombre de structures considérées ici. Les « pseudo-comptes » b_i sont évalués comme $b_i=\beta\rho_i$, ρ_i étant la fréquence des angles α_i dans les toutes structures et β une constante (en général $\beta = \sqrt{(m-1)}$) Les fréquences p_i des α_i hors des segments sont aussi calculées selon cette formule ;

  2. dans la structure exclue précédemment, pour tous les segments de taille *l*, les ratio $q_{i,j}/p_i$ sont calculés pour toutes les positions *j*, puis normalisés en probabilités. Une position au hasard est ensuite tirée selon ces probabilités. Le segment correspondant devient la nouvelle position du segment pour cette structure.

Dans la méthode originale de « Gibbs sampling » décrite ci dessus, seule l'identité des angles est prise en compte et non la similarité : ceci conduit à une estimation trop stricte et donc infructueuse des blocs structuraux similaires. Il a été décidé d'ajouter dans l'étape 1) un terme supplémentaire de « pseudo-comptes » basé sur la similarité entre les angles : si un angle à une position donnée est compté, une fraction de ce comptage est rajoutée dans le comptage des angles voisins de cet angle. La nouvelle formule s'écrit alors de la manière suivante :

$$q_{i,j} = \frac{\sum_{k=angles} \gamma_{jk} c_{i,k} + b_j}{m - 1 + \beta}$$

Où $c_{i,k}$ est toujours le nombre d'angles α_k en position i et γ_{ik} est la probabilité de l'angle discrétisé k dans une gaussienne de moyenne i et d'écart type choisi (plus l'écart type est grand, plus la similarité entre angles différents est élevée). Ceci permet donc de prendre en compte la similarité entre les angles α discrétisés, indispensable lorsqu'on travaille sur les structures.

En fait, cette méthode de « pseudo-comptes de similarité » - développée pour les structures - peut être étendue avec profit à l'alignement multiple de séquence d'acides aminés (domaine original d'application de la méthode de « Gibbs sampling »). En effet, les « pseudo-comptes » supplémentaires de similarité sont alors basés sur les fréquences de substitution (fréquences de mutabilité) tirées d'une matrice de similarité de type BLOSUM ou PAM. Cette extension permet d'améliorer la reconnaissance de blocs similaires difficiles à exhiber par le « Gibbs sampling » classique : on ajoute de l' « information biologique » quant à la substituabilité des acides aminés.

2.4.5.3 Logiciel YAKUSA

YAKUSA décrit est le troisième outil développé à l'ABI, utilisant les angles de coordonnées internes pour décrire une structure. Il utilise cette fois ci cette description dans un algorithme de recherche de motifs, dérivé de l'algorithme d'Aho-Corasik (Aho et Corasick, 1975). Un automate est construit avec tous les petits motifs (appelés « graines ») d'une taille donnée de la séquence requête et leurs « voisins ». Un motif voisin est un motif qui n'existe pas dans la structure requête, mais qui est suffisamment similaire à un motif de la requête du point de vue de sa conformation. De par sa structure, cet automate permet de rechercher tous les motifs « graines » qu'il contient, en temps linéaire sur la banque de structures, et tous les motifs sont recherchés en parallèle. Pour chaque protéine de la banque, les motifs « graines » communs avec la structure requête sont étendus si possible, donnant ainsi des motifs communs plus grands. Un classement est ensuite effectué et donne les protéines montrant les plus grandes similarités en structure avec la structure requête. Le principal avantage de cette méthode, par rapport aux quelques méthodes existantes, est sa rapidité : le « scan » par une structure requête des 1700 structures d'une banque non redondante tirée de la PDB est effectué en 1 à 2mn sur un PC de bureau. Un autre avantage est que les similarités structurales exhibées sont locales, ce qui permet la mise en évidence de sous structures communes, même si il n'y a pas de ressemblance globale des deux protéines.

2.5 Construction d'un modèle

On doit le principe de reconstruire un modèle de protéine à partir de sa séquence aux travaux d'Anfisen qui, pour comprendre le mécanisme de repliement des protéines, a émis au cours des années 60 l'hypothèse que la conformation native, fonctionnelle, d'une protéine correspond à celle qui présente l'énergie libre la plus basse. Par conséquent, si le repliement s'effectue sous des contraintes thermodynamiques, cette conformation ne devrait dépendre que de la séquence en acides aminés de la protéine (Anfinsen, 1973). Cette hypothèse est fondamentale dans la mesure où elle laisse supposer que la seule connaissance de la séquence en acides aminés de la protéine devrait – en théorie – être suffisante pour obtenir sa structure 3D. De nombreux travaux sont venus nuancer l'hypothèse d'Anfinsen sans pour autant la remettre totalement en cause, notamment avec l'addition au contrôle thermodynamique d'un contrôle cinétique (lié à la vitesse de repliement) et avec l'identification de protéines assistant le repliement comme les protéines chaperons. Aujourd'hui, le problème du repliement protéique est encore loin d'être résolu. Par contre, il a conduit à l'émergence de méthodes bioinformatiques visant à prédire la structure d'une protéine à partir de sa séquence en acides aminés. Ces méthodes constituent une véritable alternative aux méthodes expérimentales de détermination de structures pour réduire l'écart entre le nombre de structures et de séquences connues.

Les différents traitements de données sources décrites dans le paragraphe précédent sont nécessaires à l'obtention d'un modèle qui copie au mieux la réalité. Selon les données à disposition et les traitements qu'on a pu leur faire subir, différentes techniques de reconstruction sont possibles pour prédire la structure d'une protéine à partir de sa séquence. Les méthodes de prédiction *in silico* de la structure 3D des protéines à partir de la séquence en acides aminés sont généralement regroupées en trois grandes catégories : la modélisation par homologie (ou comparative), les méthodes de reconnaissance de repliement (ou d'enfilage) et les méthodes *ab initio* / *de novo*. Le choix de la méthode dépend de l'existence ou non dans la PDB d'une protéine de séquence similaire à celle de la protéine à modéliser (ou protéine homologue), et du taux d'identité de séquences entre ces protéines (homologie plus ou moins distante).

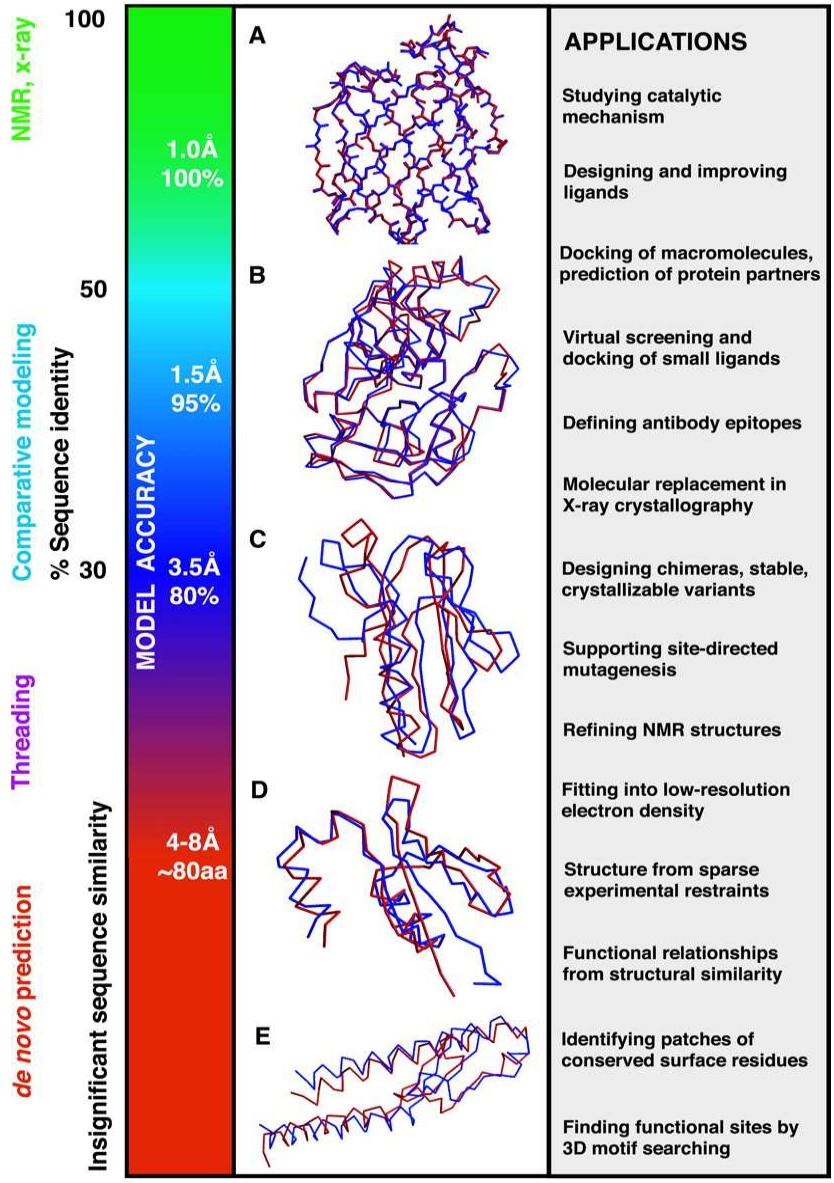


Figure 2-19 Précision et applications possibles des modèles protéiques. L'application des méthodes de modélisation par homologie (ou comparative), d'enfilage (ou threading) et *ab initio* / *de novo* dépend principalement du taux d'identité de séquence avec des structure connue. La précision des modèles diminue lorsque ce taux diminue : la valeur du RMSD des atomes de la chaîne principale du modèle (en rouge) par rapport à la vrai structure (en bleu) augmente. Lorsque le taux d'identité de la séquence se situe entre 30% et 50% par exemple, les modèles construits par homologie ont généralement environ 90% des atomes de la chaîne principale à 1,5Å de la vraie structure. Différentes applications sont possibles en fonction de la précision des modèles. Figure extraite de (Baker et Sali, 2001)

Des problèmes majeurs sont liés à l'évaluation des performances des méthodes de prédiction développées et à leur comparaison. Une solution à ces problèmes a été la mise en place d'une réunion nommée CASP (Critical Assessment of Structure Prediction) qui a lieu tous les deux ans (Moult et *al.*, 1995). Cette dernière est née d'une volonté de tester la fiabilité des méthodes développées en évaluant leurs performances sur un même jeu de protéines, et avec les mêmes critères. Elle consiste à proposer aux différents groupes de recherche d'appliquer leurs méthodes sur des protéines dont la structure vient d'être déterminée expérimentalement mais n'est pas encore déposée dans la PDB, ni publiée. Les méthodes sont donc appliquées en aveugle. A la fin de la compétition, les différents modèles proposés sont évalués par rapport à la vraie structure 3D. La première compétition CASP (ou CASP1) date de 1994. Les résultats publiés les plus récents sont ceux de CASP6 qui a eu lieu en 2004 (Moult et *al.*, 2005). Ceux de CASP7 qui s'est déroulé en Novembre dernier à Asilomar Conference Center en Californie devraient être publiés cette année. Cette compétition permet d'évaluer les progrès réalisés au cours des années dans les différentes catégories de méthodes de prédiction de structure (Ginalski et *al.*, 2005 ; Moult, 2005).

La précision des modèles obtenus par les différentes méthodes de modélisation ainsi que leurs applications possibles sont résumées dans la Figure 2-19.

2.5.1 Modélisation par homologie

2.5.1.1 Principe

La modélisation par homologie (ou modélisation comparative) nécessite l'existence d'une protéine de structure connue présentant plus de 30% d'identité de séquence avec la protéine à modéliser (protéine cible), et qui pourra servir de structure de référence (Marti-Renom et *al.*, 2000). En effet, ce fort taux d'identité de séquence laisse supposer que les deux protéines sont homologues et qu'elles adoptent des structures tridimensionnelles proches (Chlothia et Lesk, 1986).

Différentes étapes sont nécessaires. (i) Une structure de référence est sélectionnée pour servir de support. (ii) Les séquences des protéines cible et support sont alignées. Le modèle peut alors être construit avec (iii) la modélisation des régions conservées de la chaîne principale, (iv) la modélisation des boucles, puis (v) le positionnement des chaînes latérales. Les dernières étapes correspondent à l'optimisation et à la validation du modèle.

Actuellement, cette méthode est celle qui permet d'obtenir les modèles les plus précis. Toutefois, ces méthodes sont loin d'être parfaites, de nombreux problèmes majeurs restent à traiter. L'alignement

des séquences par exemple est particulièrement délicat à réaliser lorsque le taux d'identité de séquence diminue (Venclovas, 2003). De même, la modélisation des boucles, régions structurellement divergentes, est complexe. De nombreuses études proposent des méthodes pour prédire la conformation des boucles, soit ab initio (Fiser et *al.*, 2000 ; Galaktionov et *al.*, 2001), soit à partir de banques de boucles (Donate et *al.*, 1996 ; Oliva et *al.*, 1997 ; Rufino et *al.*, 1997 ; Wojcik et *al.*, 1999 ; Michalsky et *al.*, 2003), ou encor par des méthodes mixtes (Deane et Blundell, 2001).

2.5.1.2 Quelques exemples

CASP donne un bon aperçu des différentes méthodes utilisées en modélisation comparative. Toutes les méthodes présentes plus ou moins leurs avantages et leurs défauts, les facteurs pouvant affecter la qualité d'un modèle sont souvent soit le choix de la séquence de départ (et donc le nombre d'insertions – délétions qui apparaîtront au cours de l'étape d'alignement) soit le logiciel utilisé, soit enfin l'expérience de la personne qui reconstruit la protéine. Outre les méthodes manuelles, on citera ici trois logiciels, les plus communs, et surtout ceux que j'ai utilisé durant ma thèse : COMPOSER (Sutcliff et *al.*, 1987), SwissModel (Schwede et *al.*, 2003) et Modeler (Sali et Blundell, 1993).

COMPOSER est un programme de modélisation comparative utilisé dans SYBYL, un ensemble de programmes distribué par Tripos. Il correspond plus ou moins à une automatisation de la méthode manuelle classique de reconstruction par homologie, et procède en six étapes : La première est une étape de recherche d'homologue en séquence sur la PDB. La seconde étape consiste ensuite à déterminer les graines (« seeds »), à savoir des résidus à topologie équivalente dans l'ensemble du set de « structures homologues ». Cette identification se fait sur la similarité de séquence. En troisième étape, un alignement structural est réalisé, à partir des graines déterminées dans l'étape précédente. C'est à cette étape que des régions structuralement conservées (RSC) sont déterminés, donnant une structure moyenne de RSC. Ces RSC sont trouvés par calcul de RMSD entre les $C\alpha$. Les séquences des structures avec la séquence cible sont alors alignées. Au cours de la cinquième étape, le modèle est d'abord reconstruit au niveau des RSC, et enfin, pour les régions plus variables, COMPOSER se sert d'une base de boucles et de rotamères pour les chaînes latérales. Une fois le modèle construit, il faudra alors procéder à son raffinement, et son évaluation.

SwissModel est quant à lui un serveur qui propose de prédire la structure d'une protéine à partir d'une séquence fournie par l'utilisateur. Contrairement à COMPOSER, SwissModel est un logiciel académique, qui ne nécessite donc aucune licence pour utilisation. Il procède en quatre étapes. La première étape est similaire à celle de COMPOSER, c'est-à-dire la recherche de « *templates* » qui

serviront de références pour reconstruire le modèle. Cette recherche se fait à l'aide d'un BLASTP sur une base ExNRL-3D, une base de séquences dont les structures sont connues. La seconde étape consiste à définir les zones à modéliser. Un programme, SIM, ne retient que les séquences dont l'identité locale de séquence est supérieure à 25% et définit des fragments de la séquence cible qui seront modélisés en fonction de la similarité des séquences *templates*. Suite à ce travail, un alignement structural des *templates* est réalisé avec construction du modèle. Pour cela, le programme ProMod (Peitsch, 1996) est sollicité pour aligner les structures 3D des *templates*. Les parties structurales des *templates* proches spatialement sont alors utilisées pour définir un nouvel alignement structural en séquences primaires des *templates*. La séquence de la protéine à modéliser est alors alignée sur cet alignement multiple. Les parties de la séquence cible qui n'ont pu être alignées (généralement les boucles) sont alors modélisées à partir d'une bibliothèque de fragments issue de la Protéine Data Bank. Les chaînes latérales manquantes sont ensuite reconstruites et celles incorrectes, corrigée. Enfin, une minimisation énergétique est réalisé sur la structure obtenue, assurée par le logiciel GROMOS96 (Scott et *al.*, 1999). La validation du modèle se fait alors par l'outil Profil 3D. Il existe des limites à l'utilisation de ce serveur, qui sont d'au moins 30% d'identité de séquence sur au moins 60% de la longueur de la séquence.

Modeller est certainement le logiciel de modélisation comparative le plus utilisé dans la communauté scientifique. Gratuit à usage académique, c'est un programme entièrement automatisé : l'utilisateur doit fournir un alignement de séquence (réalisé par ses propres soins) et le programme se charge de calculer un modèle (sans atome d'hydrogène) en satisfaisant à la fois les contraintes spatiales, la stéréochimie, les angles, les torsions, liaisons hydrogènes etc…. Les contraintes sont en fait issues d'une analyse statistique des relations entre paires de structures homologues. Cette analyse repose sur une banque de données de 105 alignements de familles qui incluent 416 protéines de structures connues. En scannant la base, des tables quantifiant diverses corrélations sont obtenues, comme celles entre deux distances équivalentes $C\alpha - C\alpha$, ou entre deux angles dièdres équivalents. Ces relations sont exprimées en tant que fonctions de probabilité de densité (PDF) et peuvent être utilisées directement comme des contraintes spatiales. Celles-ci sont donc obtenues de manière empirique, à partir de banque d'alignement de structure protéique. Ces contraintes sont combinées avec les énergies de CHARMM[5] en une fonction objective. Finalement, un modèle 3D est obtenu après optimisation de la fonction objective dans le plan cartésien. Cette optimisation fait appel à

[5] Les champs de forces seront expliqués un peu plus tard dans le manuscrit

plusieurs méthodes dont le gradient conjugué, la dynamique moléculaire et le recuit simulé. Enfin, le programme est également en mesure de reconstruire *de novo* les boucles des structures protéiques.

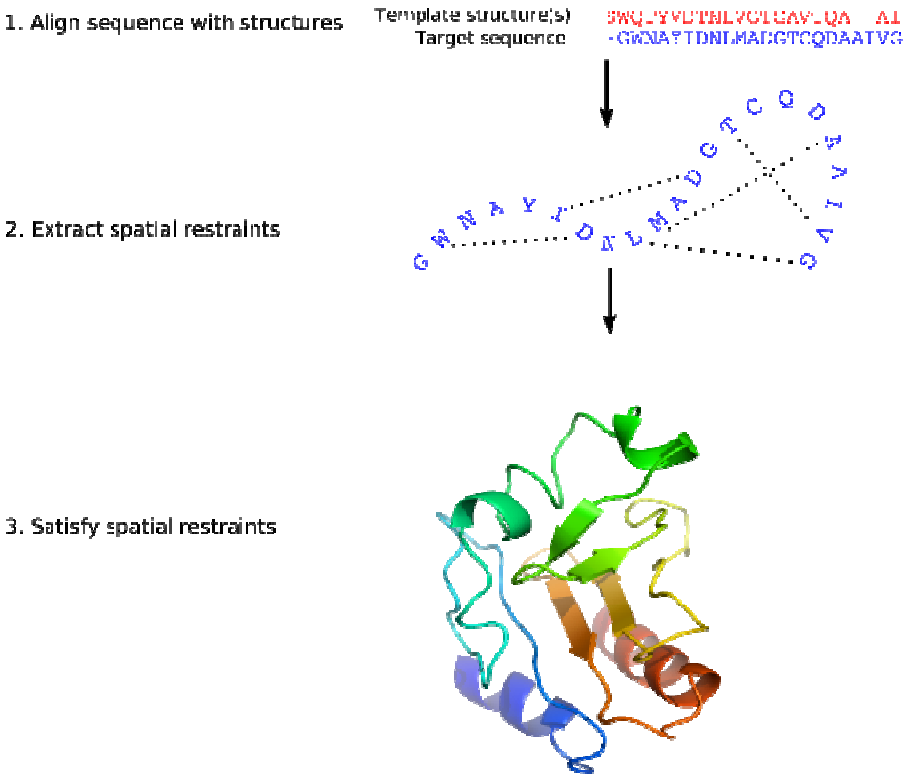


Figure 2-20 D'abord, les *templates* sont alignés avec la séquence cible à modéliser. Puis, les contraintes de distances, d'angles etc. sont transféré à la cible afin d'obtenir des contraintes spatiales pour cette structure à construire. Enfin, un modèle 3D est construit en essayant de satisfaire ces contraintes autant que possible.

2.5.2 Méthodes de reconnaissance de repliement

La modélisation devient beaucoup plus difficile lorsque le taux d'identité de séquence diminue et qu'aucun homologue proche de structure connue n'est disponible. Les méthodes de reconnaissance de repliement, qui cherchent à mettre en évidence des similitudes structurales lorsque l'identité de séquence est beaucoup plus faible, sont une alternative. L'idée consiste à identifier parmi les repliements connus celui qui pourrait adopter la séquence cible. Comme il l'a été déjà évoqué, le nombre de repliements protéiques est limité, une nouvelle protéine a par conséquent de grandes chances d'adopter un repliement déjà connu. Ces méthodes sont également appelées méthodes d'enfilage (ou « threading » : de manière imagée, la séquence de la protéine à modéliser est « enfilée » dans des repliements connus afin de déterminer celui qui lui correspondrait le mieux par calcul d'un

score de compatibilité. Ces méthodes sont généralement applicables entre 15% et 30% d'identité de séquence. Leurs principales difficultés résident dans la réalisation des alignements de séquence – structure et dans l'évaluation des repliements potentiels. De plus, elles sont limitées par le nombre de repliements uniques connus (Sippl et *al.*, 2001 ; Kinch et *al.*, 2003).

2.5.3 Méthodes ab initio / de novo

Finalement, les méthodes *ab initio* cherchent à prédire la structure d'une protéine à partir de sa seule séquence en acides aminés (sans utilisation de structure de référence). Elles constituent donc un défi scientifique majeur et trouvent tout leur intérêt dans ce contexte de séquençage à grande échelle qui génère un grand nombre de séquences pour lesquelles aucun homologue de structure connue n'est disponible ou tout du moins trop difficile à déceler. Il convient ici de distinguer les méthodes *ab initio* « pures » des méthodes dites *de novo*. Les méthodes *ab initio* « pures » reposent uniquement sur les propriétés physico-chimiques des protéines et sur l'hypothèse que la structure 3D adoptée par une protéine (ou structure native) est celle qui est la plus stable (d'énergie la plus faible) parmi l'ensemble des structures possibles (même si quelques contre-exemples existent) (Bonneau et Baker, 2001). Les méthodes *de novo* vont quant à elles exploiter des informations statistiques obtenues de l'analyse des structures 3D connues (Skolnick et *al.*, 2003).

Actuellement, les résultats les plus encourageants sont obtenus par des méthodes de modélisation *de novo* construisant des modèles protéiques à partir de courts fragments de structure. La plus connu est ROSETTA développé par le groupe de Baker et cool. (Simons et *al.*, 1997). On trouve également parmi ces méthodes de reconstruction l'utilisation de bibliothèques de fragments comme ceux présentés pour l'alphabet structural.

2.5.4 Évaluation des modèles

Afin d'interpréter et exploiter les modèles 3D de protéines, il est essentiel d'estimer leur précision, aussi bien globale que locale à certaines régions du modèle. Les erreurs dans les modèles proviennent de deux sources principalement : (i) l'échec de la recherche conformationnelle pour trouver la conformation optimale ainsi que (ii) l'échec de la fonction de scoring à identifier la conformation optimale. Les modèles 3D sont généralement évalués selon des préférences géométriques des résidus ou des atomes, qui sont dérivées de structures protéiques connues. En pratique, on a coutume d'aborder une évaluation d'un modèle donné dans une manière hiérarchique. Dans un premier temps, il est nécessaire d'évaluer si le modèle dispose au moins d'un repliement (« fold ») correct. Le modèle

aura un repliement correct si le « bon » *template* est utilisé et correctement aligné sur la séquence cible. Une fois le repliement du modèle confirmé, une autre évaluation détaillée sur la précision globale de la protéine peut être effectuée, basée sur la similitude globale de séquence sur lequel le modèle est basé. Finalement, une variété de profils d'erreurs peut être construite pour quantifier les erreurs possibles dans les différentes régions du modèle. Une bonne stratégie est d'utiliser différentes méthodes d'évaluation de modèles et d'en sortir un consensus. De plus, les fonctions d'énergies sont généralement développées pour travailler sur un certain degré de détails, et ne sont pas appropriées pour juger les modèles à un degré plus fin ou plus grossier (Park et *al.*, 1997). Il existe un nombre important de programmes ou de serveurs d'évaluation (cf. Tableau 2.3).

Les modèles ont besoin d'une stéréochimie correcte. Les programmes les plus usuels pour évaluer la stéréochimie sont PROCHECK (Laskowski et *al.*, 1993), AQUA (Laskowski et *al.*, 1996), SQUID (Oldfield, 1992) et WHATCHECK (Hooft et *al.*, 1996). Ces programmes vérifient les caractéristiques suivants : les longueurs des liaisons (« bond length »), les angles de liaisons, les liaisons peptidique et la planéité des cycles des chaînes latérales, la chiralité, les angles de torsion pour la chaîne principale et les chaînes latérales, et les collisions entre les paires d'atomes non liés. En plus d'une bonne stéréochimie, un modèle doit également avoir une faible énergie en terme d'énergie de champs de force comme CHARMM, AMBER ou encore GROMOS. Toutefois, une faible énergie de mécanique moléculaire ne signifie pas forcément que le modèle est correct (Novotny et *al.*, 1984,1988). Ainsi, les distributions de plusieurs caractéristiques spatiales ont été compilées à partir de structures de protéines à haute résolution et chaque déviation aussi large soit-elle de la valeur la plus « probable » est interprétée comme une indication forte d'erreurs dans le modèle. De telles caractéristiques incluent l'empaquetage (Gregoret et Cohen, 1991), la formation de cœurs hydrophobes (Bryant et Amzel, 1987), l'accession au solvant (Chiche et *al.*, 1990 ; Holm et Sander, 1992), la distribution spatiale des groupes chargés (Bryant et Lawrence, 1991), la distribution des distances inter atomes (Colovos et Yeates, 1993), le volume atomique (Pontius et *al.*, 1996), les liaisons hydrogènes de la chaîne principale (Laskowski et *al.*, 1993).

D'autres groupes de méthodes pour tester les modèles 3D qui prennent en compte implicitement plusieurs critères cités ci-dessus, mettent en jeu des profils 3D et des potentiels statistiques (Sippl, 1990 ; Luthy et *al.*, 1992). Ces méthodes évaluent l'environnement de chaque résidu dans le modèle en respectant l'environnement attendu comme celui trouvé dans les structures X à haute résolution. Les programmes qui implémentent ces approches sont VERIFY3D (Luthy et *al.*, 1992), PROSA (Sippl, 1993), HARMONY (Topham et *al.*, 1994) et ANOLEA (Melos et Feytmans, 1998).

Tableau 2.3 Quelques exemples de programmes d'évaluation

Programme d'évaluation	Adresse internet	Commentaires
PROCHECK	http://www.biochem.ucl.ac.uk/~roman/procheck/procheck.html	Vérification de la stéréochimie
WHATCHECK	http://www.sander.embl-heidelberg.de/whatcheck/	Système de validation structurale
ProsaII	http://www.came.sbg.ac.at	Détermine le repliement natif de la protéine
ProCyon	http://www.horus.com/sippl/	
BIOTECH	http://biotech.embl-ebi.ac.uk:8400/	Suite de programme dévaluation comprenant PROCHECK, PROVE et WHATIF
VERIFY3D	http://www.doe-mbi.ucla.edu/Services/Verify3D.html	Outil d'évaluation pour le raffinement des structures RX
ERRAT	http://www.doe-mbi.ucla.edu/Services/Errat.html	Evaluation des structures RX
ANOLEA	http://www.fundp.ac.be/pub/ANOLEA.html	Evalue l'environnement non local des atomes lourds de la molécule
AQUA	http://www-nmr.chem.ruu.nl/users/rull/aqua.html	
SQUID	http://www.yorvic.york.ac.uk/~oldfield/squid	Programme d'analyse de cristallographie
PROVE	http://www.ucmb.ulb.ac.be/UCMB/PROVE.html	Evaluation par calcul du volume atomique

2.6 Exploitation du modèle

Une fois le modèle obtenu, on dispose enfin d'un support d'étude. Toutefois, ce modèle n'est pas encore exploitable : de même que pour les structures cristallographiques, il faut « relâcher »la molécule afin qu'elle adopte une conformation stable de basse énergie. Ce travail de relaxation peut être effectué par différentes méthodes dont (i) la chimie quantique (aussi appelée *ab initio*) dans laquelle seuls les électrons sont traités, (ii) la méthode semi-empirique qui étudie seulement les électrons de valence avec un hamiltonien plus simple qu'en *ab initio*, et (iii) la mécanique moléculaire qui ne traite pas de l'électronique du système mais utilise les lois de la mécanique newtonienne. En raison de la puissance de calcul demandé, les deux premières méthodes limitent la taille de molécules étudiées à quelques atomes. C'est la mécanique moléculaire qui représente la forme la plus courante des logiciels utilisés car elle est rapide, fiable et permet d'optimiser des molécules de poids très élevé (macromolécules), avec un temps de calcul raisonnable.

2.6.1 Mécanique moléculaire

La structure 3D d'une molécule va dépendre des interactions intramoléculaires et des interactions avec le solvant. La recherche des conformations consiste donc à déterminer les minima de l'énergie globale d'interaction qui en général est réduite aux interactions intramoléculaires. La surface d'énergie

d'une molécule peut être approchée à l'aide d'une fonction analytique dont les termes sont une somme de plusieurs fonctions analytiques aux constantes déterminées empiriquement. Celles-ci englobent généralement une série de terme qui sont les suivants : un terme d'élongation des liaisons, un terme de variation des angles, un terme d'énergie de torsion des angles dièdres, un terme prenant en compte les énergies d'interaction entre atomes non-liés (Van der Waals) , un terme prenant en compte les énergies d'interaction électrostatiques entre atomes non liés et enfin un terme prenant en compte les liaisons hydrogènes. Tous ces termes se décrivent sous la forme $K_s(r - r_0)$ ou $K_\theta(\theta - \theta_0)$ … K_s, K_θ sont appelées constantes de force. Ces constantes dépendent bien sûr du type d'atome, mais aussi de leur environnement. La liste complète de ces constantes est appelée **champs de forces**.

2.6.2 Champs de Forces

Un champ de force est une liste de constantes qui décrit les interactions entre atomes liés et non liés. L'élaboration d'un champ de force doit répondre à 2 critères : (i) tout d'abord un critère de simplicité pour pouvoir être calculée rapidement et ensuite (ii) un critère de précision pour calculer de manière acceptable les propriétés structurales et thermodynamiques des molécules. De tels champs de force sont apparus dés 1970 et continuent à évoluer aujourd'hui. On distingue dans les champs de force deux catégories d'interaction : (i) les interactions entre atomes liés correspondent à des énergies de déformation des liaisons, des angles de valence et de torsion des angles dièdres ; (ii) les interactions entre atomes non liés correspondent aux interactions de Van der Waals, électrostatiques, liaisons Hydrogène. En biologie, les champs de forces habituellement utilisés sont CHARMM, AMBER, OPLS et GROMOS.

2.6.2.1 Atomes liés

La combinaison des interactions entre atomes liés est appelée champs de force de valence : il s'agit d'une description simplifiée des forces du système, élaborée pour reproduire les propriétés structurales, thermodynamiques ou dynamiques des molécules.

Élongation des liaisons ("Stretching"). La déformation des liaisons est en général faible (de l'ordre de 0.05 Å) et est exprimée par un potentiel harmonique[6] décrit par la fonction suivante :

$$E_l = \frac{1}{2}\sum_{i=1}^{n} k_{r,i}(r_i - r_i^0)^2$$

[6] Il existe un autre potentiel pour les élongations des liaisons appelés Potentiel de Morse. Celui-ci prend en compte la dissociation, mais est nettement plus cher en temps de calcul, c'est pourquoi le potentiel harmonique est le plus utilisé.

Déformation des angles ("Bending"). La déformation des angles est également faible (de l'ordre de quelques degrés) et est exprimée par la fonction suivante :

$$E_\theta = \frac{1}{2}\sum_{ij} k_{\theta,ij}(\theta_{ji} - \theta_{ij}^0)^2$$

Torsion des angles dièdres ("Torsion"). Pour des atomes co-planaires, la torsion des angles dièdres est décrite par une fonction périodique développée en série de Fourrier :

$$E_\tau = \frac{1}{2}\sum_i A_{i,n}[1 + \cos(n\tau_i - \phi)]$$

2.6.2.2 Atomes non liés

Interactions de Van der Waals. L'énergie de Van der Waals entre 2 atomes *i,j* distants de r_{ij} comprend: (i) un terme attractif variant en $-1/r_{ij}^6$.Cette fonction est connue sous le nom d'énergie de dispersion de London $E_{disp} = -C_{ij}/r_{ij}^6$. Les coefficients C_{ij} sont établis pour les différentes paires d'atomes présents dans la molécule. (ii) Un terme répulsif variant en en $1/r_{ij}^{12}$, traduisant le recouvrement des nuages électroniques à courtes distances. Cette fonction (Lennard-Jones) est couramment appelée potentiel 6-12 :

$$E_{vdw} = \frac{1}{2}\sum_{ij}\left[A_{ij}\left(\frac{r_{ij}^0}{r_{ij}}\right)^{12} - B_{ij}\left(\frac{r_{ij}^0}{r_{ij}}\right)^6\right]$$

Interactions électrostatiques.

1. Il y a d'abord les calculs de charge-charge : l'énergie électrostatique entre 2 atomes *ij* de charges q_i et q_j est représentée par la loi de Coulomb

$$E_{el} = \sum_{ij}\frac{q_i q_j}{4\pi\varepsilon_{ij} r_{ij}}$$

   Où l constante diélectrique prend en général des valeurs de 2 à 4(1.5 couramment).Dans certains programmes elle peut également varier en $1/r_{ij}$.
2. Dans les interactions électrostatiques, on comptabilise également les calculs dipôle-dipôle

Énergie des liaisons hydrogènes. La liaison Hydrogène est une interaction de 2 à 5 Kcal/mole d'importance. Elle s'exerce entre un Hydrogène déficient en électrons et un atome de forte densité électronique (comportant des doublets).

2.6.2.3 CHARMM, AMBER et GROMOS

CHARMM (**Chemistry at HARvard Macromolecular Mechanics**) est le nom d'un ensemble de champs de force largement utilisés en biologie pour la dynamique moléculaire. C'est aussi le nom du programme de simulation et d'analyses de mécanique moléculaire associé (Brooks et *al.*, 1983 ; Mackerelle et *al.*, 1998). Le développement et la maintenance de CHARMM regroupent un réseau de collaborateurs à travers le monde autour de M.Karplus et de son groupe à Harvard. Des licences sont disponibles pour le milieu académique. La version commerciale de CHARMM, appelée **CHARMm** (avec un 'm' minuscule), est fournie par Accelrys. Il existe plusieurs types de champs de force CHARMM se différenciant tous par le numéro de version qu'ils portent, selon que l'on s'intéresse à l'étude de protéines ou d'acides nucléiques ou lipide. Les numéros de version de ces champs de force correspondent à la première version de CHARMM qui les a incorporés, cependant ils peuvent être utilisés avec n'importe quelle version ultérieure du programme, ainsi que par d'autres programmes de dynamique moléculaire compatibles, par exemple NAMD.

AMBER (Assisted Model Building and Energy Refinement) est une autre famille de champs de force pour la dynamique moléculaire de biomolécules. Il a été développé par le groupe de P. Kollman à l'Université de Californie (Duan et *al.*, 2003). AMBER est aussi le nom du « package », l'ensemble d'outils pour les simulations de MD, qui implémente ces champs de force. Il est maintenu par une active collaboration entre D. Case et ses collaborateurs. Comme pour CHARMM, il existe différents champs de force AMBER plus ou moins adapté pour un type de molécule en particulier.

GROMOS est le troisième champs de force largement utilisé pour les biomolécules (Van Gunsteren et *al.*, 1996). Il a été développé à l'université de Groningen et l'ETH de Züirch. C'est le logiciel GROMACS (Groningen Machine for Chemical Simulation) qui utilise principalement ce champ de force. Contrairement aux deux autres champs de force, GROMOS est un champ de force de type « atome unis » plutôt que de type tout atome, ce qui signifie par exemple qu'un groupe méthyle est considéré comme un seul atome. Il a été optimisé pour respecter les propriétés des phases condensées des alcanes. Cette simplification dans sa représentation des éléments fait de GROMACS le programme le plus rapide en simulation moléculaire.

2.6.3 Minimisation

Avant toutes simulations de dynamique moléculaire, on a coutume de procéder préalablement à une minimisation. Cette étape de minimisation consiste à diminuer l'énergie du système par optimisation géométrique jusqu'à obtention d'un minimum loc*al.* En effet, aucun algorithme de minimisation ne permet de garantir de façon absolue l'obtention d'un vrai minima (ou minimum global). Il existe néanmoins des techniques comme celles de « recherche conformationnelle » qui visent à extraire la molécule de son puits de potentiel.

Les algorithmes de minimisation consistent à chercher à partir de la géométrie initiale, des jeux de coordonnées cartésiennes qui réduisent à son minimum la somme de toutes les contributions énergétiques. Les méthodes couramment utilisées reposent sur la dérivée première de l'énergie potentielle (*Steepest descent* ou *Conjugate grandient*). Ces algorithmes consistent à rechercher la direction de la plus grande pente c.-à-d. celle sur laquelle l'énergie décroît le plus rapidement et à modifier la structure en conséquence. La direction imposée aux coordonnées suit alors celle indiquée par l'opposé au gradient d'énergie.

2.6.4 Dynamique Moléculaire

2.6.4.1 Définition

La dynamique moléculaire consiste à étudier la trajectoire d'une molécule en appliquant les lois de la mécanique classique newtonienne c'est-à-dire, à simuler les mouvements atomiques au cours du temps. Ces mouvements correspondent à des vibrations autour d'un minimum ou au passage d'un minimum à un autre minimum d'énergie. Ainsi la dynamique moléculaire permet de s'extraire d'un minimum loc*al.*

2.6.4.2 Description

Dans les simulations de dynamiques moléculaires, on souhaite connaître la position et la vitesse des particules à chaque pas de temps. Ce temps évolue de manière discrète au cours de la simulation. Le calcul des forces d'interaction s'effectue entre les atomes : il permet de déterminer l'évolution des vitesses, et donc des positions, en utilisant les lois de la dynamique classique de Newton discrétisées. Un des points important est la conservation d'énergie totale du système au cours de la simulation. La méthode utilisée pour calculer les forces d'interaction (ou le potentiel dont elles dérivent) caractérise une simulation. Par exemple on parle de **dynamique moléculaire *ab initio*** si le potentiel est calculé à

partir des calculs de la mécanique quantique. Si en revanche les forces dérivent d'un potentiel fixé empiriquement, on parlera de **dynamique moléculaire classique**.

La dynamique moléculaire s'applique aussi bien à l'étude structurale des molécules qu'à des systèmes en interaction de grande taille. Néanmoins, en raison des capacités limitées de calcul, le nombre de particules dans une simulation l'est aussi. Pour simuler un matériau infini dans une, deux ou trois dimensions, les particules sont placées dans un espace périodique : on parlera alors d'une boîte de simulation. Lors du calcul des forces, la périodicité de l'espace devra être tenu en compte. En pratique, on distinguera dans la force d'interaction des termes à courte portée, qui ne seront pas affectés par la périodicité, c'est-à-dire que seules les particules les plus proches seront prises en compte, et un terme à longue portée, qui devra en tenir compte. Le terme à longue portée est généralement de type coulombien et sera calculé par la somme d'Ewald.

2.6.5 Le docking (ou arrimage)

En biologie structurale, on s'intéresse au rapport entre la structure des molécules et leur fonction biologique. De manière générale, on peut dire que le domaine recouvre des questions relevant d'une part de la pharmacologie (conception de médicaments) et d'autre part de la biologie cellulaire (étude du fonctionnement de la cellule). Dans le premier cas, un récepteur étant connu, il s'agit de trouver un ligand, c'est-à-dire une (ou plusieurs) molécule(s) complémentaire(s). Dans le second, on s'intéresse plutôt à l'interaction entre macromolécules (protéines ou acides nucléiques) intervenant dans les cycles cellulaires. On distingue donc les interactions entre ligand-protéine et les interactions entre protéine-protéine. Une question centrale commune aux deux problématiques est le *docking*. Le *docking* est l'étude des interactions intervenant lors de la formation de complexes moléculaires. Il repose sur l'hypothèse que les ligands formant des interactions favorables avec le récepteur doivent avoir une affinité de liaison élevée. Ces interactions entre molécules sont non liantes et concernent donc : (i) les interactions VDW, (ii) les interactions électrostatiques, et (iii) les interactions hydrogènes. Pour le sujet de thèse, on s'attachera plus au *docking* ligand-protéine. La plupart des programmes de *docking* automatique effectuent une exploration systématique de l'espace des configurations pour générer et évaluer un grand nombre de liaisons potentielles. Les structures générées sont alors soumises à un critère de score pour identifier les plus intéressantes. C'est justement cette identification qui pose problème lors du *docking*, à savoir si la génération et l'évaluation des complexes sont plausibles. C'est pourquoi, il est préférable parfois de former manuellement les complexes, lorsque l'utilisateur a une idée du placement des molécules entre-elles. Dans les expériences de *docking*, on

distingue les algorithmes de *docking* rigide (où les molécules sont figées) et ceux de *docking* flexible. Dans le premier cas, il y a une simplification des degrés de libertés comme les deux molécules sont rigides. L'un des premiers programme à utiliser ce genre d'algorithme est le programme DOCK (Kuntz et *al.*, 1982) : DOCK a été développé pour trouver des molécules à haut complémentarité de forme avec le site de liaison. Pour cela, le site est transformé en image négative qui consiste en une collection de sphères chevauchantes dont les rayons touchent la surface moléculaire en seulement deux points. Les ligands sont ensuite confrontés à ces sphères en minimisant les interactions stériques entre le ligand et le récepteur (cf. Figure 2-21)

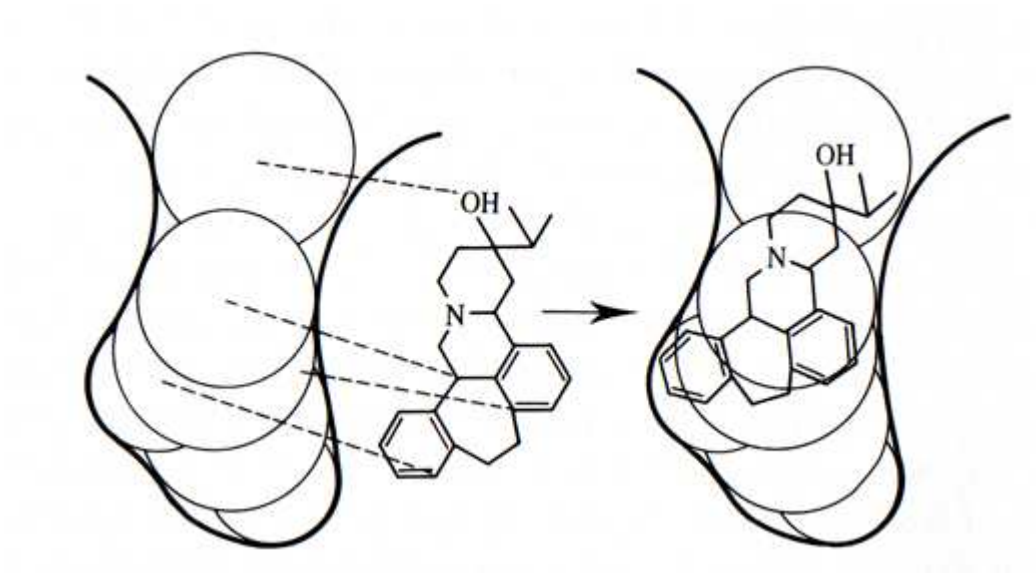


Figure 2-21 L'algorithme DOCK (Kuntz et *al.*, 1982). Les atomes sont positionnés au centre des sphères et la molécule est positionnée à l'intérieur du site de liaison. Figure extraite de la page 556 de Molecular Modelling d'AR. Leach.

Dans le cas des *docking* flexibles, tous les degrés de liberté conformationnels sont pris en compte, même si la plupart du temps ils ne sont appliqués qu'aux ligands, le récepteur restant rigide. Plusieurs méthodes sont utilisées pour effectuer cette recherche conformationnelle. On peut citer par exemple la méthode de Monte Carlo (MC), souvent associé à du recuit simulé (Goodsell et Olson, 1990). A chaque itération de la procédure MC, la conformation interne du ligand est changée (par rotation autour d'une liaison) ou la molécule entière est aléatoirement translatée ou retournée. L'énergie du ligand à l'intérieur du site est alors calculé par mécaniques moléculaires, et le mouvement est alors accepté ou rejeté suivant le critère de Metropolis. Les algorithmes génétiques peuvent aussi être appliqués pour le *docking* moléculaire (Judson et *al.*, 1994 ; Jones et *al.*, 1995 ; Oshiro et *al.*, 1995). Chaque chromosome code non seulement pour les conformations internes du ligand, mais aussi pour sont orientation à l'intérieur du site. L'orientation et les conformations internes varient donc pendant

que la population évolue. Le score de chaque structure complexée agit comme une fonction de fitness utilisée par chaque individu à chaque itération. Enfin, la distance géométrique peut également être utilisée pour ce genre de *docking* moléculaire.

2.7 Conclusion

Il a été vu au cours de ce chapitre quelles sont les étapes nécessaires pour étudier une protéine *in silico* par modélisation moléculaire. Afin de pouvoir étudier la fonction biologique d'une protéine, on s'intéresse d'abord à sa structure : c'est le concept de la biologie structurale. Celle-ci repose sur l'obtention d'un modèle des plus fiables possibles afin d'être utilisé en tant que support d'étude. Lorsqu'on s'intéresse aux mécanismes de reconnaissance d'une protéine, l'obtention du meilleur modèle outre la structure cristallographique, est le modèle par modélisation comparative. Ce modèle repose sur un alignement des plus rigoureux et des plus fiables possibles. Dans le cas des P450s, la construction de modèle ne pose pas de problème au sein d'une même famille. Ainsi, reconstruire un CYP2C8 à partir d'un CYP2C9 ne pose pas en soi une difficulté majeure. En revanche, lorsque l'on ne dispose pas de structure d'une même famille, le problème d'alignement est flagrant : les P450s étant très divergentes en séquences, il s'avère extrêmement difficile d'obtenir un alignement de séquence correct avec les outils actuellement disponibles. C'est pour cette raison que nous avons développé au sein du laboratoire, une méthode nouvelle qui prend en compte la conservation de la structure pour produire cet alignement : c'est le sujet du prochain chapitre.

Une fois le support obtenu, la compréhension des mécanismes de reconnaissance des CYPs nécessitera l'analyse de résultats de *docking* de ligands dans le site actif qui peut être à l'état natif ou légèrement modifié afin de repérer les résidus essentiels dans la reconnaissance du substrat.

Deuxième Partie

Nouvelles méthodes

CHAPITRE 3

Modélisation comparative adaptée pour les superfamilles

« Il faut construire sa vie et son bonheur avec ses propres outils et non avec ceux du voisin »
Daniel Desbiens (1954– ?? ap JC)

3.1 Introduction à la méthode

À mon arrivée au laboratoire, on m'a d'abord confié un sujet portant sur l'étude et la compréhension des mécanismes moléculaires impliqués dans les phénomènes de reconnaissance de multiples substrats par les cytochromes P450 de la famille des 3A (CYP3A). Je devais pour cela avoir recours à une approche bioinformatique qui combine des techniques variées et complémentaires, telles que la modélisation comparative, l'arrimage moléculaire (*docking*) et la dynamique. En début de thèse, aucune structure cristallographique de la famille CYP3A n'était disponible dans la PDB, mais plusieurs structures de P450s de micro-organisme ($P450_{cam}$, $P450_{terp}$, $P450_{eryF}$, $P450_{nor}$ et CYP 51) et de mammifère (en général privés de leur hélice d'ancrage à la membrane : CYP 2C5, CYP 2C8, CYP 2C9, CYP 2B4) étaient dipsonibles. L'idée était d'utiliser les structures connues de CYPs pour construire un modèle de CYP3A, mais la faible identité en séquence des structures disponibles fut le premier frein à la proposition d'un modèle de CYP3A fiable.

Toutefois, une propriété importante et caractéristique des P450s pouvait être mise à profit : leur repliement unique, de la bactérie au mammifère. Ainsi, une stratégie en 3 étapes inspirée de celle de P.Jean (Jean et *al.*, 1997) pour reconstruire un $P450_{eryF}$ a été adoptée. A la différence des techniques classiques de modélisation comparative qui s'appuient sur un alignement purement séquentiel, la méthodologie consiste ici à aligner par **éléments structuraux** plusieurs *templates*, pour définir des zones fixes, en termes de structure. Ces zones correspondent à des morceaux d'hélices ou de feuillets, mais ne sont pas limitées aux SSE uniquement : avec la technique de recherche utilisée, des morceaux de boucles sont également pris en compte. Ces zones, appelées Blocs Structuraux Communs (CSBs), sont utilisées pour être alignées avec la séquence du P450 à modéliser. Les zones variables seront quant à elles reconstruites sans information structurale *a priori*. Ainsi, cette méthode s'affranchit de l'utilisation d'une structure unique, évitant d'obtenir un modèle trop proche des *templates* initiaux, et permet également de fournir un alignement en séquence plus fiable.

Cette méthodologie en trois étapes fera l'objet de ce chapitre : dans un premier temps je montre comment obtenir les CSBs. Puis, je décrirai brièvement le principe d'alignement des CSBs sur la séquence cible à modéliser. Enfin, je présenterai les procédures suivies pour l'obtention d'un modèle à partir de l'alignement obtenu.

3.2 Recherche des éléments structuraux conservés

3.2.1 Structures de départ : un choix crucial pour la reconstruction

Comme pour toutes les méthodologies de reconstruction par modélisation comparative, la première question à laquelle un modélisateur se confronte, est celle du choix des structures de départ, celles qui serviront de « référence », les *templates*. La méthodologie de reconstruction par CSBs des P450s a été appliquée plusieurs fois dans l'histoire du laboratoire. Chaque fois, un jeu de structures de départ différent a été choisi selon la disponibilité et l'évolution des structures cristallographiques de P450s dans la PDB au moment de l'étude. Ainsi, N. Loiseau est le premier au laboratoire à avoir expérimenté cette méthode : il s'est servi d'un jeu initial de *templates* de six structures, les seules disponibles à l'époque (Loiseau, 2002) : $P450_{nor}$, $P450_{eryF}$, $P450_{terp}$, $P450_{BM3}$, $P450_{cam}$ et CYP 2C5 qui était alors la première structure de mammifère publiée dans la PDB. En 2003, M. Cottevieille expérimente à nouveau cette méthodologie pour construire un CYP 3A7 à partir d'un nouveau nouveau jeu de *templates*, assez similaire à celui de N. Loiseau mais plus restreint et dont les structures étaient mieux résolues : $P450_{eryF}$, $P450_{BM3}$, CYP 51 et CYP 2C5. À mon tour, en 2004, j'eus à exploiter cette méthode pour reconstruire un CYP 3A4, mais en bénéficiant d'une PDB significativement enrichie en structures de P450 membranaires de mammifères. Paradoxalement, l'identité de séquence avec la cible 3A4 ne fut pas réellement meilleure que pour les P450s bactériens. Le choix des structures de référence pour la reconstruction du CYP3A4 a alors été guidé par plusieurs critères, tels que les données biochimiques de chaque enzyme utilisé, la nature et la taille des substrats reconnus, mais également la résolution atomique de chaque structure. Par ailleurs, ces structures ont été choisies de façon à être de longueur adéquate pour l'alignement sur la séquence du CYP3A4. Ainsi, en accord avec ces critères, j'ai sélectionné six structures cristallines : trois bactériennes et trois de mammifère. L'idée de reconstruire un P450 humain tel que le CYP3A4 aurait pu me conduire à restreindre mon choix des *templates* aux seules structures cristallines de mammifère. Cependant les seules structures disponibles à cette époque provenaient exclusivement de la famille des 2C, très homologues entre elles. Le résultat aurait inévitablement débouché sur un modèle trop proche de celui des CYP2C, à l'image d'une stratégie « mono-*template* », alors qu'il n'y a pas de raison fonctionnelle pour privilégier le repliement du 2C, en dehors de l'origine (microsomale). Il est à noter qu'une autre structure de mammifère était disponible à cette époque : celle du CYP 2B4 à conformation ouverte. En raison de sa conformation différente des autres P450s, ce cristal n'a pas pu servir dans le jeu de *templates*.

Un bon compromis fut donc d'ajouter à ce jeu de structures « mammifères », trois structures bactériennes avec des homologies fonctionnelles. Ce choix respecte ainsi l'universalité du repliement P450 et l'absence d'identité marquée pour l'un ou l'autre des *templates*. L'ensemble des structures choisies figure dans Tableau 3.1.

Tableau 3.1 Structures cristallines utilisées pour reconstruire le CYP3A4. Les identités sont données par un Blastp de la séquence du CYP3A4 sur la PDB.

Jeux de templates	***P450***	***Code PDB***	***Organisme***	***Référence***	***Identité avec CYP3A4***	***Longueur de la séquence***	
						Réelle	***cristallisée***
A	cam	3CPP	*Pseudomonas putidas*	Raag et Poulos, 1990	20%	414	414
A	terp	1CPT	*Pseudomonas sp.*	Boddupalli et *al.*, 1992	24%	428	412
A	nor	1ROM	*Fusarium oxysporum*	Park et *al.*, 1997	20%	403	403
A B	2C5	1DT6	*Oryctolagus cuniculus*	Williams et all., 2000	26%	487	473
B	51	1E9X	*Mycobacterium tuberculosis*	Podust et all., 2001	26%	522	455
A B C	eryF	1OXA	*Saccharopolyspora erythraea*	Cupp-Vickery et Poulos, 1995	26%	406	403
A B C	BM3	2HPD	*Bacillus megaterium*	Ravichandran et *al.*, 1993	28%	1049	471
C	154C1	1GWI	*Streptomyces coelicolor*	Podust et *al.*, 2003	23%	407	411
C	2C5	1NR6	*Oryctolagus cuniculus*	Wester et *al.*, 2003	26%	487	473
C	2C8	1PQ2	*Homo sapiens*	Schoch et *al.*, 2004	26%	490	476
C	2C9	1OG5	*Homo sapiens*	Williams et *al.*, 2003	28%	490	475

Les structures qui ont servi de *templates* à chaque utilisateur sont représentées par les symboles :
- A pour ceux de N. Loiseau,
- B pour ceux de M. Cottevieille,
- C pour ceux de T.A Nguyen**.**

Les trois jeux comportent trois *templates* en commun :
- CYP 2C5, qui est un P450 microsomal (lapin),
- $P450_{eryF}$ qui reconnaît des substrats similaires à ceux du CYP 3A4
- $P450_{BM3}$ qui est lié à la CPR et donc proche des P450s microsomaux

À l'issue des travaux combinés de N. Loiseau, M. Cottevieille et de moi-même, trois jeux de blocs ont finalement été définis à partir de trois groupes de structures plus ou moins différents. Les structures utilisées présentent un taux moyen d'identité avec le CYP3A4 compris entre 20 et 28%, et ce taux est encore plus faible dans la moitié N-terminale ce qui rend la modélisation par homologie peu fiable si elle se base uniquement sur l'alignement des séquences primaires. Nous avons utilisé les informations de l'alignement tridimensionnel exclusivement au départ pour améliorer l'alignement des séquences (en particulier dans la région N-terminale) et imposer à notre modèle des contraintes spatiales.

3.2.2 GOK, l'outil d'alignement 3D

Pour réaliser l'alignement structural, et pallier la faible identité de séquence entre les différents P450s dans la partie très variable, nous avons choisi d'employer une méthode utilisant une comparaison structurale multiple à l'aide du logiciel GOK développé par Joël Pothier à l'Atelier de BioInformatique de l'Université Paris VI (A.B.I.). Cette méthode, décrite dans la section 2.4.5.1, utilise les coordonnées internes de la chaîne peptidique, à savoir un jeu d'angles dièdres, puisque la protéine peut être décrite comme une trajectoire dans le plan de Ramachandran. Il travaille avec le couple (α, τ) (Levitt, 1976), mais dans le cas de notre étude, seules les coordonnées angulaires α sont utilisées, l'angle τ étant à peu près constant (cf. section 2.2.4). Une sous-structure commune (CSB pour « Conserved Structural Blocs ») est donc définie comme un segment de trajectoire commun à toutes les structures dans une marge d'erreur définie par l'utilisateur. Le quadrillage de ce plan (la maille) et l'utilisation d'une marge de tolérance (la marge) permettent de définir un critère de similarité multiple et précis (cf. Figure 3-1).

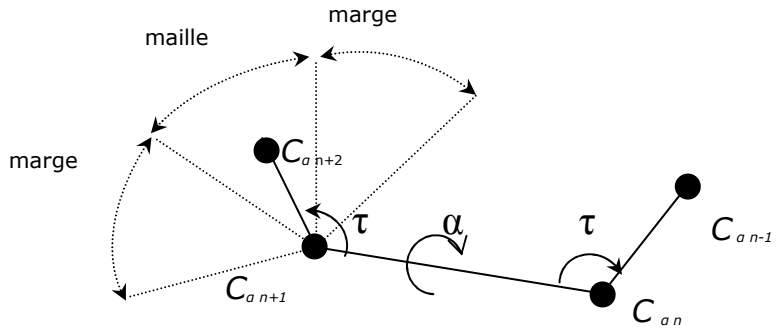


Figure 3-1 Comparaison structurale dans GOK : la maille (définie en unités d'angle) et la tolérance (marge en unités de maille) pour l'angle dièdre α définissent une zone de similarité. Les sous-structures ayant un angle α compris dans cette zone, sont considérées comme similaires. Dans la présente figure, chaque marge correspond à une maille.

GOK propose **deux mesures rms** multiple pour vérifier que les correspondances entre les coordonnées internes et les structures 3D déterminées sont bien en accord. Le premier, noté **mp-rms** (pour « *mean of all pairwise rms deviation* »), est une moyenne arithmétique des déviations calculées par paires de structures. Si on note S_1, S_2, …, S_m les m structures, chacune composée de n carbones α,

et $s_{ij} = (x_{ij}, y_{ij}, z_{ij})$ les coordonnées du $C_\alpha j$ dans la structure i ($i = 1, \ldots, m$; $j = 1, \ldots, n$), alors la déviation minimale rms Δ_{kl} entre les structures k et l est définie comme suit :

$$\Delta_{kl}^2 = \min_{Rt}\left\{\sum_{j=1}^{n}\left|Rs_{kj} + t - s_{lj}\right|^2\right\} \Big/ n$$

où le minimum est pris sur toutes les rotations propres R et translations t (Kabsch, 1976 ; Sippl et Stegbuchner, 1991). Le mp-rms est alors défini de la façon suivante :

$$mp - rms = \sum_{1 \geq k > l \geq m} \Delta_{kl} \Big/ \frac{m(m-1)}{2}$$

Le deuxième critère, noté **s-rms** (pour « *star-rms* »), est obtenu par le calcul de la moyenne arithmétique de toutes les déviations rms entre chaque *template* et une structure « moyenne » S_0 (c-à-d recalculée sur les angles moyens) calculé sur le bloc. Celle-ci est définie comme suit :

$$s - rms = \sum_{k=1}^{m} \Delta_{0k} \Big/ m$$

GOK est donc une méthode qui a l'avantage de tester toutes les solutions possibles sans favoriser l'une des structures en tant que « pivot » et permet, en théorie, la comparaison simultanée d'un grand nombre de structures.

Au moyen de cette méthode, des éléments structuraux communs sont alors identifiables sur l'ensemble des structures choisies en *template*. Ces éléments qu'on appellera également Blocs Structuraux Communs (CSB) ou encore, blocs, correspondent souvent à des éléments de structures secondaires. Pour désigner ces éléments de structure secondaire chez les P450s nous utiliserons celle proposée par Poulos (Poulos et al, 1985) et revue par Hasemann (Hasemann et *al.*, 1995). Toutefois, un bloc **ne correspond pas à un simple élément de structure secondaire**, il peut contenir un fragment d'un tel élément, par exemple une partie de brin ou d'hélice, ou encore un groupe de SSEs contigus.

Dans l'étude des P450s, et selon les structures présentes dans chaque jeu, un nombre différent de CSBs a été déterminé. Ce nombre de blocs tourne autour d'une vingtaine et la longueur totale des blocs couvre à peu près 60 à 70% de toute la protéine. Ils serviront alors d'ossature pour la reconstruction de la protéine cible, un peu à la manière des techniques d'enfilage (ou *threading*). Ils sont présentés en détail dans le chapitre 5.

3.2.3 Vers un jeu de blocs général…

Avec trois ensembles différents de structures cristallines –comprenant toutefois des structures communes– nous sommes parvenus par trois fois à obtenir à peu près le même jeu de blocs. Chaque jeu de blocs a été aligné indépendamment sur les mêmes P450s cibles en vue de reconstruire leurs modèles. La plupart du temps (comme il sera montré dans la partie résultats) les modèles obtenus sont différents. Cette différence pouvant provenir de l'alignement des blocs sur la séquence cible, je me suis intéressé de près aux positionnements des blocs pour les trois jeux de blocs. Il s'est alors avéré que selon le jeu de blocs utilisés, le positionnement de certains blocs ne concordait pas d'un jeu à l'autre. Comme nous n'avions pas utilisé les mêmes structures pour chacun des jeux, l'idée m'est venue de faire un jeu de blocs général qui comprendrait les onze différentes structures utilisées au travers des trois jeux de *templates*. GOK recherche des trajectoires 3D communes et fournit les résultats de blocs sous forme de courts sous-alignements (cf. Figure 3-2) comme le font la plupart des logiciels d'alignement structural. Pour élaborer un jeu global de blocs à partir des trois jeux disponibles, il suffisait donc *a priori* de fusionner entre eux les sous-alignements de chaque bloc. Cette tâche qui semblait aisée au premier abord s'est révélée moins évidente que prévue : certains sous-alignements de blocs ne concordaient pas avec leurs homologues provenant d'un set de *templates* différent. Par exemple, deux séquences d'un même bloc pour un jeu de *templates* donné n'étaient pas trouvées alignées de la même façon dans un autre jeu, pour ce même bloc.

La plupart des désaccords trouvent leur origine dans un décalage en séquence en début ou fin de bloc. Ces décalages de quelques résidus (d'1 résidu à 6) sont observés chaque fois au niveau d'un bloc comprenant une partie d'hélice α. Ce décalage des sous-alignements est explicable : lors d'une identification de bloc par le logiciel GOK, l'utilisateur doit choisir un sous-alignement structural qui représente une trajectoire commune à tous les *templates*, parmi plusieurs sous-alignements possibles. Cette sélection s'effectue graphiquement *via* l'interfaçage du logiciel de visualisation Midas (UCSF, Université de Californie), couplé à GOK. Les nombreux sous-alignements structuraux proposés sont souvent chevauchant entre eux, et se distinguent alors les uns par rapports aux autres par un décalage au niveau de l'initiation (et par conséquent de la terminaison) du sous-alignement, d'une structure ou plus, parmi toutes les structures présentes dans le jeu. La Figure 3-2 présente justement un exemple de ce cas de figure où un décalage d'1 résidu et un décalage de 6 résidus sont observés pour le bloc 4 (voir numérotation des blocs, chapitre 5.2). Dans cet exemple, les structures des $P450_{cam}$ (pdb 1oxa), $P450_{BM3}$ (pdb 2hpd) et CYP 2C5 (pdb 1dt6) sont présentes dans les 3 jeux de *templates*. Visuellement, et en ne prenant en considération que les alignements correctement superposés (cf. légende de la

Figure 3-2), les sous-alignements structuraux de chaque région correspondent bien à une trajectoire similaire sous GOK : il s'agit ici d'un morceau d'hélice G. En revanche, en s'intéressant de plus près aux séquences des sous-alignements de ce bloc, un décalage est observé entre les deux jeux de blocs (entre A et B) de 6 résidus pour la séquence du $P450_{BM3}$ et de 1 résidu pour la séquence du CYP 2C5 par rapport à la séquence du $P450_{eryF}$. Pour trancher entre ces deux résultats, il a donc fallu comparer ce bloc avec celui de mon jeu de blocs, en prenant également en considération les autres structures de ce même bloc non montrées ici : dans les trois jeux, lorsque les sous-alignements d'un même bloc sont en accord dans au moins deux jeux, c'est ce sous-alignement qui est choisi. Dans certains cas, l'utilisation d'un autre logiciel d'alignement structural a été nécessaire (Matras).

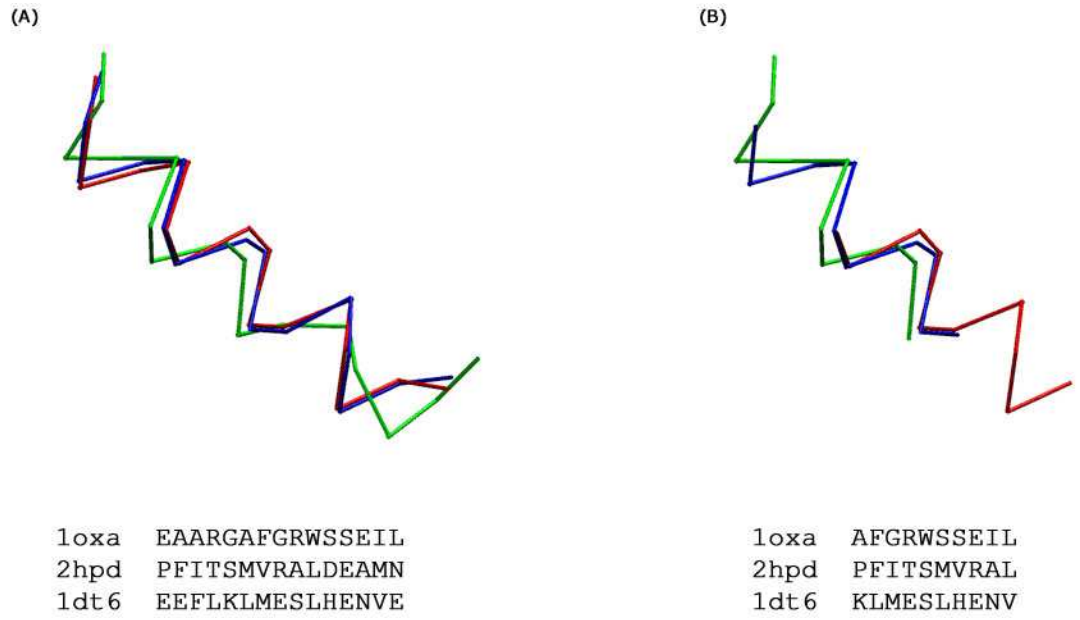


Figure 3-2 Décalage observée pour le bloc 4 entre le jeu de N. Loiseau (A) et celui de M. Cottevieille (B). L'hélice G correspondant au bloc 4 dans les structures 1oxa (en vert) 2hpd (en rouge) et 1dt6 (en bleu) est représenté en trace sous VMD. Dans (B), la représentation est volontairement calée sur le même alignement structural que dans (A) pour mettre en évidence le décalage de sous-alignement. En pratique, sous Midas (couplé à GOK), les 3 fragments d'hélices seraient bien sûr superposés pour former un bloc. L'alignement en séquence correspondant est montré. Dans cet exemple, un décalage de 6 résidus est observé sur 2hpd et un décalage de 1 résidu sur 1dt6. Comme une hélice a une même trajectoire dans l'espace, un décalage de 1 résidu ou de 3 n'est pas observable sous Midas. À noter que le bloc trouvé par M. Cottevieille est plus court que celui trouvé par NL.

Ce décalage de 1 à 6 résidus au niveau des blocs n'est pas surprenant en soi, c'est une limite de principe de la détection par les angles α : la trajectoire d'une hélice est décalable d'un ½ tour (1 à 2 résidus décalés dans les sous-alignements), ou même d'un tour (3 à 4 résidus). Sachant qu'il faut 3,6 résidus pour faire un tour d'hélice, un décalage de 3 à 4 résidus en alignement séquetiel est compréhensible pour un décalage d'un pas d'hélice. Dans certains cas, lors de la recherche, GOK propose comme sous-alignement une région d'une hélice pour cinq structures, et une région d'une

autre hélice pour la sixième structure. Comme les trajectoires des hélices se ressemblent dans l'espace, GOK a pu considérer des morceaux d'hélices différents comme une même trajectoire commune.

Au final, l'évaluation de la « bonne » trajectoire commune n'est pas facilitée par la lisibilité du programme de visualisation et il n'est pas rare, notamment au niveau des hélices α d'observer des décalages d'un résidu ou plusieurs résidus. En prenant connaissance de cette analyse, j'ai pu finalement constituer un jeu de blocs communs aux trois jeux de *templates* (cf. Figure 5-4 et Figure 5-5 dans la partie des résultats) qui **couvre plus de 60% de la séquence des P450s alignés**. À noter que les deux structures du CYP 2C5 (1nr6 et 1dt6) ont été conservées dans le jeu de bloc général car les séquences sont légèrement différentes en raison de mutations introduites pour la cristallisation.

3.2.4 Et pourquoi pas un jeu de blocs universel des 29 *templates* de P450s ?

3.2.4.1 Les limites de GOK

Le logiciel GOK a été développé courant 1997, prévu pour fonctionner sur des Silicon Graphics (SGI) et couplé à Midas, un logiciel de visualisation de structures qui avait l'avantage de pouvoir intégrer des modules « maison ». Du fait de son ancienneté et du support sur lequel il tourne, il s'est montré (et demeure) limité dans son fonctionnement. Ainsi, déterminer 24 CSBs sur un jeu de 6 *templates* nécessitait plus de deux semaines de travail et de calcul sur une SGI Octane R10000. Ce programme a été ensuite exporté et recompilé sur Linux, mais n'offrait plus la possibilité de visualisation. Sans l'outil de visualisation, les problèmes liés au décalage sont difficilement repérables. C'est une des raisons pour lesquelles je n'ai pas refait de nouveau jeu de blocs sous GOK avec un jeu de *templates* enrichi et à jour de la PDB. En effet, le rythme de publications de nouvelles structures de P450s dans la PDB s'est intensifié durant mes trois années de thèse. Par ailleurs, du fait que les structures récentes balayent un spectre plus large de P450s, les incorporer dans notre jeu de *templates* rend la reconstruction d'un P450 inconnu plus aisée : en augmentant le nombre de *templates* avec de nouvelles structures, on augmente également la chance d'avoir dans le jeu un *template* proche du P450 inconnu à reconstruire. Dans l'optique de valider ma méthodologie de reconstruction de P450s à faible taux d'identité, il était préférable de ne conserver que les 11 *templates* dont je disposais à l'époque où j'ai commencé ce travail.

3.2.4.2 GAKUSA : un remplaçant pour GOK ?

L'information des blocs que fourniraient les 29 différentes formes de P450s cristallisées (29 structures non redondantes à la date du 12 avril 2007) n'est pourtant pas négligeable : le passage à 29 *templates* permettait de vérifier la solidité du jeu de blocs inital à 11 *templates*. De façon intuitive, on

peut penser que sur un ensemble plus grand de structures de référence, des blocs en nombre et de longueur moins importante seraient identifiés. Comme l'alignement sous GOK des 29 *templates* n'était pas envisageable (lié à la limitation de calcul des machines), il fallait trouver un autre logiciel au fonctionnement similaire à celui de GOK. GAKUSA est un autre logiciel développé à l'ABI (Atelier de Bioinformatique, Institue Curie) basé sur une approche comparable à GOK dans sa représentation des structures. En revanche, (voir section 2.4.5.2), son algorithme est totalement différent. Contrairement à GOK, GAKUSA ne dispose pas d'interface graphique, mais est compensé par son automatisation : il identifie en un temps rapide tous les CSBs des structures. L'utilisateur peut de plus imposer une longueur minimale de recherche de blocs : GAKUSA détermine alors itérativement chaque position de CSBs trouvés avec un score associé. À chaque itération, les positions trouvées sont cachées pour l'itération suivante, forçant le logiciel à identifier de nouveaux blocs (comme dans GOK). Lors des premiers essais sur un jeu de structures restreintes de P450s, GAKUSA paraissait non seulement beaucoup plus rapide que GOK, mais permettait également de traiter un plus grand nombre de structures simultanément. J'ai donc utilisé ce logiciel afin de comparer les blocs obtenus sur l'ensemble des 29 *templates* de P450 par rapport aux jeux de blocs dont je disposais déjà.

Pour constituer le jeu des 29 *templates* (cf. Tableau 3.2), seules les structures les mieux résolues ont été retenues en cas de redondance (substrats, mutants, etc.). Dans le Tableau 3.2, on peut remarquer que les structures bactériennes sont plus représentées, mais le jeu dispose quand même de 10 templates de P450s microsomaux de plus en plus diversifés alors qu'ils étaient dominés par des structure de P450s de la famille 2C en 2004.

3.3 Positionnement des CSBs sur la séquence cible

Lorsque les blocs structuraux sont identifiés, la seconde étape de la méthode consiste à rechercher la meilleure correspondance entre les sous-alignements en séquences de ces blocs (définis structuralement et indépendamment de la nature des acides aminés), avec la séquence cible qui elle-même peut être alignée avec d'autres séquences à fort taux de similarité. Pour cela, tous les CSBs sont transformées en « profils » (ou PSSM) puis sont alignés sur la séquence de la P450 de structure inconnue ou contre un pré-alignement de séquences de plusieurs P450s homologues. Ce pré-alignement permet de tenir compte de la dégénérescence de séquence au sein d'une sous-famille de P450, ce qui peut aider à compenser la faible identité en séquence entre les différents P450s. Cette étape de recherche de positionnement des CSBs est par ailleurs très importante : la qualité

d'alignement des blocs sur la séquence cible est cruciale pour la reconstruction du modèle, l'alignement de la partie N-terminale pouvant conduire à de nombreuses incertitudes et ambiguïtés.

Tableau 3.2 Liste des structures ayant servi pour constituer le set de *templates* général.

CYP	*PDBID*	*Rés (A)*	*Date de publication*	*Espèce*	Nombre de structures disponibles
$P450_{cam}$	1re9	1.45	16/11/2003	*Pseudomas putida*	57
$P450_{BM3}$	2ij2	1.20	07/11/2006	*Bascillus megaterium*	21
$P450_{nor}$	2jfb	1.00	20/12/2001	*Fusarium oxysporum*	12
$P450_{eryF}$	1z8o	1.70	12/04/2005	*Saccharopolyspora erythrea*	9
$P450_{Oxyb}$	1lfk	1.70	11/12/2002	*Amicolatopsis orientalis*	3
$P450_{Oxyc}$	1ued	1.89	09/12/2003	*Amicolatopsis orientalis*	1
$P450_{epok}$	1q5d	1.93	28/10/2003	*Polyangium cellulosum*	2
CYP 119	1io7	1.50	28/02/2001	*Solfobus solfataricus*	5
CYP 121	1n40	1.06	04/02/2003	*Mycobacterium tuberculosis*	4
CYP 152A1	1izo	2.09	18/03/2003	*Bacillus subtillis*	1
CYP 154A1	1odo	1.85	02/01/2004	*Streptomyces coelicolor*	1
CYP 154C1	1gwi	1.92	29/01/2003	*Streptomyces coelicolor A3(2)*	1
CYP 158A2	1s1f	1.50	11/01/2005	*Streptomyces coelicolor A3(2)*	5
CYP 175A1	1n97	1.79	25/02/2003	*Thermus thermophilis*	2
CYP 51	1x8v	1.55	23/11/2004	*Mycobacterium uberculosis*	6
$P450_{st}$	1ue8	3.00	13/07/2004	*Sulfobus tokodaii*	1
$P450_{PIKC}$	2cd8	2.85	20/02/2007	*Streptomyces venezuelae*	5
$P450_{terp}$	1cpt	2.29	31/01/1994	*Pseudomas sp.*	1
CYP 199A2	2fr7	2.00	16/01/2007	*Rhodopseudomonas palustris*	1
CYP 2A6	2fdu	1.85	28/11/2006	*Homo sapiens*	6
CYP 2B4	1suo	1.89	20/07/2004	*Oryctolagus cuniculus*	3
CYP 2C5	1nr6	2.03	12/08/2003	*Oryctolagus cuniculus*	3
CYP 2C8	1pq2	2.70	13/01/2004	*Homo sapiens*	1
CYP 2C9	1r9o	2.00	15/06/2004	*Homo sapiens*	3
CYP 2D6	2f9q	3.00	20/12/2005	*Homo sapiens*	1
CYP 3A4	1tqn	2.04	27/07/2004	*Homo sapiens*	6
CYP 8A1	2iag	2.15	10/10/2006	*Homo sapiens*	1
CYP 2R1	2ojd	2.70	30/01/2007	*Homo sapiens*	1
CYP 1A2	2hi4	1.95	20/02/2007	*Homo sapiens*	1

3.3.1 SmartConsAlign, le premier logiciel d'alignement de CSBs sur la séquence cible

N. Loiseau et M. Cottevieille, ont tous deux utilisé un même logiciel pour aligner leur jeu de blocs : *SmartConsAlign*, développé à l'ABI (Jean et *al.*, 1997). Ce logiciel été écrit pour trouver la meilleure correspondance possible entre le profil de chaque CSBs et la séquence à aligner. Pour ce faire, à chaque bloc structural était associé une matrice dite consensus (ou profil/PSSM) donnant la répartition des différents acides aminés pour chaque position dans ce bloc (cf. Figure 3-3). Dans

SmartConsAlign, la dimension de cette matrice est de 20 lignes, et d'un nombre de colonnes égale à la longueur de la séquence correspondante. Chaque ligne correspond à un symbole de l'alphabet utilisé, généralement à un type d'acide aminé. D'autre part, les coefficients de la matrice sont pondérés par un facteur tenant compte des similarités entre acides aminés : en pratique la matrice finalement utilisée est le produit de la matrice consensus par la matrice de similarité BLOSUM 62. Un score de similarité est alors défini pour chaque position de la matrice le long de la séquence de la protéine à modéliser. La recherche du meilleur score par déplacement de la matrice consensus considérée sur l'alignement préétabli de séquences, permet donc de déterminer la portion la plus similaire. Il est à noter que les matrices consensus sont construites sans insertions ni délétions, mais la position correspondante sur la séquence peut en comporter.

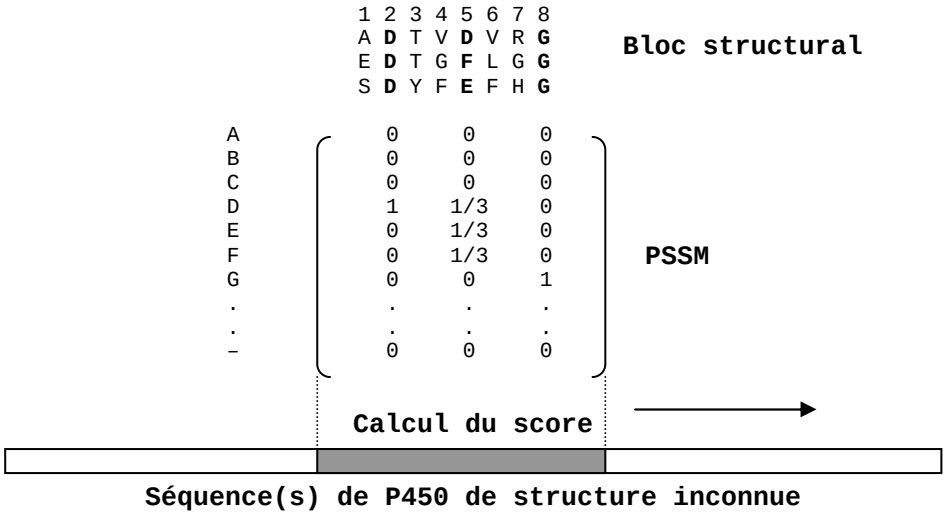


Figure 3-3 Construction et schéma d'utilisation d'une matrice consensus dans *SmartConsAlign*

Pour chaque bloc, *SmartConsAlign* propose différentes solutions classées suivant le score et calcule également pour chaque solution la probabilité (*p-value*) que le score obtenu soit égal au score obtenu avec une séquence quelconque (Goldstein et Waterman, 1994). Cette probabilité permettait d'estimer la fiabilité d'une solution : plus la valeur donnée était proche de 1, plus la fiabilité était faible. Ainsi, le positionnement final des blocs sur la séquence cible résulte d'une prise en compte à la fois des scores d'alignement, mais aussi des positions relatives des CSBs en cas d'ambiguïté.

La principale faiblesse de ce programme vient justement de ces ambiguïtés : chaque bloc est traité indépendamment des autres, sa position peut alors chevaucher celle d'un autre bloc, voire être inversée par rapport à l'ordre initial des blocs structuraux identifiés dans les *templates*. Ces erreurs peuvent être corrigées en utilisant des informations biochimiques ou issues de la littérature.

3.3.2 Caliseq, vers une extension multi bloc de SmartConsAlign

En réponse aux problèmes d'ambiguïté du positionnement des blocs, il fallait mettre en place un nouveau logiciel permettant le traitement simultané de tous les blocs, d'une part qui tienne compte de l'ordre logique des blocs structuraux, d'autre art qui empêche le chevauchement des blocs. L'idée était de reprendre le principe de « matrice consensus » pour représenter chaque bloc, de les traiter en un seul passage et surtout de permettre à ces blocs de « glisser » librement sur la séquence cible jusqu'à leur positionnement définitif. La prise en compte de tous les blocs en un passage ainsi que ce glissement de blocs sur la séquence cible permettrait d'éviter le chevauchement des blocs entre eux et surtout prendrait en compte l'ordre des blocs tels qu'ils ont été identifiés sur les structures de références. Selon ces critères, j'ai donc été amené à développer une nouvelle extension pour le logiciel *SmartConsAlign*, un module qui a été baptisé ***Caliseq*** pour Consensus Alignement of Sequences. Les caractéristiques de cet outil seront détaillées dans le chapitre 4.

3.3.3 Du positionnement des blocs vers l'obtention de l'alignement final

En sortie de *Caliseq*, on obtient un alignement du jeu de blocs sur une séquence (ou sur un alignement de séquence). Cette forme actuelle d'alignement n'est pas immédiatement exploitable par les logiciels de reconstruction 3D par homologie : il faut supprimer les séquences alignées avec la séquence de P450 à reconstruire lorsqu'elles sont présentes, et surtout compléter les parties entre les blocs par les séquences respectives. En effet, les logiciels de reconstruction requièrent la séquence complète des structures de référence. Selon la méthode utilisée pour la reconstruction du modèle, le « remplissage » de ces parties « inter-blocs » a plus ou moins son importance.

3.3.3.1 Les différentes stratégies utilisées

Dans la méthodologie initiale utilisée au laboratoire, il faut construire un jeu de contraintes de distances et d'angles de type RMN dans chaque bloc, représentant une « moyenne » des informations structurales. Ainsi, pour chaque région structuralement conservée, les données structurales de tous les *templates* sont intégrées simultanément, tandis que les parties variables (inter-blocs) sont reconstruites sans contraintes en même temps que les parties structuralement conservées. De ce fait, l'alignement des parties hors blocs des séquences des *templates* sur la séquence cible est sans importance. L'utilisateur pouvait aligner ces régions à sa guise sachant que l'information des résidus au niveau inter-blocs ne serait pas prise en compte lors de la reconstruction.

Il n'en est pas de même lorsqu'on opte pour une autre stratégie utilisant des outils bioinformatiques automatiques de reconstruction par homologie. En effet, la précédente stratégie est très efficace mais nécessite des manipulations manuelles pour un bon nombre d'étapes. Dans un souci de simplification, d'automatisation et de production à grande échelle de modèles, il a été convenu de changer de stratégie, à savoir utiliser des logiciels disponibles de reconstruction par homologie. Cette stratégie devait néanmoins incorporer la « philosophie » des reconstructions par blocs avec priorité absolue des calculs de contraintes dans les blocs. Le logiciel Modeller pouvait répondre au cahier des charges : un fichier d'alignement ainsi qu'un fichier de directives étaient pris en entrée et la construction d'un (ou de plusieurs) modèle pouvait être effectué en tenant compte des contraintes spatiales déterminées par l'alignement.

Avec Modeller, deux approches ont pu être exploitées ici : soit en respectant l'esprit de la première stratégie, utiliser seulement l'automatisation et la rapidité de reconstruction de Modeller, soit en tirant profit des avantages de Modeller pour reconstruire les régions inter-blocs grâce à l'utilisation de sa banque de repliement pour imposer des contraintes spatiales dans les zones inter-blocs. Dans le premier cas, il suffit de ne mettre aucune séquence de *templates* en correspondance avec les régions inter-blocs tel que cela est montré en Figure 3-4. La seule contrainte dans le second cas est la nécessité de fournir un alignement aussi précis que possible des régions inter-blocs.

```
template  aaaaaaaaaaaaaaaaaaaaaaa-----bbbbbbbbbccccccccccccccccccccccccccc
target    ddddddddddddddddddddddddeeeee--------fffffffffffffffffffffffffff
```

Figure 3-4 Méthodes pour neutraliser le calcul de contraintes dans les régions inter-blocs sous Modeller : les régions hors blocs du template ne sont pas alignées avec la séquence. Dans l'exemple, aucune contrainte spatiale dérivée de la séquence *template* ne sera utilisée pour construire la région 'eeeee' de la cible, et la région 'bbbbbbbb' du *template* n'est pas prise en compte dans le calcul.

3.3.3.2 L'alignement inter-bloc, mesure de sécurité et nouvelle difficulté ?

Durant ma thèse, j'ai donc opté pour l'utilisation de Modeller pour la reconstruction à grande échelle des modèles. Les deux approches ont été utilisées. Dans le cas de la seconde approche (utilisation de la banque de repliement de Modeller pour les régions inter-blocs), il était important de bien aligner les parties entre les blocs. La majeure difficulté de cette opération venait du fait que ces régions étaient par nature structuralement variables (non sélectionnées par GOK) et aussi, souvent de tailles très différentes. Par exemple, les P450s microsomaux possèdent de nombreuses boucles non présentes chez leurs homologues bactériens. Il y a donc deux difficultés : aligner des zones de

longueur très hétérogènes entre les *templates*, puis aligner ces régions sur la séquence cible. Plusieurs stratégies ont été essayées pour réaliser cet alignement inter-bloc.

Alignement visuel. La première stratégie est celle de l'approche manuelle. Dans un premier temps, j'ai effectué manuellement tous les alignements inter-blocs des séquences des P450s de référence (cf. Figure 5-5 page 194). Pour cela, j'ai essayé de maximiser les correspondances d'acides aminés en fonction de leur identité ou de leurs propriétés (polarité, hydrophobicité). Cette stratégie est manipulateur-dépendant.

Utilisation de l'information des SSE. Il est possible d'améliorer les résultats d'un alignement en séquence en tenant compte des structures secondaires des *templates*. Dans l'alignement des régions inter-blocs, j'ai utilisé à la fois les informations de SSE décrites dans les fichiers PDB de chaque structure de référence, mais j'ai également eu recours à des logiciels comme le serveur MATRAS (cf. section 2.4.3.3) pour l'alignement des structures secondaires entre elles. La principale difficulté rencontrée ici vient de fait que les régions inter-blocs correspondent la plupart du temps à des régions non structurées. Or et les alignements de structures secondaires proposés sont en désaccord avec les alignements de GOK : n'étant déjà pas en accord sur les régions considérés comme structuralement conservée par GOK, l'alignement inter-bloc est difficilement réalisable.

Utilisation de Clustalw adapté. Finalement, une dernière stratégie a été expérimentée en ayant recours à Clustalw dans un mode de fonctionnement un peu détourné. En effet, il est possible de fournir à Clustalw sa propre matrice de similarité afin d'adapter l'alignement produit en fonction des correspondances indiquées dans la matrice. Pour forcer le calage des blocs (et donc des débuts et fin de zones inter-blocs), j'ai introduit un nouveau caractère dans la matrice (cf. Figure 3-5) pénalisé fortement lorsqu'il se retrouve aligné face à un autre résidu et favorisé fortement lorsqu'il est aligné face à lui-même.

```
#  Matrix made by matblas from blosum62.iij
#  X matches with itself with a high score
#  * column uses minimum score
#  BLOSUM Clustered Scoring Matrix in 1/2 Bit Units
#  Blocks Database = /data/blocks_5.0/blocks.dat
#  Cluster Percentage: >= 62
#  Entropy =   0.6979, Expected =  -0.5209
 A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V  B  Z  X  *
 4 -1 -2 -2  0 -1 -1  0 -2 -1 -1 -1 -1 -2 -1  1  0 -3 -2  0 -2 -1 -9 -4
-1  5  0 -2 -3  1  0 -2  0 -3 -2  2 -1 -3 -2 -1 -1 -3 -2 -3 -1  0 -9 -4
-2  0  6  1 -3  0  0  0  1 -3 -3  0 -2 -3 -2  1  0 -4 -2 -3  3  0 -9 -4
-2 -2  1  6 -3  0  2 -1 -1 -3 -4 -1 -3 -3 -1  0 -1 -4 -3 -3  4  1 -9 -4
 0 -3 -3 -3  9 -3 -4 -3 -3 -1 -1 -3 -1 -2 -3 -1 -1 -2 -2 -1 -3 -3 -9 -4
-1  1  0  0 -3  5  2 -2  0 -3 -2  1  0 -3 -1  0 -1 -2 -1 -2  0  3 -9 -4
-1  0  0  2 -4  2  5 -2  0 -3 -3  1 -2 -3 -1  0 -1 -3 -2 -2  1  4 -9 -4
 0 -2  0 -1 -3 -2 -2  6 -2 -4 -4 -2 -3 -3 -2  0 -2 -2 -3 -3 -1 -2 -9 -4
-2  0  1 -1 -3  0  0 -2  8 -3 -3 -1 -2 -1 -2 -1 -2 -2  2 -3  0  0 -9 -4
-1 -3 -3 -3 -1 -3 -3 -4 -3  4  2 -3  1  0 -3 -2 -1 -3 -1  3 -3 -3 -9 -4
-1 -2 -3 -4 -1 -2 -3 -4 -3  2  4 -2  2  0 -3 -2 -1 -2 -1  1 -4 -3 -9 -4
-1  2  0 -1 -3  1  1 -2 -1 -3 -2  5 -1 -3 -1  0 -1 -3 -2 -2  0  1 -9 -4
-1 -1 -2 -3 -1  0 -2 -3 -2  1  2 -1  5  0 -2 -1 -1 -1 -1  1 -3 -1 -9 -4
-2 -3 -3 -3 -2 -3 -3 -3 -1  0  0 -3  0  6 -4 -2 -2  1  3 -1 -3 -3 -9 -4
-1 -2 -2 -1 -3 -1 -1 -2 -2 -3 -3 -1 -2 -4  7 -1 -1 -4 -3 -2 -2 -1 -9 -4
 1 -1  1  0 -1  0  0  0 -1 -2 -2  0 -1 -2 -1  4  1 -3 -2 -2  0  0 -9 -4
 0 -1  0 -1 -1 -1 -1 -2 -2 -1 -1 -1 -1 -2 -1  1  5 -2 -2  0 -1 -1 -9 -4
-3 -3 -4 -4 -2 -2 -3 -2 -2 -3 -2 -3 -1  1 -4 -3 -2 11  2 -3 -4 -3 -9 -4
-2 -2 -2 -3 -2 -1 -2 -3  2 -1 -1 -2 -1  3 -3 -2 -2  2  7 -1 -3 -2 -9 -4
 0 -3 -3 -3 -1 -2 -2 -3 -3  3  1 -2  1 -1 -2 -2  0 -3 -1  4 -3 -2 -9 -4
-2 -1  3  4 -3  0  1 -1  0 -3 -4  0 -3 -3 -2  0 -1 -4 -3 -3  4  1 -9 -4
-1  0  0  1 -3  3  4 -2  0 -3 -3  1 -1 -3 -1  0 -1 -3 -2 -2  1  4 -9 -4
-9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 999 -9
-4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -9  1
```

Figure 3-5 Matrice BLOSUM 62 modifiée : le caractère X est fortement pénalisé (-9) lorsqu'il se retrouve en face d'un autre résidu et fortement favorisé (+999) lorsqu'il se retrouve en face de lui-même.

En utilisant ce caractère pour remplacer les résidus à l'intérieur des blocs, le résultat escompté est un alignement où Clustalw force les blocs à s'aligner entre eux, et traite normalement les résidus inter-blocs. Cette méthode n'est pas toujours adaptée : certains blocs ne pourront pas être utilisés dans Clustalw, principalement ceux qui ne sont pas trouvés dans toutes les structures. En effet, n'ayant pas de correspondance dans les autres séquences des *templates*, ils viendraient à fausser le résultat par un décalage des blocs. Par exemple, dans l'alignement de la Figure 5-5 page 194, le premier bloc (CSB0) n'est présent que sur 5 des 11 structures. En remplaçant tous les résidus des blocs par le symbole X, certaines séquences du CSB0 sont venus s'aligner sur les séquences du CSB1* provoquant ainsi quelques décalages dans l'alignement comme le montre la Figure 3-6. Par ailleurs, même en supprimant des jeux les blocs non trouvés dans toutes les structures, Clustalw n'a pas été en mesure d'aligner correctement les blocs entre eux à l'aide de ce caractère : des décalages sont toujours observés, et dans certains cas, les décalages se prolongent dans tout l'alignement.

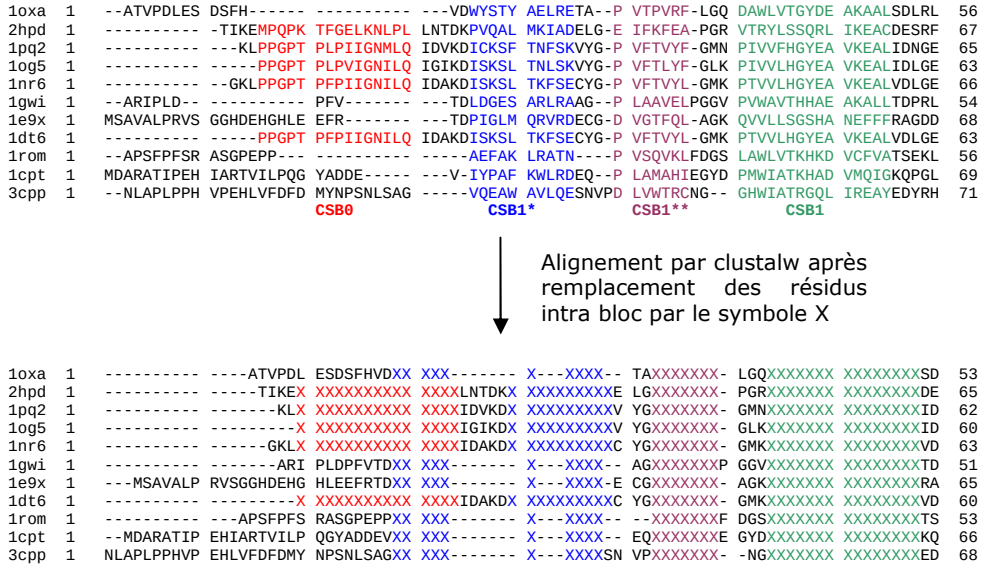


Figure 3-6 Problème lié aux blocs non présents sur tous les *templates*: lorsque les résidus à l'intérieur de ces blocs sont remplacés par le symbole X, Clustal a du mal à les aligner correctement. Un code de couleur a été utilisé pour représenter les résidus de chaque bloc. Dans les *templates* où le bloc 0 est absent, des résidus X du bloc suivant (CSB 1*) se sont détachés pour s'aligner sur CSB0 des autres *templates*.

La stratégie initiale par calcul de jeu de contraintes uniquement dans les blocs de type RMN n'accordait pas de valeur à l'alignement inter-bloc : les régions variables inter-blocs étaient censées ne contenir aucune information structurale et donc être reconstruite sans information *a priori*. En reproduisant cette stratégie par la première approche de Modeller, les régions inter-blocs sont généralement mal reconstruites, Modeller étant incapable de construire ces régions de façon *ab initio*. C'est pourquoi, c'est finalement la seconde approche (par utilisation de la banque de repliements de Modeller) qui a été utilisée pour produire les modèles de P450s. Dans cette approche, on pouvait se contenter de produire un alignement grossier inter-bloc pour satisfaire Modeller, et optimiser les régions inter-blocs une fois le modèle obtenu. Il était cependant préférable de produire le meilleur alignement inter-bloc possible : aucune stratégie hormis l'alignement manuel n'a pu donner de résultats convaincants. Il a été vu en effet que ni l'utilisation d'autres informations (SSE par exemple) ni l'alignement Clustalw « adapté » pour les zones inter-blocs n'ont pu être exploitables. Au final, pour construire les modèles de P450s obtenus sous Modeller, **l'alignement des zones inter-blocs a été réalisé manuellement**.

3.4 Construction des modèles de Cytochrome P450

Pour la troisème étape, après l'alignement des *templates* sur la séquence cible, deux procédures ont été mises au point pour construire les modèles : (i) la procédure initiale mettant en jeu des logiciels et des concepts issus de la détermination structurale par RMN, utilisée par N. Loiseau et M. Cottevieille, et (ii) la procédure que j'ai appliquée, plus automatisée et plus commune pour la construction de modèles par homologie.

3.4.1 Méthode initiale basée sur la RMN

L'alignement structural obtenu par GOK distingue deux types de régions : (i) les blocs structuralement conservés et (ii) les segments d'acides aminés qui les séparent et qui ne fournissent aucune information structurale. Les premiers servent de support à la reconstruction du P450 désiré, les seconds sont quant à eux sans importance. Dans les régions alignées (blocs), les atomes conservés pour le calcul des contraintes dépendent des chaînes latérales des résidus alignés. S'il n'y a pas de glycine à une position donnée de l'alignement, les C_β sont conservés. Les atomes de rang γ peuvent également être conservés s'ils existent dans tout le bloc (cf. Figure 3-7). La conservation des atomes de rang δ et au-delà est plus rare, sauf dans le cas d'identité absolue dans le bloc.

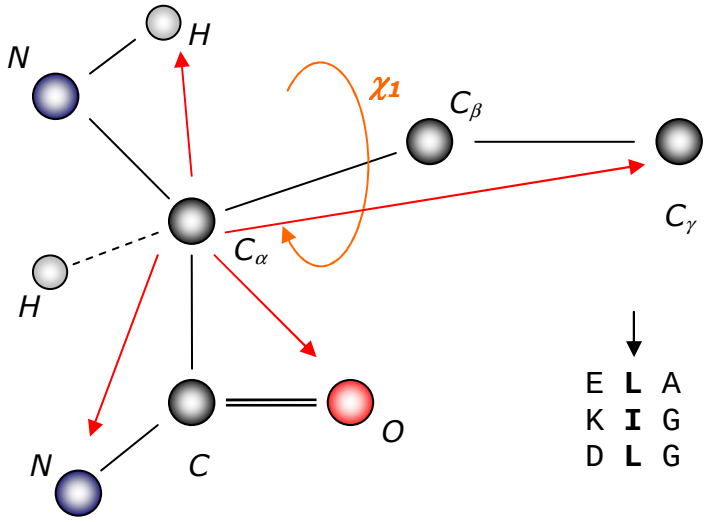


Figure 3-7 Exemple de conservation des atomes pour le calcul des contraintes de distances. Si C_β et C_γ sont conservés dans le bloc, ces atomes du résidu cible sont imposés et le χ_1 fixé. En revanche, si seuls les C_β sont conservés et que le résidu cible comporte un C_γ (ou plus), il faut alors choisir un rotamère χ_1. Sur cette figure sont également représentées (en flèches rouges) les distances calculées pour les contraintes (distances au plus proche voisin, ne prenant pas en compte les atomes covalemment liés).

L'une des premières étapes de cette méthode consiste en la sélection des atomes conservés dans les chaînes latérales des résidus du bloc, puis à sélectionner les rotamères quand il manque de l'information. Les angles de torsion des chaînes latérales, χ_1, χ_2, etc… des acides aminés d'une protéine adopte généralement trois orientations préférentielles : +60°, –60° ou +180°. Au sein de la protéine, la distribution de ces angles de torsion dépend de l'élément de structure secondaire auquel participe l'acide aminé et donc de la conformation locale de son squelette. L'équipe de Karplus (Dunbrack et Karplus, 1993) a établi une librairie de rotamères à partir de l'étude de 132 structures, qui permettait de corréler la position d'un couple (ϕ,ψ) dans le plan de Ramachandran avec un rotamère particulier (cf. Figure 3-8). Ces tables permettent de sélectionner le rotamère le plus probable pour chacun des acides aminés des blocs conservés à partir des angles (φ,ψ) « expérimentaux ».

	χ_2=−60 ± 60	χ_2=0 ± 30	χ_2=−60 ± 30
χ_1=−60 ± 60	3,5 (1)	8,8 (2)	4,7 (3)
χ_1=180 ± 60	6,8 (4)	9,8 (5)	11,8 (6)
χ_1= 60 ± 60	29,6 (7)	18,7 (8)	5,3 (9)

Pourcentage de chacune des combinaison (χ_1,χ_2) sur la population d'asparagine de 132 protéines sélectionnées à la PDB. Chaque rotamère est désigné par un numéro entre parenthèse.

ψ/φ	-180	-160	-140	-120	-100	-80	-60	-40	-20	0
180	3	3	3		2	8				
160		5	9	9	8	9	8			
140			5	5		5	5			
120		5	5	5	5	5				
100			5	8	4					
80		5	5		7					
60				9						
40		2	2	8	2					
20			2	8	8					
0				7	8	8				
-20					9	8	7			
-40				8		9	8			
-60										
-80										
-100										
-120										
-140										
-160		2								
-180	3	3	2	2						

Figure 3-8 Exemple de distribution des rotamères les plus probables en fonction des angles (φ,ψ) : cas de l'asparagine (d'après Dunbrack et Karplus, 1993). Le tableau de gauche montre une liste des 9 rotamères possible de l'asparagine et le pourcentage de précesne de ces rotamères dans la PDB. Le tableau de droite correspond à une table de correspondance des rotamères les plus probables en fonction de la configuration locale (φ,ψ).

L'étape suivante consiste en la construction des fichiers de contraintes d'angles et de distances. Pour chacun des résidus des blocs conservés, les angles dièdres φ, ψ, χ_1, χ_2, sont dérivés de l'alignement ou inférés à partir des rotamères probables pour constituer le **jeu de contraintes angulaires**. Chacun des angles φ, ψ est défini par une valeur et un écart type résultant de la moyenne arithmétique des données expérimentales de l'ensemble des structures *templates* tandis que les valeurs et écart types des angles χ_1, χ_2, sont repris de celles fournies par Dunbrack et Karplus. Le **fichier de contraintes de distances** nécessaire à la construction des modèles est lui établi à partir du calcul de toutes les distances entre un atome donné et tous les atomes voisins conservés (non liés covalemment) pour chaque structure *templates* dans un rayon maximal donné (ou cut off) (cf. Figure 3-7). Pour ne pas générer un fichier de contraintes trop important ou redondant, on choisit de : (i) sélectionner toutes les distances inter atomiques inférieures à 8 Å, (ii) de compléter par des distances conservées

hème-résidus pour ancrer l'hème, (iii) les distances entre chaînes latérales (par exemple C_β,– C_β de 2 résidus voisins dans l'espace ou dans la séquence) sont intégrées pour conserver l'agencement des différentes blocs entre eux, en particulier entre blocs non contigus. Les distances ainsi conservées permettent de décrire l'environnement local de chacun des atomes. Chaque contrainte est définie par deux valeurs (moyenne ± écart type).

Lors de la reconstruction, les 30 premiers résidus en N-terminal – notamment lors de reconstruction de P450s membranaires à partir de *templates* bactériens – sont généralement supprimés : n'ayant pas d'équivalent structural, cette partie modélisée sans contrainte pouvait perturber l'ensemble du modèle. La reconstruction à partir des contraintes géométriques et la minimisation sont conduites sous le logiciel DYANA (Güntert et al, 1997) initialement destiné au calcul des structures de protéines ou d'acides nucléiques à partir de contraintes conformationnelles provenant d'expériences RMN[7].

Dans cette étape d'optimisation géométrique, les boucles (ou régions inter-blocs) sont reconstruites sans contraintes. Il peut en résulter des configurations anormales dans l'espace des φ, ψ au sens des régions autorisées de l'espace de Ramachandran. Pour rendre ces structures plus acceptables au sens de Ramachandran, un programme établissant des contraintes angulaires est appliqué pour le squelette des acides aminés dans les régions hors blocs. Il consistait à déplacer un point (φ,ψ) d'une zone interdite vers une zone autorisée la plus proche. Les modèles ainsi travaillés étaient ensuite soumis à un recuit simulé sous contrainte par le logiciel X-Plor (Brunger, 1992), permettant de relâcher les modèles obtenus parfois trop compacts après minimisation par DYANA.

Cette première méthode globale procurait en son temps des modèles de très bonne faction aussi bien d'un point vue géométrique qu'énergétique. Elle nécessitait hélas une succession importante d'étapes, longues et lourdes en calcul avec manipulations de nombreux fichiers au format différent. Aucune automatisation n'était envisageable dans ce cadre, c'est pourquoi j'ai choisi une autre approche pour la construction des modèles, approche réalisée en grande partie par un seul logiciel : Modeller (cf. section 2.5.1.2).

[7] DYANA a succédé au logiciel DIANA (Distance geometry Alogrithm for NMR Applications) et n'est plus supporté actuellement. La dernière version en date est DYANA v1.5

3.4.2 Modeller, un logiciel de modélisation comparative prêt à l'emploi

3.4.2.1 Utilisation conventionnelle de Modeller

Largement répendu dans la communauté scientifique, le logiciel Modeller offre l'avantage de calculer lui-même les contraintes d'angles et de distances de façon automatisée : il suffit pour cela de lui fournir un ensemble de structures de références, un fichier d'alignement de ces structures et enfin un fichier de commandes contenant une liste de directives. Il dispose également de procédures de recuit simulé pour la reconstruction de boucles et de minimisation pour produire des modèles stables énergétiquement. Bref, c'est un package complet pour la construction de modèles par homologie. La principale contrainte pour nous est le respect de la « philosophie » des blocs, à savoir le respect de la topologie et les contraintes structurales au niveau des blocs. Le problème lors d'utilisation de logiciels de type « boîte noire », c'est qu'il n'est pas toujours évident d'avoir un contrôle sur ce genre de chose. Nous avons d'abord pensé utiliser le programme pour construire les modèles et exploiter les fichiers de contraintes générés par Modeller par retour à la stratégie d'optimisation en recuit simulé sous contraintes (section 3.4.1). Cela n'a pas été possible dans la mesure où le fichier de contraintes récupéré n'était pas compréhensible et donc non exploitable. Par ailleurs, ce logiciel suit une procédure rigoureuse qu'il faut respecter au mieux si l'on désire profiter de sa pleine potentialité pour construire un modèle correct. Par exemple, une des contraintes imposée par le programme est de fournir un alignement des *templates* le plus précis possible sur la séquence cible. Ceci suppose un alignement à la fois des blocs sur la séquence cible, mais aussi des régions inter-blocs, jugées non prioritaires pour la reconstruction dans la philosophie des blocs. Par conséquent, Modeller traite donc à niveau égal aussi bien les régions reconnues par GOK comme structuralement conservées et les régions hors blocs. En effet, Modeller détermine lui-même les régions conservées et les régions dites variables et se charge de reconstruire ces régions variables à l'aide d'informations qu'il puise dans sa banque de repliements.

Une petite remarque peut être formulée pour l'attachement de l'hème lors de la reconstruction du P450. En fait, Modeller n'est pas en mesure de reconnaître de nouveaux résidus (par exemple une cystéine modifiée) ou construire des cofacteurs (comme l'hème par exemple), et se contente de copier les coordonnées de l'hème d'une des structures de référence au modèle. En conséquence, il n'y a pas d'attachement de l'hème à la cystéine proximale du P450. Il devra être effectué *a posteriori* par l'utilisateur.

3.4.2.2 Traitement additionnel lors de reconstruction de boucles

Une première possibilité est de laisser le programme agir avec les paramètres par défaut, mais j'ai pu également expérimenter un autre moyen qui « mimait » au mieux la première méthode pour le traitement des régions inter-blocs, puisque Modeller offre également la possibilité de construire des boucles *de novo*. Il est possible en effet de « forcer » Modeller à ne pas s'appuyer sur les contraintes spatiales fournies par les *templates* au niveau des régions inter-blocs : il suffit de ne pas aligner ces régions comme il a été présenté à la Figure 3-4. Ne disposant pas de contraintes spatiales dérivées des structures de référence, Modeller reconstruit alors ces régions de façon *ab initio*. Une reconstruction *ab initio* donne toujours de longues boucles non structurées au niveau de ces régions privées de contraintes spatiales. Ainsi, dans ce cas, on applique une seconde procédure à Modeller pour « raffiner » ces régions, exactement de la manière utilisée dans la première stratégie issue de la détermination structurale par RMN. Cette procédure peut s'effectuer durant la reconstruction (voir exemple en annexe 4.2) Elle ne nécessite que le fichier de structure du modèle à traiter et un script de commandes similaire au fichier général de commandes, utilisé pour la reconstruction des modèles.

L'utilisateur doit donc informer au programme au moyen de ce script, les régions qu'il considère comme des boucles à reconstruire (voir annexe 4.2). Afin de conserver l'arrangement spatial des blocs les uns par rapport aux autres au cours de ce processus, un jeu de contraintes entre chaque C_α des résidus de fin de bloc avec celui de début du bloc suivant, est également défini dans le fichier de directives. Ces contraintes spatiales sont appliquées au niveau de la distance, des angles de torsions, des angles impropres et dièdres. Une fois toutes ces informations déclarées, le fichier de directives commande Modeller qui va déplacer aléatoirement les coordonnées cartésiennes des C_α de chaque boucle, puis procéder à une optimisation locale par recuit simulé de chaque boucle, et ce de façon itérative. C'est un peu l'équivalent du recuit simulé effectué par le programme Xplor pour reconstruire les boucles dans la première stratégie, mais cette fois-ci, de manière automatisée.

3.5 Évaluation, sélection et raffinement des modèles

Quelles que soient les méthodes utilisées pour la reconstruction, un nombre important de modèles doit être généré. En effet, les méthodes de reconstruction donnent rarement une solution unique puisque tous les modèles générés sont triés par le seul critère de la fonction énergétique. Pour explorer au mieux l'espace des solutions les plus énergétiquement favorables, il faut multiplier le nombre de modèles et sélectionner selon les critères d'évaluation définis par l'utilisateur.

3.5.1 Les différents critères d'évaluation

Les différents outils et critères d'évaluation ont été déjà abordés lors de la section 2.5.4. Selon les méthodes utilisées pour la reconstruction, les critères d'évaluation n'ont pas été les mêmes. Ainsi, pour les premières méthodes basées sur des outils RMN, le critère essentiel était celui de la géométrie au niveau du plan de Ramachandran. Le logiciel de référence pour ce genre d'évaluation est PROCHECK. Il permet entre autres de vérifier si les couples (φ,ψ) sont bien situés dans les zones autorisées du diagramme de Ramachandran, de contrôler de la distorsion géométrique des chaînes latérales, la planéité des cycles aromatiques etc. Les meilleurs modèles PROCHECK étaient alors sélectionnés pour une relaxation, et une seconde sélection se fait ensuite sur le critère de l'énergie totale de chaque modèle après relaxation.

Pour les modèles issus de Modeller, un processus assez similaire a été appliqué. Modeller dispose de son propre score, appelé fonction objective, qu'il calcule pour tous les modèles qu'il génère. Ce score est une combinaison de plusieurs paramètres aussi bien statistiques qu'énergétiques. Plus ce score est bas, meilleure est la structure. Cependant, la meilleure fonction objective n'est pas toujours synonyme de meilleure solution. En effet, la notion de « meilleur modèle » est variable selon les logiciels d'évaluation utilisés. De ce fait, chaque fois que j'ai eu à sélectionner un modèle parmi l'ensemble de ceux générés par Modeller, j'ai dû chercher un compromis entre les différentes méthodes d'évaluation. Avec l'expérience, certains programmes ont été privilégiés : ainsi, je n'ai pas eu à utiliser PROCHECK dans la mesure où tous les modèles sortis par Modeller étaient « propres » au niveau du plan Ramachandran. J'ai en revanche utilisé ANOLEA, PROQ ainsi que Prosa II pour évaluer les modèles (cf. Tableau 2.3 à la page 120). Il n'a pas été possible pour moi d'établir un script pour le choix des modèles par ces outils : les scores étant calculés différemment, un consensus n'est pas réalisable. La recherche d'un meilleur compromis entre les scores calculés par ces différents programmes est encore manuelle.

3.5.2 Affinement avant d'être exploité

Une fois le modèle choisi, vient l'étape d'affinement. Le modèle est dans un premier temps soumis au logiciel SCWRL (http://bioserv.cbs.cnrs.fr/HTML_BIO/frame_scwrl.html) (Canutsecu et *al.*, 2003) qui réattribue les chaînes latérales de façon à minimiser les « clashes » entre chaînes latérales, et entre chaînes latérales et squelette peptidique. SCWRL dispose pour cela d'une bibliothèque de rotamère et propose une liste de valeurs de χ_1-χ_2-χ_3-χ_4 en fonction des valeurs φ et ψ

du squelette peptidique. Lorsque les chaînes latérales sont correctement positionnées, une minimisation est opérée suivie d'une dynamique pour « relâcher » la molécule. C'est d'ailleurs au cours de cette étape que l'on procède à la « fixation » de l'hème sur la cystéine proximale, et ce quel que soit le logiciel utilisé pour la dynamique. Au cours de ma thèse, j'ai été conduit à tester différents champs de force sur les modèles que j'obtenais : GROMOS, CHARMM et AMBER. J'ai d'abord travaillé avec le champ de force GROMOS en raison de la simplicité du programme GROMACS pour le paramétrage et sa robustesse. Dans ses premières versions (V3.1.4) il était très aisé d'attacher l'hème à l'apoprotéine : GROMACS disposait en effet des fichiers nécessaires à la reconnaissance de l'hème et sa fixation à la cystéine la plus proche. Cette procédure de fixation se faisait automatiquement, sans intervention de l'utilisateur. Dans les versions plus récentes, la présence de l'hème dans le site actif, empêche l'encapsulation de la protéine entière dans une boîte d'eau. Comme je n'ai pas réussi à trouver l'origine du problème, je suis passé à une version plus ancienne de GROMACS et plus tard à une nouvelle suite de logiciels. J'ai adopté par la suite le programme NAMD pour réaliser les simulations, car c'est un outil qui offre la possibilité d'utiliser aussi bien le champ de force CHARMM qu'AMBER. Les différentes simulations opérées sur les modèles ont montré qu'il n'y avait pas de différences notables entre ces deux champs de force, au niveau des résultats. Pour la suite des expériences, c'est finalement le champ de force AMBER que j'ai utilisé.

Quels que soient les outils utilisés pour la simulation, les paramétrages des simulations ont été à peu près équivalents. Dans tous les cas, le modèle à relaxer a été placé dans une boîte d'eau périodique –on parle alors de simulations en solvant explicite– dont les bords sont situés à 10 Å du bord des résidus de surface de la protéine. En fait, la taille de la boîte est définie par l'utilisateur, et peut être plus petite. Par sécurité, la marge de 10 Å autour de la protéine a toujours été appliquée, conduisant à des boîtes périodiques parallèlipédiques de l'ordre de 95x85x75 $Å^3$. La forme de la boîte n'est quant à elle pas importante (cubique, octaédrique …), seule sa périodicité l'est. En effet, les conditions périodiques (correspondant à une duplication de la boîte le long des trois axes du référentiel) permettent le maintient du nombre de molécules total, de volume et de pression dans les ensembles NVT ou NPT. Une des conséquences indirecte lors de l'utilisation de ces conditions périodiques est de s'affranchir d'un nombre trop important de molécules dans le système et donc soulager le temps de calcul pour chaque simulation. Ainsi, si une molécule d'eau par exemple sort de la boîte par une face, elle est générée systématiquement aussitôt sur la face opposée, permettant au système de conserver le même nombre d'atomes. Dans un souci de neutralité du système, des contre-ions sont ajoutés au système (Na^+ ou Cl^-). Une fois la solvatation réalisée, le système peut alors être minimisé, équilibré et soumis à une simulation de MD pour des temps variables allant de la picoseconde à la nanoseconde,

qui se traduisent en temps de calcul de quelques jours à quelques semaines. En fin de dynamique, l'énergie du système est évaluée pour vérifier si le modèle obtenu est stable ou non. Après cette ultime étape, je dispose alors d'un support de travail pour analyser les mécanismes de reconnaissance des P450s *in silico*.

3.5.3 Traitement additionnel (et optionnel) des régions inter-blocs sur les modèles obtenus

Lorsque j'ai voulu valider la méthodologie de reconstruction à l'aide de blocs structuraux conservés, j'ai souvent été confronté à des « divergences structurales » au niveau des régions inter-blocs entre le modèle et la structure de référence (exemple 1tqn pour le modèle du CYP 3A4). Quelle que soit l'approche utilisée sous Modeller (avec ou sans alignement inter-bloc), des différences de repliements sont observées entre la structure-test et les modèles générés, localisées surtout au niveau de ces régions qui ne devaient pas porter d'information structurale. La méthode de reconstruction des boucles proposées par le logiciel Modeller n'a pas donné les résultats espérés, comme cela a été déjà évoqué. Dans l'ancienne stratégie basée sur les outils RMN, N. Loiseau et M. Cottevieille essayaient de s'affranchir de ce problème par l'utilisation du recuit simulé sous Xplor. C'était possible car les fichiers de contraines étaient générés explicitement en format Xplor, et le programme pouvait calculer toutes les violations des contraintes spatiales pour chaque modèle généré. Avec Modeller, cette étape n'est pas accessible du fait d'un format obscure des fichiers : la récupération des fichiers de violations des contraintes de Modeller n'était d'aucune aide, étant dans l'incapacité de les exploiter.

Il fallait donc trouver une alternative à cette étape ultime d'affinement sous Xplor. Cela m'a conduit à prendre contact avec K. Zimmermann du MIG (Mathématique, Informatique et Génome) à l'INRA de Jouy-en-Josas. Je cherchais un minimisateur efficace et seul ORAL (Zimmermann, 1991) semblait adapté à l'esprit de la minimisation par blocs : il relaxe uniquement des portions d'une molécule bordées par deux blocs rigides (ou semi-rigides) déterminés par l'utilisateur. Son concept plutôt novateur était cependant limité par la génération de champ de force qu'il intègre : AMBER4. Avec mes simulations conduites sous AMBER6, les topologies n'étaient pas compatibles. Il a donc fallu reconstruire toutes les topologies sous AMBER4. Par ailleurs, ORAL utilise le package d'AMBER, qui est sous licence. Je ne disposais pas de licence sur mes propres machines.

3.6 Conclusion

Au travers de ce chapitre, j'ai présenté la stratégie de reconstitution de P450s inconnus à basse identité de séquences mise au point au laboratoire. Avec les différentes générations de modélisateurs, cette stratégie s'est vue modifiée. J'ai réalisé les principaux changements au cours de ma thèse, en apportant une nouvelle façon d'aligner ainsi qu'une nouvelle manière de construire les modèles, tout en conservant la philosophie initiale : celle d'utiliser les éléments structuraux communs à tous les P450s pour servir d'ossature aux nouveaux modèles. Ainsi, pour chaque étape, j'ai présenté les deux méthodes (l'ancienne et la nouvelle) afin que l'on puisse se rendre compte à la fois de leurs différences, mais également des problèmes auxquels j'ai été confronté pour adapter cette méthodologie aux nouveaux outils mis à disposition. À chaque difficulté rencontrée dans l'adaptation de la méthode, différentes approches pour trouver une solution ont été présentées.

On dit souvent qu'un beau dessin vaut mieux qu'un long discours, aussi, je terminerai ce chapitre par un schéma récapitulatif présentant la méthodologie générale de reconstruction de P450s utilisée au laboratoire sous ses deux aspects : avant et après mon arrivée. La dualité sera alors plus visible pour chaque étape de la reconstruction.

Le prochain chapitre est consacré au logiciel que j'ai développé pour automatiser de façon non ambiguë le positionnement des blocs sur la séquence cible : ***Caliseq***. Ce logiciel prend donc place à l'une des étapes les plus importantes de la méthodologie, à savoir fournir au logiciel de construction de modèle par homologie, l'alignement le plus fiable.

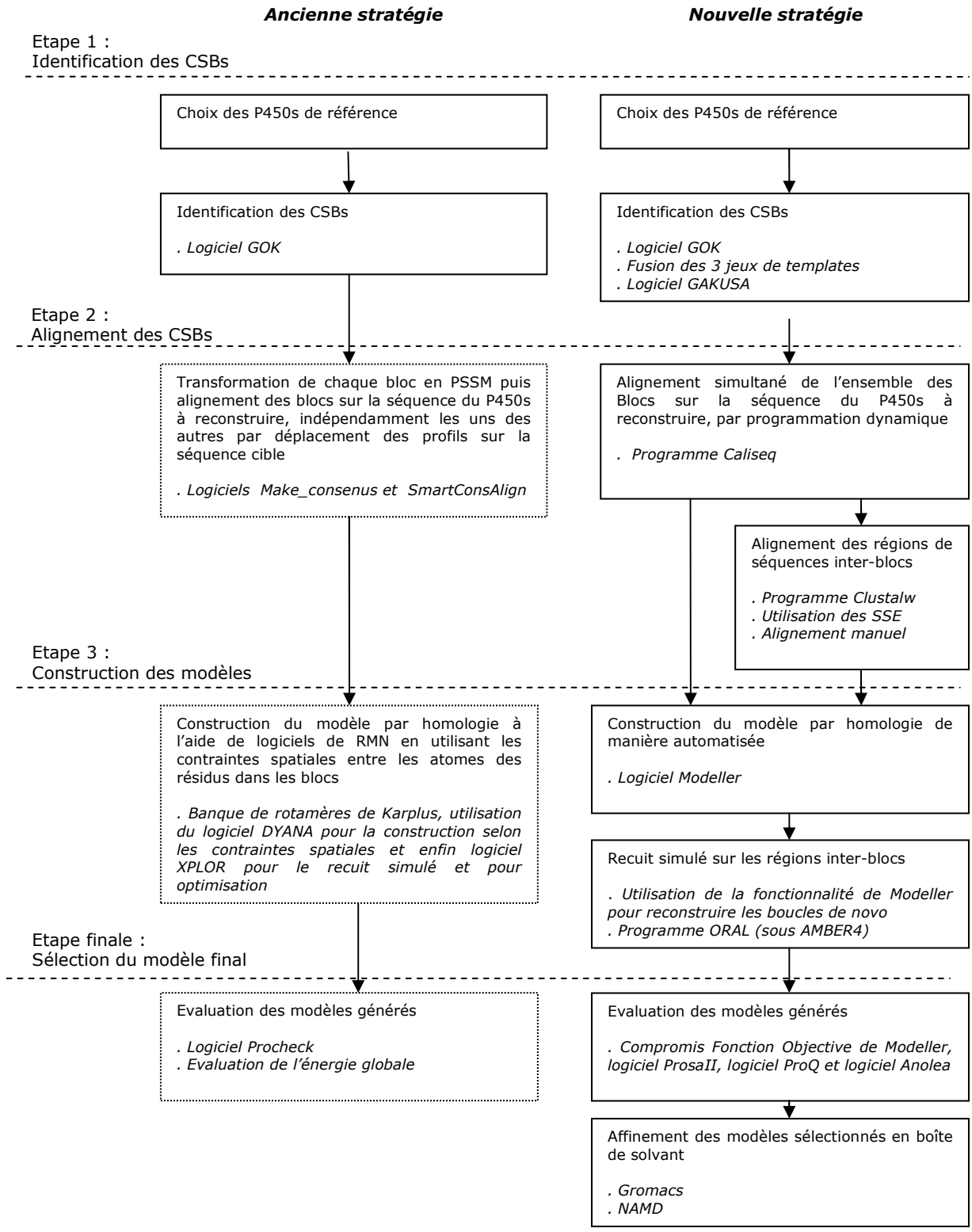


Figure 3-9 Schéma récapitulatif de la méthodologie appliquée à la reconstruction à bas taux d'identité des P450s

CHAPITRE 4

Voyage au centre de *Caliseq*

« Mon oncle Lidenbrock nous met tous à la diète
jusqu'au moment où il aura déchiffré un vieux
grimoire qui est absolument indéchiffrable !»
Jules Verne (1828– 1905 ap JC)

4.1 Présentation de l'outil

Ce chapitre présente le programme qui réalise non seulement l'alignement des blocs sur la séquence cible pour la reconstruction ultérieure du modèle, mais qui offre une autre possibilité: celle de scanner les banques de séquences à la recherche de nouveaux P450s. *Caliseq* est donc un outil à deux facettes, auquel j'ai consacré une grande partie de ma thèse, pour son développement, puis pour son optimisation à la suite de résultats obtenus lors de son utilisation. À l'origine, *Caliseq* a été conçu comme une extension au programme *SmartConsAlign*, utilisé par mes prédécesseurs pour réaliser le positionnement des blocs indépendamment les uns des autres. En réponse à des attentes liées à la méthodologie, comme par exemple l'alignement simultané de tous les blocs (représentés par un ensemble de PSSMs séquentielles), un nouveau cahier des charges précis a été défini. Celui-ci imposa de nouvelles contraintes au programme si bien que son mode de fonctionnement ainsi que son algorithme se sont progressivement éloignés de ceux du programme d'origine, conduisant *Caliseq* à une « entité » à part dans *SmartConsAlign*. Du programme *SmartConsAlign*, *Caliseq* n'en utilise plus que « l'interfaçage » et partage encore quelques fonctions d'ouverture, de lecture et d'écriture de fichiers de séquences. Toutes les parties d'alignement, de calcul de scores et de calcul de seuils sont propres à *Caliseq*. L'interfaçage de *SmartConsAlign* est des plus simples, sans représentation graphique : *SmartConsAlign*, de même que *Caliseq*, ont en effet été tous deux écrits en **langage C** et sont utilisables sur différentes plateformes (Linux, Unix et MacOSX).

Au cours de ce chapitre, je décrirai dans une première partie les grands points du logiciel ainsi que ses paramètres, puis dans une seconde partie, je discuterai des différentes évolutions apportées au programme pour l'adapter au mieux à l'étude des P450s. Le manuel d'utilisation du programme est présenté en Annexe 3, avec une description succincte des fonctions principales qui composent le programme.

4.2 Fonctionnement de Caliseq

4.2.1 De nouvelles entrées pour le programme

Bien que reprenant le principe d'alignement de blocs du programme *SmartConsAlign*, les fichiers d'entrées du nouveau programme devaient être redéfinis pour prendre en compte un ensemble de blocs à la place d'un bloc unique. Tous les blocs (donnés sous forme de sous-alignements de séquences par GOK) sont donnés en entrée dans un même fichier. Le format d'entrée regroupe chaque sous-

alignement de séquences les uns à la suite des autres, séparées par un délimiteur (cf. Annexe 2). Ce fichier d'entrée n'est alors plus adapté à une transformation en matrice consensus effectuée lors d'une étape préalable dans la version originale de *SmartConsAlign* : la construction des profils (*profile* en anglais) liés à ces alignements se fait désormais directement dans le programme *Caliseq*.

Par ailleurs, deux autres fichiers étaient présentés en entrée standard au programme *SmartConsAlign* : i) la séquence de la protéine inconnue à reconstruire (ou son alignement avec d'autres P450s de la même famille) et ii) optionnellement, une matrice de similarité BLOSUM ou PAM pour rajouter du « flou » à la recherche. Dans *Caliseq*, le fichier d'alignement « cible » ne nécessite pas de transformation préalable. La matrice de similarité pour sa part n'est plus utilisée dans la dernière version de *Caliseq*, pour des raisons expliquées plus loin.

Il est à noter que contrairement à *SmartConsAlign*, le programme *Caliseq* autorise la présence des gaps ('–') (insertion ou délétions) dans les profils des blocs : certains blocs n'ayant pas été trouvés sur toutes les structures, leur séquence dans le sous-alignement correspondant sera remplacée par des gaps.

4.2.2 Un algorithme de programmation dynamique adapté

En raison du traitement simultané de tous les blocs, le procédé initial de recherche par déplacement des matrices position par position le long de la séquence cible comme le faisait *SmartConsAlign* (cf. section 3.3.1) n'était plus envisageable. Ce procédé a donc cédé la place dans *Caliseq* à une méthode d'alignement plus adaptée : la programmation dynamique de type NWS (cf. section 2.4.1.3). J'ai dû toutefois adapter cette méthode à nos besoins : d'une part le programme devait prendre en considération des profils au lieu de séquences, et d'autre part il devait autoriser le « glissement » libre –sans pénalité– des blocs sur la séquence cible.

Dans la programmation dynamique classique, il faut, pour aligner deux séquences ***A*** et ***B***, de longueur *M* et *N* respectivement, créer une matrice d'alignement de dimension *M x N* où chaque colonne et chaque ligne correspondent à un acide aminé de la séquence ***A*** et ***B*** respectivement. Pour prendre en compte les profils, ce principe de création d'une matrice *M x N* reste inchangé sauf qu'à la place d'un symbole unique, chaque ligne et chaque colonne correspondent désormais à un ensemble de 20 symboles, un pour chaque type d'acide aminé (voir la Figure 3-3, page 142). En fait, j'ai utilisé plus de 20 symboles dans les matrices consensus puisqu'il faut aussi plusieurs poids de gap : il y a le poids de gap contre acide aminé (asymétrique, selon que le gap est inséré dans la séquence ou dans le

profil, donc deux valeurs différentes) ou encore le poids gap contre gap. Dans *Caliseq*, le poids attribué à une insertion à une position donnée dépend de la fréquence des gaps à cette position du profil du jeu des blocs (des « PSSMs séquentielles ») ainsi que du poids d'ouverture de gap. Cet aspect est abordé plus en détail dans la section 3.4.6 de l'Annexe 3. Concernant le « glissement » libre des profils sur la séquence cible, l'astuce a consisté à ne pas pénaliser les gaps entre les blocs (en fait, sur la première ou la dernière colonne de chaque PSSM). L'utilisation de cette astuce impliquera toutefois certaines contraintes, notamment d' « accrochage » de certains blocs sur la séquence cible comme il sera vu plus tard. L'alignement proprement dit est affiché en sortie de programme au format clustal ou fasta.

4.2.3 Le deuxième rôle de Caliseq

Caliseq a été développé initialement pour aligner un jeu de blocs sur une séquence cible à reconstruire, et ce de façon simultanée et automatique. Par son fonctionnement, *Caliseq* est capable de lire une séquence cible ou un alignement de séquences comprenant la séquence cible, pour le confronter à un ensemble de blocs. Dans la mesure où il peut lire des alignements de séquences, rien ne l'empêche *a priori* de lire individuellement chaque séquence de l'alignement comme des séquences d'une banque[8]. Le but était d'utiliser le jeu de blocs comme une signature spécifique aux P450s contenant l'information de conservation structurale pour identifier de façon fiable les séquences de P450s inconnues (non annotées) ou encore, mettre en évidence des P450s dans des génomes nouvellement séquencés. Cette recherche a alors un intérêt double : non seulement elle permet d'avoir à disposition une banque de séquences de P450s, mais également, en combinant avec la méthodologie de reconstruction par blocs, d'obtenir une banque de structures et/ou sites actifs de P450s reconstruits à partir des séquences identifiées. La banque de modèles 3D pourrait ainsi servir comme base d'études de criblage virtuel à haut débit[9].

L'identification de P450s sur les banques de séquences par *Caliseq* se fait au moyen du score d'alignement des blocs sur la séquence traitée, obtenu en fin de programmation dynamique : ce score servant de critère de sélection, est évalué à un seuil d'acceptation imposé par l'utilisateur. Dans les premières versions de *Caliseq*, ce score ne pouvait pas être utilisé tel quel, et nécessitait une

[8] Il s'agit de banques de séquences enregistrées au format fasta, au sein d'un même fichier. C'est le genre de fichier qui est généralement récupérable à partir des FTP de banques de données telles que Swiss-Prot, TrEmbl, Pfam, etc.

[9] Ces criblages se font généralement avec des méthodes de docking sur site rigide. Les approches virtuelles à haut débit sont utilisées dans le domaine pharmaceutique pour tester des banques de composés contre une cible thérapeutique (ADN ou protéine)

normalisation préalable. Les scores normalisés étaient alors comparés au seuil d'acceptation basé sur un calcul de ratio. Cette première version du seuil n'étant pas suffisamment rigoureuse pour détecter les nouveaux P450s, un nouveau seuil a été ensuite implémenté, basé cette fois-ci sur un calcul de z-score. L'évolution de ce seuil, ainsi que la nécessité de son remplacement sont expliquées dans le paragraphe qui suit.

4.3 Les différentes évolutions de *Caliseq*

Depuis son développement initial, l'outil *Caliseq* a été décliné en plusieurs versions, chacune apportant son lot de nouveautés. Initialement, *Caliseq* était prévu pour remplacer le programme *SmartConsAlign* en réalisant les alignements d'un ensemble de blocs –correspondant à des sous-alignements de séquences de structures de P450s cristallisés– sur une séquence ou un pré-alignement de séquences cibles dans le but de reconstruire un modèle 3D de la protéine cible. *Caliseq* apportait par rapport à *SmartConsAlign* la possibilité de traiter en une seule fois l'ensemble des blocs, en conservant l'information de leur ordre, et en évitant ainsi leur chevauchement possible observé lorsqu'ils étaient traités séparément. Autrement dit, l'information contenue dans l'ensemble de blocs était renforcée par rapport à celle de chaque CSB pris séparément : l'information de repliement commun à tous les P450s n'est plus morcelée par blocs, mais au contraire, présente sur toute la protéine. Partant de ce constat, on a attribué au programme la possibilité de scanner les banques de séquences : celui-ci utilise alors cette information contenue dans les ensembles de blocs pour détecter et identifier des séquences de P450s. L'outil *Caliseq* offrait donc un intérêt nouveau dans le domaine de la génomique, en cherchant de nouvelles séquences de P450s dans les génomes récents. Ce « deuxième mode » de *Caliseq*, celui de recherche sur banque, a donné lieu à la première évolution de l'outil *Caliseq*. De nombreuses modifications dans la structure même du programme ont alors dû être opérées : de nouvelles fonctions ont été conçues, et certains algorithmes, repensés.

Ainsi, par rapport à la version initiale, le calcul de la matrice de substitution a été complètement revu pour mieux répondre au besoin du « scoring » d'alignement sur banque, score qui a servi de critère de sélection des CYPs. Par ailleurs, un système de seuil a été mis en place, testé puis affiné. L'ensemble de tous ces ajustements avait pour objectif principal d'améliorer la sensibilité et la fiabilité de l'outil. Un comparatif des résultats obtenus pour chaque version sera exposé en résultat dans le paragraphe 5.5.

4.3.1 Vers un nouveau système de calcul des « PSSMs séquentielles »

4.3.1.1 Une matrice initiale de similarité empruntée à PAM/BLOSUM

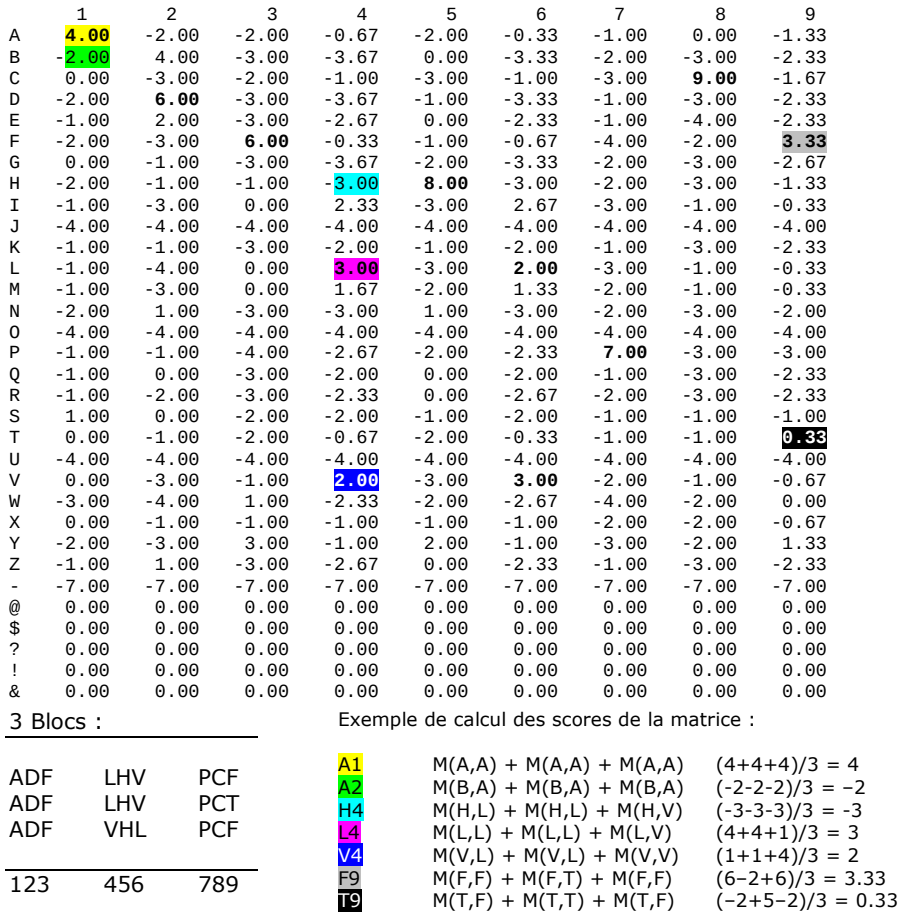
Dans la première version de *Caliseq*, la création du profil pour le jeu des blocs s'appuyait sur une matrice de similarité de type PAM ou BLOSUM qui permettait de pondérer les fréquences des acides aminés dans le profil. Le calcul des poids pour une position donnée du profil (des « PSSMs séquentielles ») correspondait donc dans la version initiale de *Caliseq* à un calcul de fréquences, pondérées par les coefficients de similarités des acides aminés tels qu'ils apparaissent dans la matrice de similarité utilisée. Ainsi le score de substitution *W(j,i)*, pour aligner la position *i* des « PSSMs séquentielles » avec la position *j* dans le pré-alignement de séquence s'obtenait selon la formule suivante :

$$W(j,i) = \sum_{k=1}^{20} \left(\frac{\sum_{n=1}^{N} \left(M(a_k, a_{ni}) \right)}{N} \right) \times f_{pre}(a_{kj})$$

Où a_k est l'un des 20 acides aminés, a_{ni} l'un des acides aminés à la position *i* de la « PSSMs séquentielles », $M(a_k,a_{ni})$ le coefficient de l'acide aminé a_k contre l'acide aminé a_{ni} dans la matrice de similarité (PAM/BLOSUM), *N* le nombre de séquence dans un CSB donné des « PSSMs séquentielles » et $f_{pre}(a_{kj})$ la fréquence de l'acide aminé a_k à la position *j* des séquences pré-alignées. Un exemple de calcul de profil à partir de cette formule est présenté dans la Figure 4-1.

Comme il sera vu dans le chapitre des résultats, la construction de la matrice d'alignement par cette formule a permis au final d'obtenir un alignement de blocs sur la séquence cible similaire à celui de N. Loiseau et M. Cottevieille et ce, sans avoir à replacer manuellement les blocs dont les positions attribuées par *SmartConsAlign* étaient incertaines. L'alignement par programmation dynamique a donc donné directement le résultat optimal, sans avoir recours à l'intervention humaine, prouvant ainsi l'apport de l'outil *Caliseq* par rapport à *SmartConsAlign*.

Néanmoins, l'utilisation des scores de la matrice de similarité était moins bien adaptée pour la recherche sur banque. En effet, lorsqu'on se penche à nouveau sur la matrice de score (Figure 4-1), force est de constater que les scores attribués aux résidus présents dans l'alignement (marqués en gras dans la matrice) ne correspondent pas toujours aux scores les plus élevés. Ceci est lié en fait à l'utilisation d'une matrice de similarité.



	1	2	3	4	5	6	7	8	9
A	**4.00**	-2.00	-2.00	-0.67	-2.00	-0.33	-1.00	0.00	-1.33
B	-2.00	4.00	-3.00	-3.67	0.00	-3.33	-2.00	-3.00	-2.33
C	0.00	-3.00	-2.00	-1.00	-3.00	-1.00	-3.00	**9.00**	-1.67
D	-2.00	**6.00**	-3.00	-3.67	-1.00	-3.33	-1.00	-3.00	-2.33
E	-1.00	2.00	-3.00	-2.67	0.00	-2.33	-1.00	-4.00	-2.33
F	-2.00	-3.00	**6.00**	-0.33	-1.00	-0.67	-4.00	-2.00	**3.33**
G	0.00	-1.00	-3.00	-3.67	-2.00	-3.33	-2.00	-3.00	-2.67
H	-2.00	-1.00	-1.00	-3.00	**8.00**	-3.00	-2.00	-3.00	-1.33
I	-1.00	-3.00	0.00	2.33	-3.00	2.67	-3.00	-1.00	-0.33
J	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00
K	-1.00	-1.00	-3.00	-2.00	-1.00	-2.00	-1.00	-3.00	-2.33
L	-1.00	-4.00	0.00	**3.00**	-3.00	**2.00**	-3.00	-1.00	-0.33
M	-1.00	-3.00	0.00	1.67	-2.00	1.33	-2.00	-1.00	-0.33
N	-2.00	1.00	-3.00	-3.00	1.00	-3.00	-2.00	-3.00	-2.00
O	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00
P	-1.00	-1.00	-4.00	-2.67	-2.00	-2.33	**7.00**	-3.00	-3.00
Q	-1.00	0.00	-3.00	-2.00	0.00	-2.00	-1.00	-3.00	-2.33
R	-1.00	-2.00	-3.00	-2.33	0.00	-2.67	-2.00	-3.00	-2.33
S	1.00	0.00	-2.00	-2.00	-1.00	-2.00	-1.00	-1.00	-1.00
T	0.00	-1.00	-2.00	-0.67	-2.00	-0.33	-1.00	-1.00	**0.33**
U	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00	-4.00
V	0.00	-3.00	-1.00	**2.00**	-3.00	**3.00**	-2.00	-1.00	-0.67
W	-3.00	-4.00	1.00	-2.33	-2.00	-2.67	-4.00	-2.00	0.00
X	0.00	-1.00	-1.00	-1.00	-1.00	-1.00	-2.00	-2.00	-0.67
Y	-2.00	-3.00	3.00	-1.00	2.00	-1.00	-3.00	-2.00	1.33
Z	-1.00	1.00	-3.00	-2.67	0.00	-2.33	-1.00	-3.00	-2.33
-	-7.00	-7.00	-7.00	-7.00	-7.00	-7.00	-7.00	-7.00	-7.00
@	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
?	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
!	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
&	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 4-1 Exemple de « PSSMs séquentielles » générées à partir de l'ancienne formule. La matrice du haut donne les scores obtenus pour le set des trois blocs présentés en bas à gauche. Chaque colonne de la matrice correspond à une position dans l'alignement des blocs, allant de 1 à 9. Des exemples de calcul de scores sont donnés en bas à droite, où M(X,Y) correspond au poids de similarité BLOSUM62 (cf. Figure 2-9, page 85) entre les deux résidus X et Y. Il n'y a que trois séquences dans les blocs, donc N est égale à 3.

C'est le cas de la dernière colonne de l'exemple (Figure 4-1) pour la thréonine dont la valeur est de 0.33. Il est vrai que ce genre de substitution (F↔T) n'est pas usuel, et le remplacement d'une phénylalanine (F) par une tyrosine (Y) (d'un score 1.33) est davantage favorisé. Toutefois, comme il a été remarqué lors de la recherche de CSBs par le logiciel GOK, certains blocs présentent un alignement en séquences non conventionnel, avec de nombreux mésappariements (cf. l'exemple de bloc de la Figure 3-2, page 138). L'existence d'un grand nombre de résidus différents à une même position de l'alignement, entraînerait alors une création de scores assez homogènes à cette position, masquant ainsi l'information qui pouvait y être présente. Ce constat m'a donc conduit à réviser la formule pour obtenir les scores de substitution, mettant plus en valeur les résidus présents dans les alignements des blocs, tout en conservant toutefois une certaine « flexibilité » à l'aide d'une matrice de similarité.

4.3.1.2 Vers la création d'une matrice de similarité propre à l'alignement de la superfamille

L'idée est de calculer les scores de la matrice d'alignement à l'aide d'une formule de log-odd qui s'apparente à celle utilisée pour la matrice BLOSUM ou PAM : en quelque sorte, il s'agit de reconstruire une matrice « BLOSUM like » tirée de nos alignements. Pour mémoire, le score de similarité de la matrice BLOSUM d'un couple $(a_i a_j)$ est obtenu par la formule suivante :

$$S_{ij} = \log_2 \left[\frac{f(a_i a_j)}{2 \times f(a_i) f(a_j)} \right]$$

Où $f(a_i a_j)$ correspond à la fréquence observée du couple $(a_i a_j)$ à une position donnée de l'alignement (voir plus bas) et $f(a_i)$, $f(a_j)$ les fréquences attendues (dues à la composition) – « background frequencies» récupérées des statistiques de SwissProt– des résidus a_i et a_j respectivement. Ce score de type « log-odd » est alors multiplié par deux et arrondi à l'entier le plus proche pour obtenir le score BLOSUM définitif pour le couple $(a_i a_j)$. Le calcul de la fréquence observé du couple $(a_i a_j)$ s'effectue quant à lui de la manière suivante :

$$f(a_i a_j) = 2 \times \frac{Na_i \times Na_j}{L(L-1)}$$

Où Na_i et Na_j sont respectivement le nombre de résidus a_i et a_j dans une colonne d'alignement de longueur L. Il est à noter que cette formule dérive de celle utilisée pour connaître le nombre total de couples dans un ensemble L d'objets.

Le score de substitution $W(j,i)$, pour aligner la position i des « PSSMs séquentielles » avec la position j dans le pré-alignement de séquence est alors calculé selon la nouvelle formule qui suit :

$$W(j,i) = \sum_{k=1}^{20} \log_2 \left[\frac{f_{CSB}(a_{ki})}{f_{bck}(a_k)} \right] \times f_{pre}(a_{kj})$$

Où a_k est l'un des 20 acides aminés, $f_{CSB}(a_{ki})$ est la fréquence de l'acide aminé a_k à la position i dans les « PSSMs séquentielles », calculée avec des pseudo-comptes (voir plus bas), $f_{bck}(a_k)$ est la fréquence « background » de l'acide aminé a_k dans le set de données et $f_{pre}(a_{kj})$ est la fréquence de l'acide aminé a_k à la position j dans les séquences pré-alignées. Pour la position i dans les « PSSMs séquentielles », la fréquence $f_{CSB}(a_{ki})$ pour l'acide aminé a_k est calculée de la manière suivante :

$$f_{CSB}(a_{ki}) = \frac{N_{CSB}(a_{ki}) + e_{ak}}{N + E}$$

$N_{CSB}(a_{ki})$ est le nombre d'acides aminés a_k à la position i dans les « PSSMs séquentielles », N est le nombre de séquences dans la CSB donnée des « PSSMs séquentielles », e_{ak} donne les pseudo-

comptes (Lawrence et *al.*, 1993) pour un acide aminé a_k où $e_{ak}=E*f_{bck}(a_k)$. Les pseudo-comptes, comme expliqué précédemment, ont été mis en place pour pallier l'apparition d'une fréquence nulle pour un acide aminé a_k (qui aurait pour effet d'invalider le score). L'idée est d'ajouter au compte réel, une proportion de pseudo-comptes calculés à partir d'une « probabilité attendue » de l'acide aminé a_k. La valeur de E, à l'image de N, donne le poids entre les pseudo-comptes et les comptes réels (E est habituellement fixé à $\sqrt{N}$). Une matrice de score obtenue par cette nouvelle formule et à partir des trois mêmes blocs que dans l'exemple de la Figure 4-1 est présentée dans la Figure 4-2.

3 Blocs :

ADF	LHV	PCF
ADF	LHV	PCT
ADF	VHL	PCF
123	456	789

	1	2	3	4	5	6	7	8	9
A	**3.09**	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
B	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99
C	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	**5.35**	-1.45
D	-1.45	**3.62**	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
E	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
F	-1.45	-1.45	**4.01**	-1.45	-1.45	-1.45	-1.45	-1.45	**3.44**
G	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
H	-1.45	-1.45	-1.45	-1.45	**4.82**	-1.45	-1.45	-1.45	-1.45
I	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
J	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99
K	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
L	-1.45	-1.45	-1.45	**2.25**	-1.45	**1.36**	-1.45	-1.45	-1.45
M	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
N	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
O	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99
P	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	**3.75**	-1.45	-1.45
Q	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
R	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
S	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
T	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	**2.08**
U	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99
V	-1.45	-1.45	-1.45	**1.81**	-1.45	**2.74**	-1.45	-1.45	-1.45
W	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
X	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
Y	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
Z	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99
-	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45	-1.45
@	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
?	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
!	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
&	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 4-2 Exemple de « PSSMs séquentielles » générées à partir de la nouvelle formule de score basé sur une « BLOSUM like ». La matrice du haut donne les scores obtenus pour le jeu des trois mêmes blocs que la Figure 4-1 et rappelés en bas à gauche. La matrice de score est cette fois-ci construite à la manière d'une matrice BLOSUM en utilisant les acides aminés présents dans les séquences des blocs.

Cette matrice n'est pas très variée en raison des séquences utilisées pour la construire : je voulais juste montrer les différences entre les matrices au niveau des scores des acides aminés présents dans l'alignement. Bien entendu, cette matrice aurait été plus « riche » si les séquences dans les blocs étaient plus variées : comme on construit une « BLOSUM like » à partir des séquences contenues dans les CSBs, les mésappariements « exceptionnels » ne seront plus « dilués ».

Il est à noter que la matrice peut être plus diversifiée, donnant un caractère plus « flexible » aux « PSSMs séquentielles », si on ajoute aux séquences dans les blocs d'autres séquences proches d'elles.

4.3.2 Seuils pour la sélection des séquences dans les banques

Le « nouveau mode » de *Caliseq*, celui de recherche de P450s dans les banques, nécessitait la détermination d'un critère ainsi que la mise en place d'un système de seuil pour sélectionner les séquences identifiées par les CSBs comme étant des cytochromes P450 avec un bon taux de confiance. Dans l'outil *Caliseq*, l'unique critère spécifique à chaque séquence est le score d'alignement entre cette séquence et le jeu de blocs. Un seuil a été mise en place, en dessous duquel l'alignement n'est pas considéré comme significatif. Ce seuil, contrôlable par l'utilisateur en paramètre d'entrée au programme, a également connu diverses modifications dans les différentes versions de *Caliseq*.

4.3.2.1 Premier seuil, basé sur un ratio

Le premier système de calcul du seuil s'apparente à celui des poids de gaps dans le programme *SmartConsAlign* (cf. Annexe 3) : l'utilisateur fournit en entrée au programme un coefficient qui sert à ajuster le seuil à partir duquel le score d'alignement est significatif. Ce dernier dépend d'une part du score maximal S_{MAX} obtenu par un alignement parfait des blocs, à savoir sans insertion de gap ni délétion, et d'autre part, d'un score moyen S_{MEAN} obtenu par alignement de ces blocs sur une séquence « aléatoire ». Pour obtenir le score maximal S_{MAX}, il suffit juste d'aligner les blocs sur eux-mêmes, donnant le score maximal pour ces blocs. Pour obtenir le score moyen S_{MEAN}, le programme calcule le score du jeu de blocs contre les fréquences de base des acides aminés (celles de la banque Swiss-Prot). Le calcul du seuil *Threshold* est alors calculé à l'aide de la formule suivante :

$$Threshold = \frac{S_{MEAN} + coefficient \times (S_{MAX} - S_{MEAN})}{S_{MAX}}$$

Le score d'alignement pour chaque séquence de la banque est alors normalisé par le score maximal S_{MAX} puis comparé à ce seuil *Threshold* correspondant au seuil minimal de significativité pour l'alignement. La normalisation du score d'alignement était en effet nécessaire pour la comparaison : de valeur variable, il dépendait beaucoup de la nature, de la longueur de chaque séquence. L'idée est donc de récupérer les séquences dont les scores sont supérieurs à celui obtenu par une séquence aléatoire, légèrement ajustée par l'utilisateur.

Pour la recherche sur banque, cette méthode de sélection offre des résultats assez satisfaisants (détaillés au chapitre des résultats). Le problème de cette méthode vient du fait que le seuil n'est

calculé qu'à partir du jeu de blocs et des statistiques sur la banque, et non de l'information qu'apporte l'alignement des blocs sur chacune des séquence de la banque de séquences (c'est-à-dire le score individuel de chacune de ces séquences).

4.3.2.2 Deuxième seuil, limitant la longueur des séquences alignées

Pourquoi le score d'alignement est-il un frein à la méthode ? En fait, *Caliseq* offre aux blocs la possibilité de « glisser » sur la séquence cible sans pénaliser pour autant le score. Cette astuce concède aux blocs une extrême liberté pour leur positionnement, conduisant à des scores d'alignement élevés parfois même lorsque deux blocs sont distants d'un très grand nombre de résidus. Cela se traduit par l'existence d'un grand nombre de gaps '–' entre l'alignement de deux blocs sur une séquence. Ainsi, ce genre de cas était rencontré lors d'alignements du jeu des blocs sur de très longues séquences : certains blocs trouvaient leurs positions idéales en début de séquence, d'autres en fin de séquence, sans que le score d'alignement n'en soit affecté. Un tel résultat n'était pas compatible avec notre objectif, apportant dans les résultats un lot de faux positifs. Pour remédier à ce problème, un « cut off » a été implémenté. Ce dernier devait limiter la longueur de l'alignement des blocs sur la séquence cible. En pratique, il a suffi d'établir un seuil de distance, au-delà duquel l'alignement ne pouvait être considéré comme significatif. Cette distance a été mise en place entre la première position du premier bloc et la dernière position du dernier bloc. Concernant les P450s, la moyenne des longueurs de séquences tourne autour de 400 à 500 résidus. Certains P450s sont associés à leur domaine réductase, comme c'est le cas de $P450_{BM3}$, conduisant à des séquences plus longues de l'ordre de 700 à 800 résidus. J'ai choisi un « cut off » sur la distance entre 800 et 1200 résidus pour laisser de la marge, en particulier pour de nouvelles séquences qui associeraient au domaine oxydase un domaine réductase.

Par ailleurs, une deuxième vérification a été implémentée dans le programme : lorsque la taille de la séquence à aligner est inférieure à la taille totale des séquences contenues dans les blocs. En effet, l'outil *Caliseq* réalise l'alignement d'un ensemble de blocs sur une séquence, et il n'a pas été prévu que cette dernière soit plus petite que l'ensemble des blocs. Cette condition est vraie pour les deux modes de fonctionnement de *Caliseq*.

4.3.2.3 Amélioration du premier seuil par calcul d'un Z-score

Comme il a été signalé en section 4.3.2.1, le calcul du seuil de significativité pour le score d'alignement ne dépendait que du jeu de blocs et d'une statistique Swiss-Prot, et n'exploitait nullement l'information d'alignement sur les « vraies » séquences de la banque de données. Le calcul du seuil a donc été repensé conduisant à la mise en place d'un calcul de Z-score. Le Z-score

correspond à une mesure statistique permettant de quantifier la distance (mesurée en écart type) à laquelle une donnée individuelle se trouve par rapport à la moyenne du jeu de données dont est issue la donnée à traiter. Le z-score s'obtient par un processus appelé standardisation donnée par la formule suivante :

$$z = \frac{score - \mu}{\sigma}$$

Score correspond ici au score d'alignement, μ correspond à la moyenne des scores et σ l'écart type à la moyenne. L'écart type est donné par la formule suivante :

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(score_i - \mu_{score})^2}$$

N est le nombre total de séquences que *Caliseq* a pu aligner et pour lesquelles il a pu fournir un score (ce nombre dépend des contraintes imposées sur la longueur des séquences à aligner ou de la longueur total de l'alignement), $score_i$ le score d'alignement pour une séquence i donnée, et μ_{score} la moyenne des scores d'alignement. Il est à noter que je n'ai pas utilisé la formule de l'écart type sans biais, dans la mesure où je ne traitais pas un échantillon, mais l'ensemble des séquences contenues dans la base. Le z-score peut être représenté par une distribution normale comme le montre la Figure 4-3.

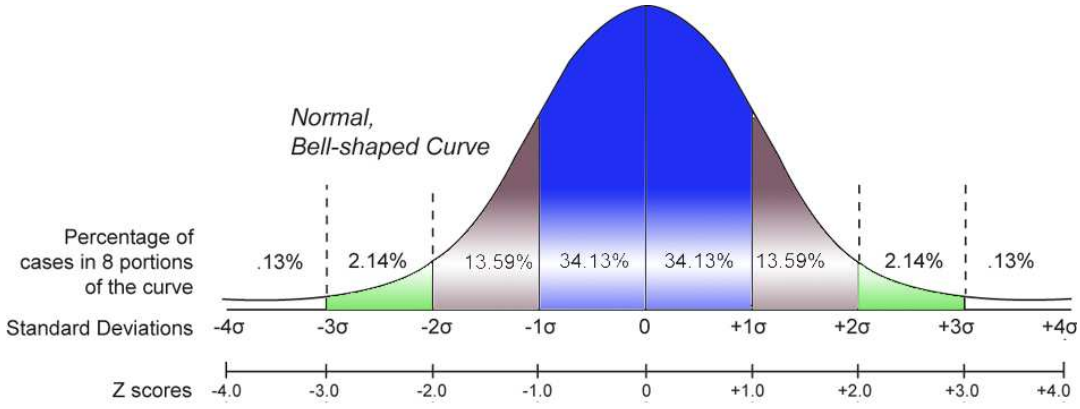


Figure 4-3 Diagramme d'écarts-type. La partie bleu foncée correspond à un écart type de la moyenne. Pour une distribution normale, cela correspond à une quantité de 68,27% de l'ensemble des données, tandis que deux écarts (bleu foncé et marron) comptent pour 95,45% et trois écarts-types (bleu foncé, marron et vert) compte pour 99,73%. (source : http://en.wikipedia.org/wiki/Standard_score)

L'ensemble des « grandes valeurs » de scores d'alignement, ceux qui sont significatifs, est présent dans la partie droite de la distribution. Ainsi, le seuil donné par l'utilisateur en paramètre d'entrée correspondra au seuil d'écart type au dessous duquel le score d'alignement n'est pas significatif. Plus grand sera ce seuil, plus on sera sélectif et moins il y aura de séquences récupérées.

Pour le calcul du z-score, il était nécessaire de « scanner » une première fois la banque de séquences afin de pouvoir mesurer la moyenne et l'écart type des scores d'alignement de toutes les séquences. En raison de la complexité et du nombre important de séquences à traiter dans une seule banque (pouvant atteindre l'ordre du million de séquences) les séquences présentant les meilleurs scores d'alignement étaient gardées lors du premier scan. Ainsi, la sélection ne s'opérait plus que sur un sous-ensemble, allégeant fortement le temps de calcul. Pour réaliser ce sous-ensemble, une structure en « arbre » a été implémentée dans le programme, retenant les séquences présentant les meilleurs scores d'alignement avec le jeu de blocs. Ainsi, c'est lors du second passage de *Caliseq* que le z-score est calculé pour chaque séquence. Les meilleures séquences sont alors filtrées sur le critère du seuil de leur z-score : elles correspondent aux séquences identifiées comme P450s selon notre critère.

4.3.3 Pondération des profils pour les séquences trop identiques

Lorsqu'il y a dans les alignements, aussi bien pour les séquences des structures de référence que pour le pré-alignement de séquences cibles, des séquences présentant une identité trop importante, ces dernières peuvent « biaiser » le profil en favorisant l'alignement final vers des séquences proches d'elles. C'est le cas par exemple de mon set de *templates* : sur les six structures de référence que j'ai choisies pour identifier les CSBs puis les positionner sur des séquences de P450s inconnus, trois d'entre elles sont de mammifère, de la sous famille 2C. Ces trois structures ont des séquences très proches, si bien qu'elles conduisent à la création d'un profil en leur faveur. Pour éviter de donner un poids trop important à ces séquences proches et donc pour favoriser les séquences peu représentées, un système de pondération a été mis en place. Comment cette pondération a-t-elle été implémentée ? La méthode de pondération résulte d'une discussion collective avec J. Pothier et AL. Abraham (Université Paris VI) dont le travail conduisait à un problème similaire au mien. Cette méthode est la suivante : sur un ensemble de N séquences, on calcule pour chaque séquence la fréquence moyenne d'identité entre cette séquence et les autres de l'ensemble. Le complémentaire de cette fréquence moyenne permettra alors d'accentuer le poids des séquences « sous représentées » et d'amoindrir celui des séquences présentant une identité trop importante avec les autres. La formule de la pondération *Weight(i)* pour une séquence i d'un alignement de N séquences, peut s'écrire de la manière suivante :

$$Weight(i) = 1 - \left(\frac{1}{N-1} \sum_{j=1;\, j \neq i}^{N-1} \%_{id}(i,j) \right)$$

Dans la formule, *j* est une autre séquence de l'ensemble, différente de *i*, et *%id(i,j)* est le taux d'identité entre les séquences *i* et *j*. Un exemple de calcul de pondération est présenté dans le Tableau 4.1.

Tableau 4.1 Exemple de calcul de pondération sur un ensemble de trois séquences A, B et C. A et B sont très proches tandis que C est plus éloignée des deux autres séquences.

Taux d'identité entres les séquences	***Séquences***	***Calcul de fmoy***	***fmoy***	***1– fmoy***
$\%_{id}(A,B) = 0,9$	A	$(\%_{id}(A,B)+\%_{id}(A,C))/2$	0,5	0,5
$\%_{id}(B,C) = 0,1$	B	$(\%_{id}(A,B)+\%_{id}(B,C))/2$	0,5	0,5
$\%_{id}(A,C) = 0,1$	C	$(\%_{id}(A,C)+\%_{id}(B,C))/2$	0,1	0,9

fmoy est la fréquence moyenne des identités entre les séquences
1-fmoy est le complémentaire de *fmoy*

4.4 Conclusion

L'outil *Caliseq* a brièvement été présenté le long des paragraphes de ce chapitre. Une description plus détaillée du programme figure en Annexe 3. Certains paragraphes de cette annexe pourraient sembler obscurs à des personnes non habituées au vocabulaire de la programmation, mais j'ai essayé de rendre cette partie annexe du manuscrit la plus claire possible pour mettre en valeur les différents mécanismes et principes sous-jacents à l'outil *Caliseq*, les différents problèmes auxquels j'ai été confronté lors de son développement, ou lors de son utilisation, et quelles solutions j'ai pu apporter.

Au final, l'outil *Caliseq* n'est pas qu'une simple amélioration du programme *SmartConsAlign* : en plus de réaliser un alignement de jeux de blocs sur une séquence cible, il offre la possibilité d'utiliser l'information contenue dans les CSBs pour explorer les banques de données de séquences ou les génomes nouvellement séquencés à la recherche de P450s. Cette nouvelle caractéristique du programme met en jeu un nombre important de paramètres, ajustables par l'utilisateur, qui influent sur les filtres de sélections. L'intérêt au-delà d'une simple identification de P450s ou même de création d'une banque « propre » de séquences de P450s serait à long terme la combinaison de la recherche sur banque et de la reconstruction tridimensionnelle des P450s non résolus sur le plan structural : *Caliseq* offrirait-il un moyen de constituer une banque de sites actifs de P450s ? L'existence d'une telle banque serait un support précieux pour l'étude des mécanismes de reconnaissance des P450s : à l'aide d'une chimiothèque, on pourrait tester les différents ligands potentiels dans les sites actifs de ces modèles et au travers des résultats de docking (molécules reconnues ou non reconnues, en face de l'hème ou sur

un site distant) commencer à formuler des hypothèses sur le mode de reconnaissance des différents P450 de microorganismes ou microsomaux. Avant de pouvoir se lancer dans une telle fiction, il faut bien sûr estimer la fiabilité de l'outil *Caliseq* pour identifier les séquences de P450s, et maîtriser l'alignement des blocs pour l'obtention du modèle le plus fiable. C'est là le sujet du prochain chapitre qui portera sur la place de nos nouvelles méthodes par rapport à celles déjà existantes. Apportent-elles oui ou non une réelle innovation, une valeur ajoutée ?

Troisième Partie

Résultats

CHAPITRE 5

Comparaison des méthodes

« La prospérité découvre nos vices
Et l'adversité nos vertus.»
Francis Bacon (1561– 1626 ap JC)

5.1 Introduction

Les propriétés structurales des P450s, ainsi que les problématiques que ces enzymes soulèvent, ont été décrites dans les précédents chapitres. La plupart des études conduites sur ces protéines *in silico* par des méthodes conventionnelles se sont heurtées à des limites liées à la complexité de ces P450s. La première d'entre elles est certainement l'obtention d'un modèle, rendu difficile en raison de la grande variabilité en séquence de cette superfamille. Plusieurs solutions ont été proposées pour contourner ces limites, en particulier, la construction fiable des modèles de cytochromes P450 qui passe selon nous par une recherche préalable de motifs structuralement conservés sur l'ensemble des P450s. L'idée sous-jacente est de nous appuyer sur le repliement extrêmement conservé dans cette superfamille comme information 3D à utiliser pour la reconstruction de modèles de P450s. Dans le but de vérifier et de valider cette hypothèse, des expériences de modélisation de P450 à basse identité de séquences ont été menées selon la méthode développée au laboratoire. Puis, elles ont été comparées à d'autres méthodes existantes. Ce nouveau chapitre présentera donc les résultats de comparaison des différentes méthodes, aussi bien sur la détection des informations 3D au sein de cette superfamille, que leur utilisation pour améliorer l'alignement en séquences. Le chapitre comporte également les comparaisons de modèles issus des différents alignements. Ce n'est qu'à partir de cette comparaison entre modèles qu'il sera possible de juger de l'intérêt apporté par l'information 3D (sous forme de blocs structuraux conservés) pour améliorer les alignements. Cette information 3D semblait en outre caractéristique des P450s : elle a été utilisée pour la recherche de P450s nouveaux dans des banques de séquences dans le cadre de la génomique exploratoire. Les résultats de ces recherches correspondent au dernier point traité dans ce chapitre.

Dans le cadre de ma thèse, j'avais pour premier objectif d'améliorer cette méthodologie préexistante de reconstruction à l'aide de blocs structuraux conservés, en vue de construire le modèle d'un CYP d'intérêt : le CYP 3A4. À mon arrivée au laboratoire, aucune structure n'était disponible et de nombreuses équipes de par le monde tentaient de produire un modèle par homologie. Hélas, la structure du CYP 3A4 a finalement été publiée sur la PDB au cours de ma première année de thèse. Cette structure est alors devenue ma structure de référence pour valider la méthodologie que notre équipe mettait au point. En la validant avec cette structure, peu commune par rapport aux autres CYP, cette méthode devait pouvoir s'appliquer à n'importe quel P450 inconnu dont l'identité en séquence avec les structures disponibles est faible.

5.2 Les CSBs, une utilisation innovante ?

5.2.1 Une signature 3D pour les protéines à repliement conservé

Partant sur le dogme qu'au cours de l'évolution, la structure d'une protéine est mieux conservée que sa séquence, nous avons choisi d'utiliser l'information de repliement conservé des P450s pour améliorer l'alignement de séquences, nécessaire à la reconstruction d'un modèle. Il est en effet intéressant de se demander pourquoi, en dépit d'une forte variabilité en séquence notamment dans la région N-terminale (et à degré moindre C-terminale) de l'enzyme, le repliement reste inchangé pour tous les P450s. Lors des alignements structuraux par les outils bioinformatiques disponibles (cf. sections 2.4.3 et 2.4.4) certaines régions semblaient mieux conservées que d'autres, ce qui a conduit à la recherche de blocs structuralement conservés (CSB) à travers toutes les structures disponibles. L'hypothèse formulée fut la suivante : ces CSBs pouvaient servir d'empreinte ou de signature caractéristique des P450s, et donc présents sur tous les CYPs, un peu à la manière de la signature héminique dite « cys-pocket » des P450s.

5.2.2 Obtention des CSBs par l'outil GOK

5.2.2.1 Description des blocs obtenus

La recherche des CSBs a d'abord été opérée par l'outil GOK sur trois jeux historiques de *templates* différents (cf. Tableau 3.1 de la page 134). Quel que soit le jeu de *templates* de départ, les mêmes blocs ont été trouvés, avec quelques variantes minimes liées au jeu des structures utilisées. Par ailleurs, parmi les blocs obtenus, certains correspondent à ceux initialement identifiés par P. Jean (Jean et *al.*, 1997). D'autres ont été ajoutés en raison de leurs bons rms et bien qu'ils soient généralement assez courts, ils ont été conservés pour permettre d'une part d'obtenir des « points d'ancrage » pour la reconstruction ultérieure, et d'autre part de couvrir une plus grande surface de la séquence de la structure à reconstruire[10]. À noter que l'identification et la longueur des blocs dépendent des valeurs de maille et de marge, utilisées durant la recherche : plus les valeurs de maille et de marge sont faibles, plus le bloc observé est pertinent et bien conservé. Un bloc apparaissant avec une maille et une marge faible est un bloc facilement identifiable par GOK. Les blocs des trois jeux historiques sont indiqués dans les Tableau 5.1, Tableau 5.2, et Tableau 5.3 respectivement. La position de chaque bloc est représentée sur une structure de P450 microsomal (la CYP 2C5) en Figure 5-1 et Figure 5-2.

[10] Il est rappelé que les contraintes spatiales pour la construction du modèle sont calculées prioritairement au niveau de ces zones.

Tableau 5.1 Position des 22 blocs structuraux obtenus avec GOK pour le jeu de 6 *templates* (d'après la thèse de N. Loiseau, 2002). En première colonne, la nomenclature de P. Jean, basée sur l'alignement structural de 3 structures bactériennes, et que nous avons suivie dans un souci de continuité. Dans la deuxième colonne, la nomenclature de Poulos, utilisée depuis 1985, basée sur la structure du $P450_{cam}$.

CSB	*Structures Secondaires*	*EryF 1OXA*	*BM3 2HPD*	*CAM 3CPP*	*Nor 1ROM*	*2C5 1DT6*	*Terp 3CPT*	*taille*	*mp–rms*	*maille marge*
1*	A	16 – 23	23 – 32	38 – 45	20 – 27	50 – 57	27 – 34	08	0,37	15/2
1**	ß1–1	28 – 35	38 – 45	52 – 59	30 – 37	63 – 70	39 – 46	08	1,36	20/3
1	ß1–2 + B	38 – 51	48 – 61	61 – 74	41 – 54	73 – 86	50 – 63	14	0,44	15/2
2A*	B'	79 – 84	72 – 77	89 – 94	73 – 78	99 – 104	81 – 86	06	0,40	15/2
2A**	Boucle B′ - C	89 – 95	85 – 91	99 – 105	85 – 91	111 – 117	101 – 107	07	0,81	30/2
2A	C	96 – 109	94 – 107	106 – 119	92 – 105	118 – 131	108 – 121	14	1,53	15/2
2B	D	114 – 132	115 – 133	127 – 145	113 – 131	141 – 159	129 – 147	19	1,00	15/3
3	E	138 – 160	139 – 161	148 – 170	138 – 160	164 – 186	151 – 173	23	3,15	20/2
4	F	161 – 175	172 – 186	171 – 185	161 – 175	192 – 206	174 – 188	15	1,94	20/3
5	G	185 – 205	205 – 225	193 – 213	184 – 204	233 – 253	212 – 232	21	0,84	10/2
6	H	210 – 217	232 – 239	218 – 225	209 – 216	262 – 269	237 – 244	08	0,08	15/3
7A	I	227 – 259	250 – 282	234 – 266	225 – 257	280 – 312	253 – 285	33	1,23	15/3
7B	J	260 – 269	283 – 292	267 – 276	258 – 267	313 – 322	286 – 295	10	0,13	15/1
8	K	272 – 288	312 – 328	279 – 295	270 – 286	343 – 359	298 – 314	17	1,06	15/3
9	ß1–4 + ß2–1	292 – 303	332 – 343	298 – 309	291 – 302	364 – 375	318 – 329	12	0,96	20/2
10	ß2–2 + ß1–3 + K′	304 – 326	345 – 367	310 – 332	303 – 325	376 – 398	330 – 352	23	1,85	15/1
11	Meander	327 – 334	369 – 376	333 – 340	326 – 333	399 – 406	353 – 360	08	0,40	20/2
12A	Cys Pocket	341 – 347	390 – 396	347 – 353	342 – 348	422 – 428	367 – 373	07	0,30	20/3
12B	L	348 – 372	397 – 421	354 – 378	349 – 373	429 – 453	374 – 398	25	0,61	15/2
13A	ß3–3	374 – 378	422 – 426	381 – 385	375 – 379	454 – 458	400 – 404	05	1,17	20/3
13B	ß4–1	383 – 391	429 – 437	387 – 395	384 – 392	464 – 472	406 – 414	09	4,86	25/3
13	ß4–2 + ß3–2	394 – 401	439 – 446	397 – 404	394 – 401	474 – 481	416 – 423	08	1,24	30/2

Note 1 : Les nouveaux blocs intermédiaires identifiés après P. Jean sont notés par une lettre complémentaire (A ou B) après le numéro de blocs, et parfois par un asterisque (* ou **) lorsqu'une nouvelle subdivision est apparue.

Note 2 : Certains blocs identifiés par des structures secondaires peuvent en réalité englober qu'une partie de la structure, ou parfois être plus large.

Tableau 5.2 Position des 27 blocs structuraux obtenus avec GOK pour le jeu des 4 *templates* proposés par M. Cottevieille (d'après le rapport de DEA de M. Cottevieille, 2003). Pour les nomenclatures, voir la légende du Tableau 5.1. M. Cottevieille identifie cinq nouveaux blocs par rapport à N. Loiseau.

CSB	***Structures Secondaires***	***EryF 1OXA***	***BM3 2HPD***	***51 1E9X***	***2C5 1DT6***	***taille***	***mp–rms***	***maille marge***
1*	A	14 – 28	23 – 37	24 – 38	48 – 62	15	0,70	20/2
1**	ß1–1	29 – 35	39 – 45	40 – 46	64 – 70	07	0,89	10/3
1	ß1–2 + B	36 – 51	46 – 61	47 – 62	71 – 86	16	1,35	15/2
2A*	B'	80 – 84	73 – 77	77 – 81	100 – 104	05	0,31	10/3
2A**	Boucle B' – C	92 – 94	88 – 90	90 – 92	114 – 116	03	–	20/3
2A	C	97 – 113	95 – 111	93 – 109	119 – 135	17	2,57	30/2
2B	D	114 – 132	115 – 133	110 – 128	141 – 159	19	1,36	15/2
2C	Boucle D – E	133 – 135	134 – 136	129 – 131	160 – 162	03	–	100/1
3	E	137 – 157	138 – 158	133 – 153	163 – 183	21	3,04	20/2
4	F	166 – 175	172 – 181	164 – 173	196 –205	10	0,25	–
5	G	182 – 205	202 – 225	191 – 214	230 – 253	24	0,51	10/1
6	H	210 – 217	232 – 239	224 – 231	262 – 269	08	0,07	10/2
7*	Boucle H – I	219 – 223	241 – 245	232 – 236	270 – 274	05	2,58	30/2
7A	I	226 – 259	249 – 282	241 – 274	279 – 312	34	4,57	15/3
7B	J	260 – 269	283 – 292	275 – 284	313 – 322	10	0,12	10/1
7B*	J(fin)	–	293 – 298	285 – 290	323 – 328	06	0,15	40/1
7C	J'	–	301– 311	294 – 304	332 – 342	11	7,37	60/1
8	K	272 – 290	312 – 330	305 – 323	343 – 361	19	1,12	15/2
9	ß1–4 + ß2–1	292 – 303	332 – 343	325 – 336	364 – 375	12	1,44	15/2
10A	ß2–2	304 – 306	345 – 347	337 – 339	376 – 378	03	–	15/2
10B	ß1–3 + K′	307 – 326	348 – 367	340 – 359	379 – 398	20	2,02	15/1
11	Meander	327 – 335	369 – 377	360 – 368	399 – 407	09	0,85	20/2
12A	Cys Pocket	337 – 345	386 – 394	380 – 388	418 – 426	09	3,48	30/2
12B	L	347 – 372	396 – 421	390 – 415	428 – 453	26	0,48	20/2
13A	ß3–3	374 – 380	422 – 428	416 – 422	454 – 460	07	1,21	20/2
13B	ß4–1	383 – 389	429 – 435	425 – 431	464 – 470	07	1,85	25/3
13	ß4–2 + ß3–2	391 – 401	437 – 447	433 – 443	473 – 483	11	2,97	30/2

Note : Cinq nouveaux blocs ont été identifiés par rapport à ceux identifiés par N. Loiseau : il s'agit des CSB 2C, 7*, 7B*, 7C, et celui issu de la subdivision du CSB 10 en deux.

Tableau 5.3 Positions des 24 blocs structuraux obtenus avec GOK pour le jeu de *templates* proposé dans ce travail de thèse. Pour les nomenclatures, voir la légende du Tableau 5.1. Les blocs identifiés sont un compromis entre ceux identifiés par N. Loiseau et ceux identifiés par M. Cottevieille.

CSB	Structures Secondaires	eryF 1OXA	BM3 2HPD	2C8 1PQ2	2C9 1OG5	2C5 1NR6	154C1 1GWI	taille	mp-rms	maille marge
0	A′	-	8 – 19	33 - 44	33 - 44	33 - 44	-	12	0,33	10/3
1*	A	14 - 25	23 – 34	48 - 59	48 - 59	48 - 59	17 - 28	12	0,21	20/2
1**	ß1-1	28 - 34	38 – 44	63 - 69	63 - 69	63 - 69	31 - 37	07	0,45	10/3
1	ß1-2 + B	35 - 52	45 - 62	70 - 87	70 - 87	70 - 87	39 - 56	18	1,78	15/3
2A*	B'	78 - 85	71 - 78	99 - 106	99 - 106	99 - 106	79 - 86	08	1,75	20/2
2A**	B' – C	89 - 93	85 - 89	111 - 115	111 - 115	111 - 115	91 - 95	05	0,29	10/3
2A	C	96 - 113	94 - 111	118 - 135	118 - 135	118 - 135	98 - 115	18	2,05	20/2
2B	D	114 - 132	115 - 133	141 - 159	141 - 159	141 - 159	116 - 134	19	1,02	10/3
3	E	137 - 157	138 - 158	163 - 183	163 - 183	163 - 183	140 - 160	21	2,80	15/3
4	F	160 - 175	171 - 186	191 - 206	191 - 206	191 -206	163 - 178	16	1,49	20/2
5	G	182 - 205	202 - 225	230 - 253	230 - 253	230 - 253	184 - 207	27	0,65	10/2
6	H	210 - 216	232 - 238	262 - 268	262 - 268	262 - 268	212 - 218	07	0,03	10/2
7A	I	227 - 259	250 - 282	283 - 315	283 - 315	280 - 312	228 - 260	33	0,87	10/3
7B	J	260 - 268	283 - 291	316 - 324	316 - 324	313 - 321	261 -269	09	0,07	10/1
7C	J'	-	304 - 309	338 - 343	338 - 343	335 - 340	-	06	0,21	10/1
8	K	274 - 290	314 - 330	348 - 364	348 - 364	345 - 361	275 - 191	17	0,92	10/3
9	ß1-4 + ß2-1	292 - 303	332 - 343	367 - 378	367 - 378	364 - 375	294 - 305	12	1,04	15/2
10	ß2-2 + ß1-3 + K′	304 - 327	345 - 368	379 - 402	379 - 402	376 - 399	306 - 329	24	3,81	15/3
11	Meander	328 - 335	370 - 377	403 - 410	403 - 410	400 - 407	331 - 338	08	0,31	20/2
12A	Cys Pocket	339 - 346	388 - 395	423 - 130	423 - 430	420 - 427	-	08	1,47	20/3
12B	L + turn	347 - 372	396 - 421	431 - 456	431 - 456	428 - 453	351 - 376	26	0,32	10/3
13A	ß3-3	374 - 380	422 - 428	457 - 463	457 - 463	454 - 460	378 - 384	07	1,14	15/3
13B	ß4-1	381 - 389	-	465 - 473	465 - 473	462 - 470	385 - 393	09	1,91	20/3
13	ß4-2 + ß3-2	390 - 401	436 - 447	475 - 486	475 - 486	472 - 483	394 - 405	12	3,21	30/2

Note 1: Un nouveau bloc a été identifié (CSB0) par rapport aux deux autres jeux de blocs. Les blocs identifiés sont plus proches en nombre et en position de ceux de N. Loiseau (un seul bloc pour le CSB 10, pas de CSB 2C ni de CSB 7* ou 7B*). En revanche, comme pour le jeu de M. Cottevieille, le CSB 7C a été ici aussi identifié.

Note 2 : Initalement, la structure du CYP 2B4 (1po5) était présente dans ce jeu. En raison de sa conformation ouverte, elle empêchait la détection de blocs sous GOK : elle a donc été retirée du jeu

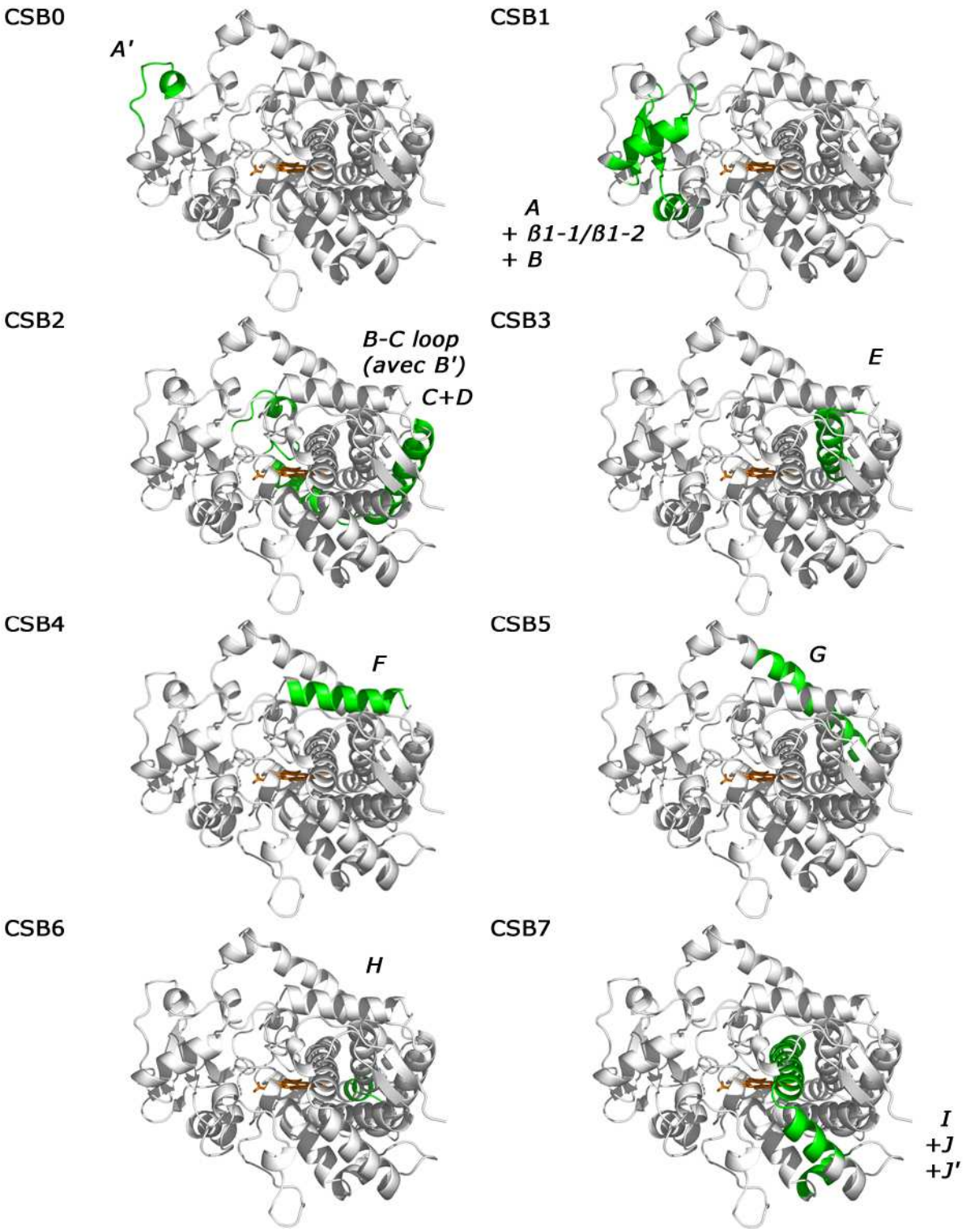


Figure 5-1 Représentation des CSB0 à CSB7 sur la structure du CYP 2C5 (1nr6). La protéine est en représentation « cartoon » en gris et l'hème figure en orange. Les CSB sont représentés en vert. Les structures secondaires correspondantes sont également annotées. Ceux-ci correspondent à des blocs postérieurs à la nomenclature de P. Jean. Les images ont été générées par PYMOL

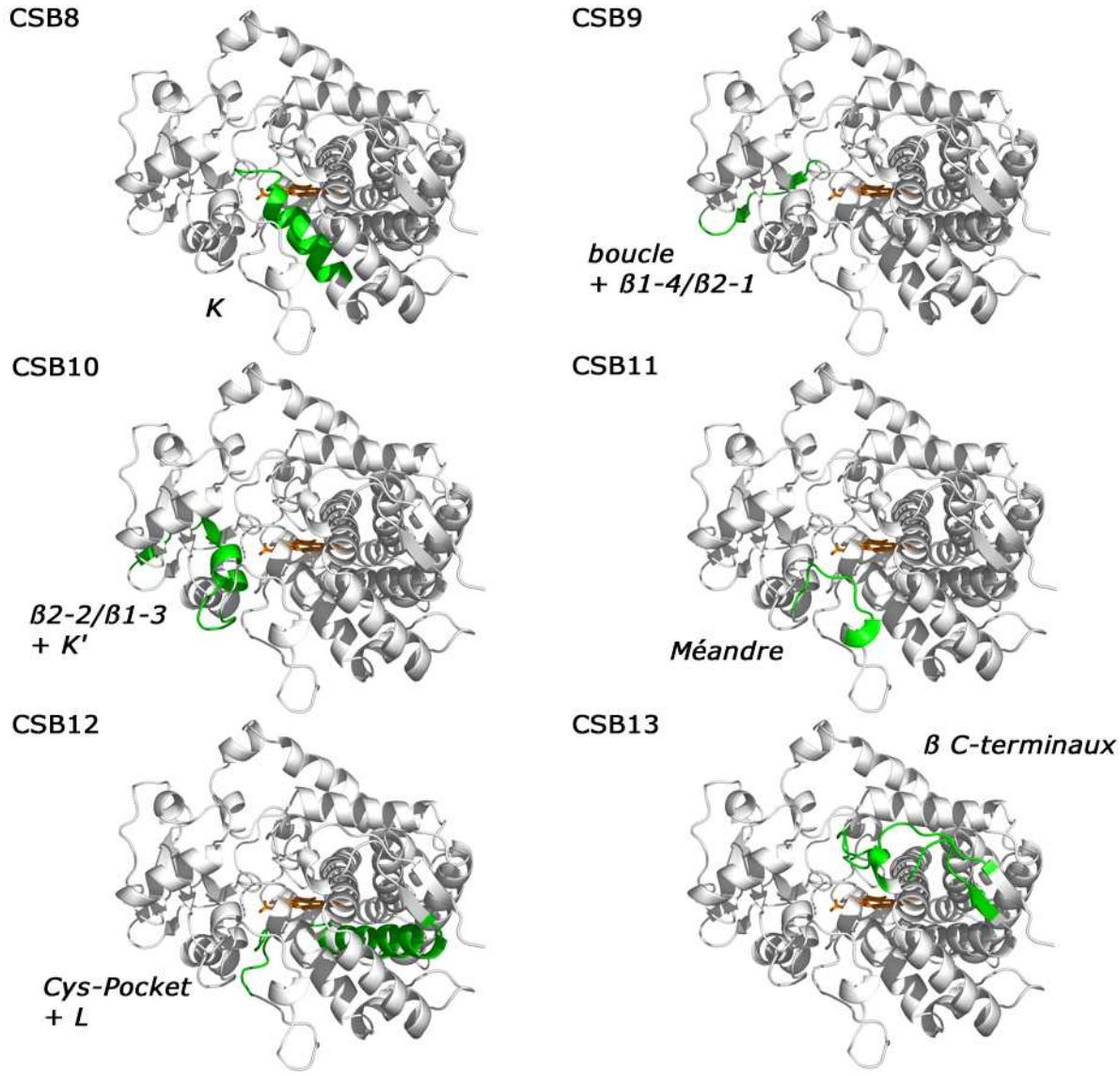


Figure 5-2 Représentation des CSB8 à CSB13 sur la structure du CYP 2C5 (1nr6). Pour les légendes, se reporter à ceux de la Figure 5-1.

Une moyenne de 24 blocs (CSBs) a été identifiée sur les trois jeux de *templates*. Ces CSBs ont été initialement numérotés de 1 à 13 selon la nomenclature de P. Jean (Jean et *al*., 1997) dans la première reconstruction d'un P450 utilisant cette méthode. Cette nomenclature a été conservée avec de légères variations lorsque certains blocs étaient subdivisés en plusieurs fragments (par exemple CSB1, CSB2 ou encore CSB7), ou lorsqu'un nouveau bloc fut identifié. Cette évolution des blocs a été nécessaire dès que les premières structures de P450 microsomaux ont été publiées. Le jeu initial de P. Jean avec 13 blocs historiquement identifiés s'appuyait en effet sur l'alignement de 3 structures bactériennes.

Ainsi, le **bloc 0** (noté CSB0 sur la Figure 5-1) n'est trouvé que dans mon jeu de *templates* qui comporte le plus de structures issues de mammifères. Ce bloc est relativement court et correspond à une partie d'hélice placée en amont de l'hélice A, située au niveau de l'hélice d'ancrage à la membrane chez les P450s microsomaux. Une particularité du CYP 154C1, d'origine bactérienne, peut être soulignée : 1gwi présente un brin β à cette position, qui n'a été observé que sur cette structure X de P450s. Cette structure confirme la règle observée pour les autres structures bactériennes : le bloc 0 ne pouvait pas être identifié sur cette structure. En revanche, il est surprenant d'identifier ce bloc dans la structure du $P450_{BM3}$ qui est d'origine bactérienne. Dans la littérature, rien ne semble pourtant indiquer que ce P450 soit ancré à la membrane. On sait juste qu'il est le seul CYP naturel présentant sur la même chaîne polypeptidique la partie hémoprotéique et la partie réductase, ce qui fait de lui un P450 proche des P450s microsomaux.

Le **bloc 1*** également court, est constitué de la première partie de l'hélice A en N-terminal de la protéine. Ce bloc est identifié dans les 3 jeux de *templates*, sur toutes les structures. Sa taille varie selon le jeu, de 8, pour le jeu de N. Loiseau à 15 résidus pour le jeu de M. Cotevieille.

Le **bloc 1**** contient le premier brin du feuillet β_1. Il fait partie des blocs simples à identifier sous GOK avec des valeurs faibles de mp-rms et de taille de maille.

Le **bloc 1** contient quant à lui deux SSE : le deuxième brin du feuillet β_1 ($\beta_{1\text{-}2}$), l'hélice B ainsi que la boucle qui les lie. C'est ici un exemple concret qui montre que les blocs ne sont pas uniquement limités aux SSEs isolés.

Les **blocs 2A*** et **2A**** appartiennent à la longue boucle B–C décrite comme une région importante pour la reconnaissance du substrat, formant une face du site actif très mobile. Le bloc 2A* correspond à la micro-hélice B'. Sachant que cette hélice est située dans un zone soumise à des changements conformationnels de la protéine (cf. section 1.4.4.2) et de surcroît, est connue pour être labile chez les CYPs de mammifères –on la trouve aussi bien structurée que déstructurée–, on peut se demander si ce bloc est pertinent ou non. Il semble bien conservé localement sur les différents *templates*, en dépit d'une grande variabilité aussi bien en séquence qu'en organisation spatiale (boucle B-C). À noter que le mp-rms de ce bloc est plutôt bas dans les jeux de N. Loiseau et M. Cottevieille (~0.5 Å), contrairement à celui de mon jeu de *templates*. Sachant que dans mon jeu, il y a davantage de cristaux de P450s de mammifère, un mp-rms supérieur (~1,75 Å) pour ce bloc n'est pas surprenant : dans la structure du CYP 2C5, l'hélice B' disparaît presque en totalité pour laisser la place à une

boucle présentant une allure d'hélice. Le bloc 2A** correspond à une portion de boucle B'–C. Celle-ci est située à proximité de l'hème et contient des acides aminés impliqués dans la reconnaissance du substrat (SRS-1, voir Figure 1-18 à la page 44). La boucle constituant le bloc 2A** correspond en quelque sorte au levier qui permet à la boucle B–C sa grande mobilité. Étant donné sa petite taille, et son rôle important, il est assez surprenant de l'identifier en tant que bloc : on s'attend en effet à une boucle très variable pour rendre compte de la diversité des substrats reconnus. Pourtant, le repliement dans l'espace des (α,τ) décrit cette région comme conservée.

Les **blocs 2A et 2B** sont contigus quels que soient les différents jeux de *templates*. La description en SSEs de cette zone correspond aux trois hélices, C, C' et D (dans cet ordre dans la séquence). L'hélice C' a déjà été observée par Hasemann (Hasemann et *al.*, 1995) qui l'a décrite comme désordonnée. Lors de l'identification des blocs par GOK, le logiciel décompose cette région en deux blocs seulement, coïncidant au milieu de cette hélice C'. L'angle formé au niveau de C' entre les hélices C et D est variable. Chacune des parties de part et d'autre de C' (hélice C et hélice D) est localement bien conservée.

Le **bloc 2C** n'est qu'observé dans le jeu de M. Cottevieille. Il est extrêmement court (3 résidus) et correspond à un morceau de la boucle D–E. Étant donné les paramètres de maille et de marge (100/1) et un mp-rms non renseigné, ce bloc n'est probablement pas significatif.

Le **bloc 3** est constitué de l'hélice E. L'hélice E', trop courte et non adjacente à l'hélice E, n'a pas pu être identifiée comme région structuralement conservée.

Le **bloc 4** comprend une partie de l'hélice F. Cette hélice est de longueur variable sur tous les *templates* utilisés. Dans la littérature, les hélices F et G formant le toit du site actif sont spécifiques selon le P450 observé : elles sont à la fois de longueur variable – plus longues chez les bactériens que chez les mammifères par exemple – et également de structuration variable (présence ou non de micro-hélices dans la boucle F–G). L'identification de ce bloc fut par conséquent délicate : la plus petite hélice (correspondant à une structure hélicoïdale dans l'espace des (α,τ)) pouvait se superposer à plusieurs positions différentes sur les longues hélices F (également des structures hélicoïdale dans l'espace des (α,τ)). À noter qu'en plus de la longueur variable de l'hélice, ce bloc est également très mal conservé en séquence. M. Cottevieille n'a pas été en mesure de trouver la même longueur pour ce bloc que N. Loiseau et moi-même, montrant ainsi la difficulté à obtenir ce bloc. Hasemann a

également rencontré des difficultés pour l'alignement structural de cette région et propose deux alignements possibles selon qu'il considère la conservation structurale ou la conservation séquentielle.

Dans le cas du **bloc 5** correspondant à l'hélice G, Hasemann a rencontré des problèmes identiques au bloc précédent. A l'inverse, le logiciel GOK a pu identifier aisément une solution unique avec un mp-rms très bas quel que soit le jeu de *templates* (0.84 Å pour le plus élevé chez N. Loiseau). Ce bloc est par ailleurs relativement long par rapport aux autres blocs, il est l'un des trois plus longs dans tous les jeux de *templates*. Cette observation est assez surprenante lorsque l'on sait que les hélices F et G sont justement des hélices de longueur variable, relativement courtes chez les mammifères (ex : CYP 3A4)…

Le **bloc 6**, formé par l'hélice H est également identifié facilement par GOK. Il semblerait d'ailleurs, au vu de tous les *templates* que l'ensemble des blocs 5 + 6 soit bien conservé spatialement : la boucle qui les relie en revanche est très variable selon les structures, et est responsable notamment de la scission du bloc en 2 parties : bloc 5 et bloc 6. Cette observation renforce l'idée d'utiliser les blocs comme armature pour le P450 à reconstruire et de laisser les régions inter-blocs se reconstruire sans contrainte apparente.

Le **bloc 7** est sans nul doute le plus long de tous : il est constitué des hélices I (bloc 7A), J (bloc 7B) et une partie de l'hélice J' (bloc 7C), observé chez les P450s de mammifères. Suivant les paramètres de recherche, il peut être obtenu en une, deux, trois ou cinq parties. Les parties correspondant à l'hélice J' (bloc 7C), n'ont été observées que dans le jeu de *templates* de M. Cottevieille et le mien, plus riche en P450s de mammifère, et il n'est pas identifié dans le jeu de N. Loiseau très dominé par les P450s bactériens. Le bloc 7C qui n'est d'ailleurs pas présent dans toutes les structures ($P450_{eryF}$ et CYP 154C1 en sont dépourvus) montre un mp-rms élevé de 7,4 Å, laissant supposer que l'hélice J' est certainement très fluctuante. De même, le **bloc 7*** correspondant à la boucle H–I n'est présent que dans le jeu de M. Cottevieille. Il semble trop court pour être retenu dans les blocs structuralement bien conservés de la famille.

Les quatre blocs suivants, du **bloc 8** au **bloc 11**, constituent un ensemble structural continu et bien conservé : les blocs y sont quasiment contigus. En fait, cet ensemble est obtenu en plusieurs parties en raison des insertions ou délétions de quelques résidus dans un des *templates*, mais les quatre blocs sont superposables simultanément. Il est à noter que seul le set de M. Cottevieille dispose de cinq blocs au

lieu de quatre : le **bloc 10** est séparé en deux dissociant le brin $\beta_{2\text{-}2}$ du reste formé par l'hélice K' et $\beta_{1\text{-}3}$, alors que ces structures sont trouvées dans un seul bloc pour le set de N. Loiseau et le mien.

Comme pour les blocs 2A et 2B, le **bloc 12**, correspondant à la Cys-Pocket et l'hélice L, est constitué de deux parties jointives et « articulées » autour d'un acide aminé voisin de la cystéine liant l'hème.

Le dernier bloc, **bloc 13**, est constitué principalement de brins β relié par des boucles. De façon moins évidente que pour la boucle B-C, cette région formée par les brins β intervient généralement dans l'obstruction du site actif. Bien que structuralement conservée, il est à noter que cette région est extrêmement variable au niveau de sa séquence primaire.

5.2.2.2 Apport des structures de mammifères pour les CSBs par comparaison aux structures bactériennes

Au fil des années dans la PDB, les structures bactériennes ont été complétées par des structures de mammifères. Ainsi, pour mon jeu de blocs, il y a eu autant de structures bactériennes que de structures de mammifères – uniquement de CYPs de la famille 2C rappelons-le – ayant servi comme *templates*. Cet apport en structures microsomales a entraîné de légères variations dans la détermination et la fiabilité des blocs (nombre de blocs, tailles des blocs, mp-rms de chaque bloc etc.). À première vue, les structures microsomales présentent les mêmes éléments de structure secondaire que ceux présents dans les P450s bactériens. Toutefois, une divergence visible de ces éléments avec ceux de l'enzyme microbienne la plus proche ($P450_{BM3}$) est observée. La plus grande différence entre ces structures avec la structure microbienne est dans le positionnement relatif du feuillet β N-terminal vis-à-vis des hélices de la partie centrale (cf. Figure 1-15 page 41) : un décalage spatial est constaté. On observe également une différence au niveau des longueurs des brins β entre les structures de mammifères et les structures bactériennes. Pourtant, pris individuellement, chacun des éléments structuraux se superpose relativement bien.

Dans le jeu de NL, la structure du P450 bactérien qui se superpose le mieux à l'unique structure de mammifère présente dans ce jeu (CYP 2C5) est celle du $P450_{Terp}$. Les différences les plus importantes entre le CYP 2C5 et le $P450_{Terp}$ se situent d'une part au niveau du positionnement des hélices F et G qui couvrent le site actif et déterminent son accès, et d'autre part au niveau des boucles entre les SSEs.

Dans mon jeu, c'est la structure du CYP 154C1 qui est la plus proche de celle du CYP 2C5. Ici aussi, de faibles différences sont remarquées, notamment au niveau des hélices F' et G' uniquement observables chez la structure du lapin (CYP 2C5), au niveau de l'hélice F plus courte chez le CYP microsomal, mais également au niveau de certaines boucles (par exemple B–C), plus longue chez le CYP 154C. À noter que le CYP 2C5 montre parfois au niveau de ces boucles des débuts de structuration en hélice α, là où la structure bactérienne n'en montre pas. Cette caractéristique est commune aux trois structures bactériennes par opposition aux trois structures microsomales. C'est d'ailleurs une des raisons qui me pousse à considérer le bloc 4 comme un bloc assez peu significatif (ou « flou ») : dans ce bloc, la courte hélice F de la structure microsomale a été identifiée à différentes positions (plusieurs choix de superposition proposés par GOK) sur les hélices plus longues des structures bactériennes, et il a fallu choisir visuellement et sans critère spécifique, la meilleure superposition pour définir le bloc.

En résumé, l'enrichissement en structures microsomales a entraîné principalement l'apparition de nouveaux blocs par rapport à ceux initialement identifiés : ceux-ci pouvant être nouveaux (comme les blocs 0 et 7C) ou correspondant à des sous-divisions de blocs déjà existants (bloc 10 devenant 10A et 10B dans le jeu de M. Cottevieille). Ces structures apportent également des changements dans la détection et la fiabilité des blocs initialement identifiés. Finalement, « l'ossature » en elle-même des P450s demeure globalement inchangée dans la mesure où un même nombre de blocs (de 22 à 27) est identifié dans les trois jeux, d'autant plus qu'il s'agit des mêmes blocs (positionnement conservé) observés dans les trois jeux.

5.2.2.3 Position des CSBs par rapport aux SRS

Les six SRS définis par Gotoh (Gotoh et *al.*, 1992) (cf. Figure 1-18 page 44) sont présents dans les blocs identifiés. Seuls les SRS-2, SRS-3, SRS-4 situés respectivement sur les hélices F, G et I sont présents dans leur intégralité dans les blocs correspondants (respectivement 4, 5 et 7). Même si les SRS-1 et SRS-6 ne sont pas présents en intégralité dans les blocs 2 et 13 respectivement, leurs résidus de liaison au substrat (identifiés par Gotoh sur le $P450_{cam}$) sont en revanche présents dans ces blocs. Dans les structures microsomales (CYP 2C5, CYP 2C8 et CYP 2C9), le SRS-5 qui est trouvé dans la partie polypeptidique entre l'hélice K et le feuillet $\beta_{1\text{-}4}$ (entre les blocs 8 et 9) formant le côté opposé du site d'interaction du substrat avec l'hélice I, n'est présent qu'en partie dans les blocs. Ces positionnements des SRS dans les blocs, pour les trois jeux, confortent l'idée d'information « pertinente » présente dans les blocs.

Il est surprenant toutefois qu'un bloc « flou » comme le bloc 4, puisse contenir un SRS dans son intégralité. En revanche, le fait que le SRS-5 soit plutôt situé dans une région inter-bloc n'est pas surprenant : il correspond à une région entre l'hélice K et le feuillet $\beta_{1\text{-}4}$ qui varie significativement d'une structure à l'autre, de bactérienne à microsomale (ex : entre les CYP 2C et le $P450_{BM3,}$ on relève une différence de 3,3 Å de rmsd sur les C_α dans cette région).

Lorsqu'on compare en fait les SSEs (hélice F, G, et le feuillet C-terminal), qui portent les autres SRS (SRS-2, SRS-3 et SRS-6 respectivement), il s'avère qu'une même divergence (structurale) est également observée sur l'ensemble des structures, de même ordre de grandeur (par rapport au $P450_{BM3}$ par exemple). Ces différences aux niveaux de ces SSEs bordant la poche du site actif, pourraient être une cause de l'orientation et le positionnement du substrat dans le site actif et par conséquent la sélectivité de l'enzyme pour les différents substrats.

5.2.2.4 Alignement des blocs entre les trois jeux de templates

Dans chacun des jeux de *templates*, les valeurs de déviation multiple mp-rms et s-rms varient parfois significativement. Il serait donc intéressant de savoir si une règle peut être dégagée en comparant les mp-rms des trois jeux : peut être serait-il possible de dégager des blocs « flous » et des blocs « nets ».

Malgré le fait qu'il comporte moins de structures *templates*, c'est pourtant le jeu de M. Cotevieille qui montre les valeurs les plus hautes. Il comporte des valeurs élevées pour les blocs 1, 2A, 2B, 3, 7*, 7C, 8, 9, 10, 12 et tous ceux de la série 13. Ces fortes valeurs résultent également du besoin de rechercher des blocs plus longs et de nouveaux blocs en forçant les paramètres de maille et de marge, pour couvrir la fraction maximale de la séquence en bloc. Ce jeu est constitué de quatre structures dont trois bactériennes et une microsomale. Il se trouve que ces quatre structures sont plutôt divergentes, conduisant à des détections de blocs par GOK à valeur élevée de mp-rms. En réalité, M. Cottevieille n'est pas la seule à avoir identifié des blocs à valeur élevée de mp-rms : pris indépendamment, chaque jeu en présente au moins au niveau d'un bloc.

Il est possible toutefois de classer ces valeurs élevées de mp-rms d'un jeu à l'autre. Ainsi, dans mon jeu de *templates*, la région N-terminale est marqué par de hautes valeurs de mp-rms (blocs 1 et 2A*, structure $\beta_{1\text{-}2}$ et hélice B, cf. Tableau 5.3). Une explication a déjà été fournie pour ce cas, où l'hélice B' est très « fluide » dans les structures microsomales. Dans le jeu de NL, au niveau de cette région, seul le bloc 1** (brin $\beta_{1\text{-}1}$) porte une forte valeur de mp-rms. Dans le jeu de M. Cottevieille,

pourtant marqué par de fortes valeurs de mp-rms, on trouve des tendances inversées : par exemple dans les blocs 4 et 5 (hélices F et G) les valeurs de mp-rms (~0.5 Å) sont trouvées plus basses que dans les deux autres jeux (~1,00 Å). Enfin, la série des blocs 13 présente des valeurs élevées de mp-rms dans les trois jeux, le bloc le plus « flou » dans le jeu de N. Loiseau est trouvé au niveau de la structure $\beta_{4\text{-}1}$ tandis que dans les deux autres jeux, il est situé au niveau des structures $\beta_{4\text{-}2}$ et $\beta_{3\text{-}2}$.

Certains blocs à valeur élevée de mp-rms, sont trouvés en revanche en accord sur l'ensemble des trois jeux : c'est le cas de l'hélice E (bloc 3) par exemple, mais également du bloc 10, comprenant 3 SSEs (brin $\beta_{2\text{-}2}$, brin $\beta_{1\text{-}3}$, K'). En dépit des valeurs élevées de mp-rms, ces blocs peuvent toutefois être considérés comme des blocs « fiables » (ou « nets ») : les valeurs élevées de mp-rms peuvent être expliquées par la divergence d'un seul *template* dans le jeu par rapport aux autres, qui se retrouve dans les trois jeux, expliquant de ce fait une valeur élevée de mp-rms similaire pour les trois jeux de blocs.

Globalement, en dépit des différences observées sur les trois jeux, on peut établir une certaine tendance pour des régions plus variables que d'autres (notion de blocs « flous ») en se basant sur les constats précédemment décrits (utilisation des valeurs de mp-rms inter jeu) : la variabilité en séquence au niveau de la région C-terminale est donc accentuée par une variabilité structurale, les blocs situés dans cette partie étant marqués par des valeurs élevées de mp-rms, exprimées à des positions différentes dans les blocs de la série 13. Pour ces mêmes raisons (variabilité entre les jeux des valeurs élevées de mp-rms), les blocs du feuillet β en N-terminal peuvent être également considérés comme des blocs « flous », très mal conservés structuralement (blocs de la série 1). Ces tendances ne sont pas liées à l'apport des nouvelles structures de P450s microsomales, elles étaient déjà présentes dans les structures bactériennes. Enfin, on peut également inclure parmi ces blocs « flous » ceux correspondant à la boucle B–C comprenant l'hélice B', et ceux comportant les hélices F et G. Le caractère « flou » de ces blocs 4 et 5 est beaucoup plus marqué dans les alignements comportant plus de structures microsomales.

Toutes ces remarques sur un plan structural, se retrouvent également observées dans l'alignement réalisé par *Caliseq*, des blocs des trois différents jeux de *templates* sur la séquence du CYP 3A4, comme présenté sur la Figure 5-3. Ainsi, les blocs de la série 1 (CSB1*, CSB1**, et CSB1) ne se trouvent pas à la même position sur les trois différents jeux de blocs. Il en est de même pour les blocs de la série 2 (CSB2A*, CSB2A** et CSB2A) et de la série 13 (CSB13A, CSB13B, CSB13). En revanche, ni le CSB4 ni le CSB5 ne semblent poser de problème de positionnement dans les trois jeux, ce qui est assez déroutant, compte tenu de ce qui a été dit précédemment. Cela dit, comme il sera vu

plus tard, le positionnement de ces blocs à des endroits similaires sur d'autres séquences cibles n'est pas toujours observé.

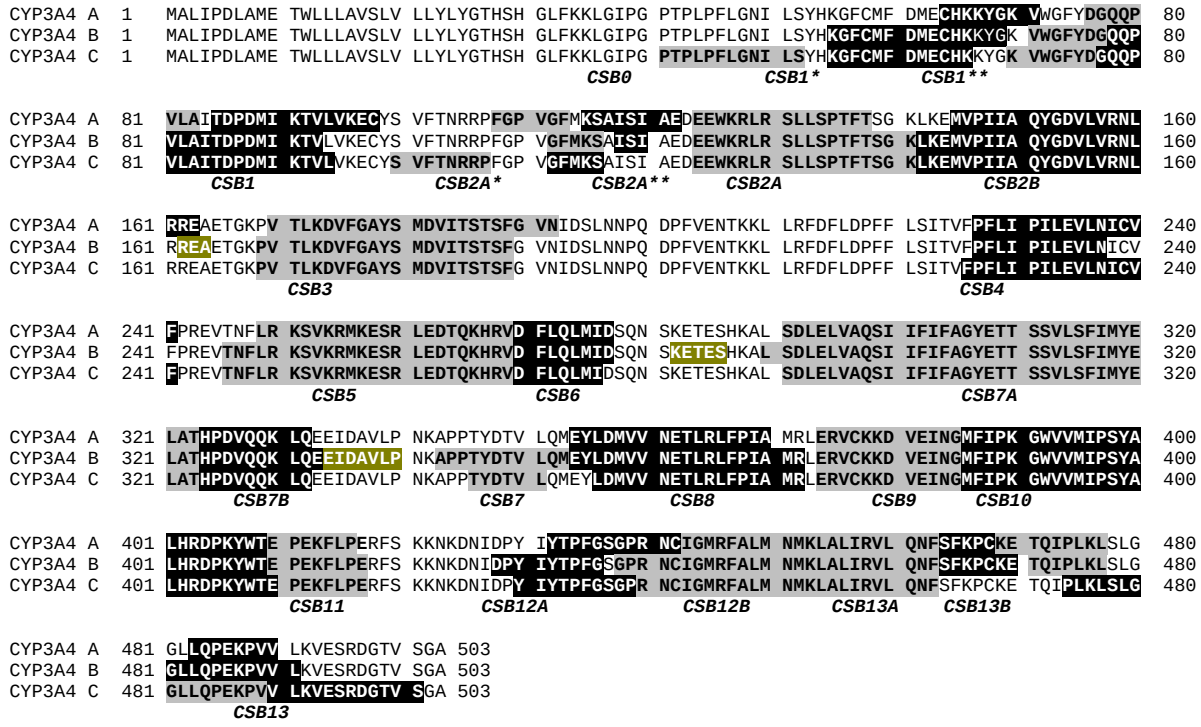


Figure 5-3 Comparaison du positionnement des blocs pour les trois différents jeux de *templates* sur la séquence cible du CYP 3A4. Les couleurs utilisées pour délimiter les blocs sont les mêmes sur les trois jeux. Le jeu de templates de N. Loiseau figure sur la première ligne (A), celui de M. Cottevieille en seconde ligne (B) et le mien en dernière ligne (C). Un décalage sur les trois jeux de blocs est observé aussi bien en région N-terminal au niveau de la série des CSB1 (1*, 1**, 1) et en région C-terminale au niveau de la série des CSB13 (13A, 13B, 13).

5.2.3 Intérêt d'un jeu de blocs généralisés

En raison des divergences sur les positionnements relatifs des blocs entre les trois jeux de *templates*, j'ai dû établir un jeu de blocs général correspondant à un compromis des sous-alignements. Ce dernier a pour but de lever les ambiguïtés de positionnement des blocs qui posent des difficultés à *Caliseq*, en particulier dans la série des blocs 1 en N-terminal ou encore de la série des blocs 13 en C-terminal. Ce jeu de blocs global est présenté Figure 5-4 et Figure 5-5 : il symbolise l'ossature générale conservée d'un P450.

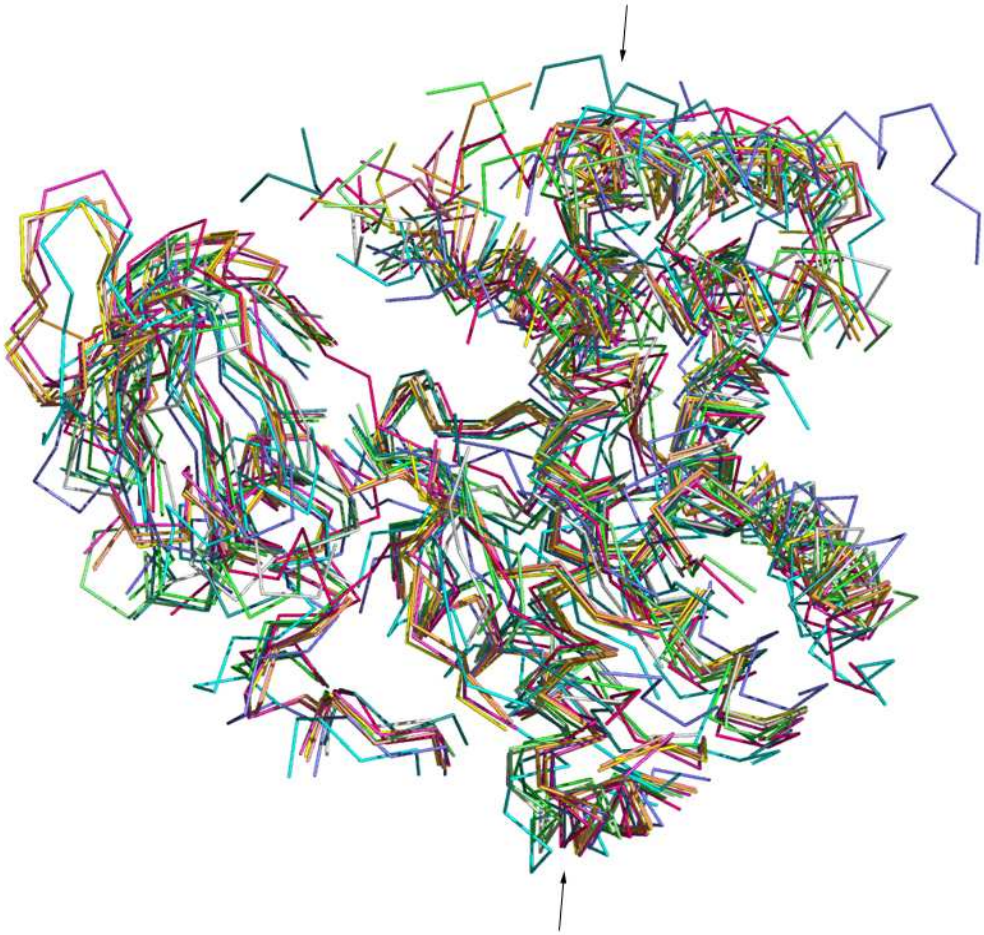


Figure 5-4 Blocs structuraux en représentation carbones C_α déterminés à partir d'un consensus des trois jeux de blocs (11 structures au total). Les structures ont été superposées sur les atomes du squelette des hélices I et J (blocs 7A et 7B) indiquées par les flèches noires. Le code couleur est le suivant : P450eryf en vert, P450BM3 en cyan, CYP 2C8 en violet, CYP 2C9 en jaune, CYP 2C5 (1nr6) en rose clair, CYP 154C1 en gris, CYP 51 en bleu, CYP 2C5 (1dt6) en orange, P450nor en vert clair, P450terp en vert kaki et enfin P450cam en rose foncé.

Le compromis des trois différents jeux de blocs (cf. méthode décrite à la section 3.2.3) fournit un jeu global composé de blocs plus courts que ceux rencontrés dans les trois jeux de blocs initiaux. Toutefois, pour certains blocs, aucune modification n'a été apportée : il s'agit de blocs généralement robustes, comme les CSBs de la série 7 ou de la série 12. À l'inverse, ceux de la série 1 et surtout ceux de la série 13 ont subi un raccourcissement drastique pour réaliser le consensus : ils étaient déjà à l'origine des principaux désaccords entre les différents jeux de blocs. Par ailleurs, selon le raccourcissement imposé aux blocs de la série 13, leur positionnement par le programme *Caliseq* sur la séquence du CYP 3A4 par exemple, se voit perturbé dans cette région : c'est la preuve que les blocs identifiés à cette région C-terminale sont plutôt peu fiables.

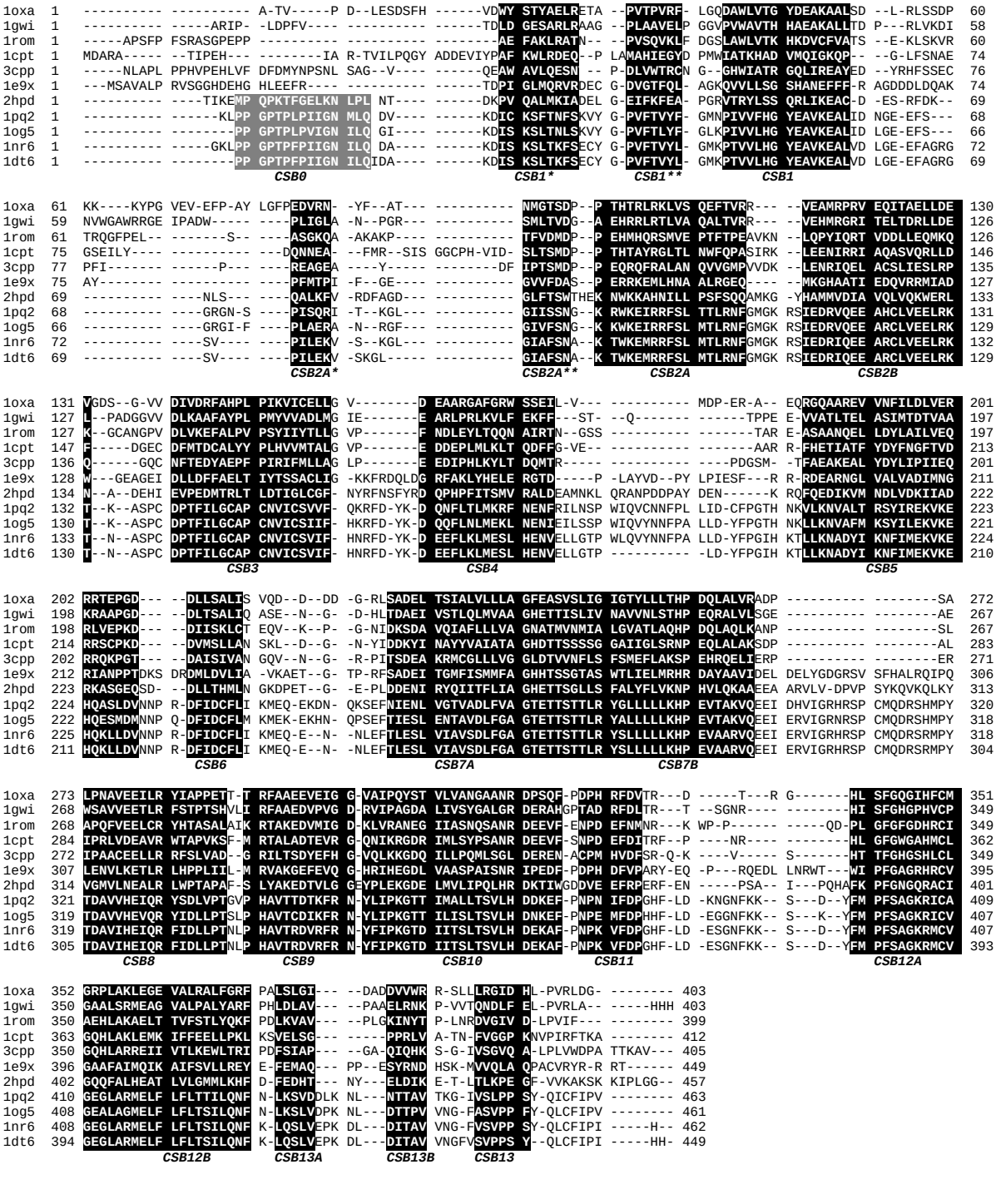


Figure 5-5 Alignement structural des 11 structures *templates* : $P450_{eryF}$ (1oxa), CYP 154C1 (1gwi), $P450_{nor}$ (1rom), $P450_{terp}$ (1cpt), $P450_{cam}$ (3cpp), CYP 51 (1e9x), $P450_{BM3}$ (2hpd), CYP 2C8 (1pq2), CYP 2C9 (1og5), CYP 2C5 (1nr6), CYP 2C5 (1dt6). L'ordre des structures dans l'alignement a été établi arbitrairement. Les blocs structuraux (CSBs) sont indiqués en blanc sur fond noir. La partie inter-bloc a été alignée à la main. La partie N-terminale est souvent tronquée en raison de son alignement incertain. Le bloc 0 détecté sur le jeu de *templates* dominé par les protéines microsomales est surligné en gris.

L'élaboration de ce jeu de blocs global ne change pas le positionnement des blocs « robustes ». Ils permettent en revanche de trancher sur les incertitudes et désaccords entre les alignements des trois différents jeux de blocs. En revanche, comme ce jeu de blocs global se base sur un compromis séquentiel, il n'est pas certain que les sous-alignements obtenus seraient identiques à ceux obtenus par une recherche 3D par GOK. En raison de la limite du logiciel à identifier les CSBs sur un nombre de templates trop important, je n'ai pas pu effectuer cette vérification sur le jeu complet à 11 *templates*, et j'ai considéré que le jeu de bloc global retraduisait correctement l'information 3D des CSBs, sachant que les sous-alignements mis en commun provenaient eux-mêmes d'une identification 3D de blocs.

5.2.4 Alignement de toutes les structures représentatives de P450 dans la PDB

Au cours de la thèse, avec l'émergence de nouvelles méthodes ainsi que la puissance croissante des machines de calcul, il a été enfin possible de déterminer des blocs structuralement conservés sur l'ensemble des *templates* des P450s. En Avril 2007, 29 structures de P450 non redondantes étaient présentes dans la PDB. Cette identification a été rendue possible par l'utilisation de l'outil GAKUSA, développé par M.Carpentier (Université Paris VI) au cours de sa thèse, dont le principe repose sur le « Gibbs Sampling » (cf. section 2.4.5.2 à la page 110). Les structures utilisées pour la recherche de blocs conservés sont présentées sur le Tableau 3.2 à la page 141. Des 29 structures présentées sur ce tableau, une structure a été finalement retirée car elle comportait trop de régions non résolues dans sa structure, empêchant GAKUSA de déterminer correctement les CSBs. Il s'agit de la structure du CYPs $P450_{Oxyb}$ (1lfk) d'origine bactérienne. Les résultats de la recherche de CSBs par GAKUSA sur l'ensemble des *templates* différents disponibles dans la PDB sont montrés sur le Tableau 5.4.

Les CSBs identifiés par GAKUSA sur l'ensemble des *templates* différents de P450 disponibles dans la PDB devaient servir initialement à vérifier la solidité ou l'incertitude des blocs du jeu global présenté à la Figure 5-5 : quelles étaient les différences observables avec l'apport de nouveaux *templates* (publiés en 2006 et 2007), et où étaient situées ces différences ?

Tableau 5.4 Positions des blocs structuraux obtenus avec GAKUSA sur un jeu de 28 *templates* : la structure du P450Oxyb a été retirée du jeu initial car elle comportait trop de régions manquantes. En première colonne, la nomenclature de P. Jean, basée sur l'alignement de 3 structures bactériennes, dans la deuxième colonne, la nomenclature de Poulos.

CSB	Structures Secondaires	cam 1RE9	BM3 2IJ2	nor 2JFB	EryF 1Z8O	OxyC IUED	epok 1Q5D	119 1IO7	taille
1	β1–1+β1–2+B	44-68	41-65	34-58	31-55	12-36	41-65	12-36	25
2B-3	D+E	126-160	127-161	126-160	126-160	100-134	131-165	103-137	35
7	I+J	222-262	248-288	223-263	225-265	195-235	238-278	194-234	41
8	K	266-290	310-334	268-292	270-294	236-260	283-307	236-260	25
9-10A	β1–4+β2–1+β2–2	291-311	336-356	294-314	295-315	261-281	309-329	261-281	21
10B-11	β1–3+K'+Meander	317-329	362-374	320-332	321-333	287-299	335-347	287-299	13
12-13	CysPocket+L+β3–3	337-373	390-426	342-378	341-377	307-343	355-391	307-343	37
13	β4–1+β4–2+β3–2	377-396	430-449	381-400	383-402	345-364	394-413	345-364	20
		121 1N40	**152A1 1IZO**	**154A1 1ODO**	**154C1 1GWI**	**158A2 1S1F**	**175A1 1N97**	**51 1X8V**	
1	β1–1+β1–2+B	34-58	38-62	32-56	35-59	43-67	35-59	42-66	25
2B-3	D+E	119-153	119-153	126-160	129-163	130-164	114-148	120-154	35
7	I+J	217-257	226-266	226-266	226-266	225-265	205-245	240-280	41
8	K	262-286	272-296	271-295	271-295	270-294	250-274	303-327	25
9-10A	β1–4+β2–1+β2–2	288-308	297-317	298-318	297-317	297-317	275-295	328-348	21
10B-11	β1–3+K'+Meander	314-326	323-335	324-336	323-335	323-335	299-311	354-366	13
12-13	CysPocket+L+β3–3	335-371	353-389	344-380	345-381	343-379	326-362	384-420	37
13	β4–1+β4–2+β3–2	374-393	393-412	383-402	387-406	385-404	364-383	426-445	20
		st 1UE8	**PIKC 2CD8**	**terp 1CPT**	**199A2 2FR7**	**2A6 2FDU**	**2B4 1SUO**	**2C5 1NR6**	
1	β1–1+β1–2+B	47-71	44-68	43-67	49-73	70-94	67-91	66-90	25
2B-3	D+E	134-168	130-164	139-173	136-170	156-190	153-187	152-186	35
7	I+J	229-269	227-267	251-291	235-275	285-325	282-322	278-318	41
8	K	274-298	272-296	296-320	280-304	348-372	345-369	341-365	25
9-10A	β1–4+β2–1+β2–2	300-320	298-318	321-341	305-325	374-394	371-391	367-387	21
10B-11	β1–3+K'+Meander	326-338	324-336	347-359	331-343	400-412	397-409	393-405	13
12-13	CysPocket+L+β3–3	346-382	344-380	367-403	351-387	429-465	426-462	422-458	37
13	β4–1+β4–2+β3–2	384-403	383-402	406-425	390-409	473-492	470-489	466-485	20
		2C8 1PQ2	**2C9 1R9O**	**2D6 2F9Q**	**3A4 1TQN**	**8A1 2IAG**	**2R1 2OJD**	**1A2 2HI4**	
1	β1–1+β1–2+B	66-90	66-90	70-94	73-97	66-90	78-102	77-101	25
2B-3	D+E	152-186	152-186	160-194	158-192	150-184	165-199	174-208	35
7	I+J	281-321	281-321	289-329	289-329	267-307	294-334	301-341	41
8	K	344-368	344-368	352-376	352-376	337-361	357-381	364-388	25
9-10A	β1–4+β2–1+β2–2	370-390	370-390	378-398	377-397	366-386	383-403	390-410	21
10B-11	β1–3+K'+Meander	396-408	396-408	404-416	403-415	393-405	409-421	416-428	13
12-13	CysPocket+L+β3–3	425-461	425-461	433-469	432-468	431-467	438-474	448-484	37
13	β4–1+β4–2+β3–2	469-488	469-488	476-495	474-493	477-496	480-499	490-509	20

Initialement, je pensais que l'utilisation d'un jeu de *templates* plus important (28 structures contre 11 dans le jeu global) conduirait à l'obtention de blocs moins nombreux et plus courts. Les résultats montrent que les blocs identifiés sous GAKUSA sont effectivement moins nombreux que dans le jeu

global (8 blocs identifiés contre 23 blocs sur le jeu global) mais de taille moyenne plus importante : la plupart des blocs identifiés sous GOK se retrouvent agglomérés lors de l'identification par GAKUSA. Ainsi, les blocs de la série 1 sont à présents réunis en un seul bloc de 25 résidus. Il en de même pour les blocs de la série 7 et 13. À noter que les blocs identifiés par GAKUSA bouleversent un peu la nomenclature mise en place par P. Jean : certains blocs correspondent à une réunification de blocs de séries voisines. C'est le cas par exemple du second bloc déterminé par GAKUSA dans le tableau, qui réunit le bloc CSB2B et le bloc CSB3 (d'autres exemples sont observés, voir Tableau 5.4).

Toutefois, l'utilisation de GAKUSA est limitée dans la recherche exhaustive de tous les blocs identifiés sous GOK : la version actuelle du programme ne permet pas d'identifier les blocs de la boucle B–C (incluant l'hélice B') et les hélices F, G, H. À l'instar de GOK, GAKUSA identifie chaque bloc pas à pas. Lorsqu'un résultat de bloc n'est pas convenable, il faut recommencer la recherche depuis la toute première étape, et modifier les paramètres de recherche une fois à l'étape précédant celle qui aboutissait à un résultat non convenable[11]. Schématiquement, lorsque le résultat de bloc n'est pas satisfaisant à l'étape ***5,*** il faut repartir de l'étape ***1*** et modifier les paramètres une fois à l'étape ***4***. Ces blocs jugés non convenables, correspondent en fait à ceux présentant des morceaux d'hélices de certains *templates* qui devraient constituer un bloc *X,* et qui se retrouvent finalement attribués avec les hélices des autres structures au niveau d'un bloc *Y*. Par exemple, l'hélice F de la structure du $P450_{OxyC}$ est identifiée dans le bloc des hélices H pour les autres structures. Cet exemple illustre d'ailleurs la dernière étape où il n'a plus été possible d'aller plus loin pour identifier de nouveaux blocs sans avoir ce cas de figure (autrement dit, il est possible d'identifier d'autres blocs mais ils comprennent des erreurs d'attribution). La difficulté à identifier correctement les blocs dans ces régions est probablement liée à la similarité des trajectoires en coordonnées internes (α,τ) –hélices principalement– ainsi que la variabilité des éléments de structures secondaires au niveau de ces régions (hélice B' parfois déstructurée et à des positions variables le long de la boucle B–C, et les hélices F et G sont variables).

[11] À la différence de GOK qui utilise des paramètres de maille et de marge, GAKUSA utilise un paramètre de longueur de bloc à ajuster.

```
Best score for this pattern scaled to len= 21 F=1116.82132 G/free param=   1.29120
1CPT OXIDOREDUCTASE(OXYGENASE)                ALADTEVRGQNIKRGDRIMLS 321  341
1IO7 OXIDOREDUCTASE                           TKERVKLGDQTIEEGEYVRVW 261  281
1N40 OXIDOREDUCTASE                           ATADIQVGDVLVRKGELVLVL 288  308
1N97 ELECTRON TRANSPORT                       LERPLLLGEDRLPPGTTLVLS 275  295
1IZO OXIDOREDUCTASE                           VKKDFVWNNCEFKKGTSVLLD 297  317
1GWI OXIDOREDUCTASE                           AAEDVPVGDRVIPAGDALIVS 297  317
1Q5D OXIDOREDUCTASE                           ARQDLEYCGASIKKGEMVFLL 309  329
1RE9 OXIDOREDUCTASE                           LTSDYEFHGVQLKKGDQILLP 291  311
1S1F OXIDOREDUCTASE                           ALEDVEIKGVRIRAGDAVYVS 297  317
2IAG ISOMERASE                                AMPMADGREFNLRRGDRLLLF 366  386
1UE8 STRUCTURAL GENOMICS, UNKNOWN FUNCTION    TKEKVKIRDQVIDEGELVRVW 261  281
1UED OXIDOREDUCTASE                           AIKDVVIDGQLIKAGDYVLCS 300  320
1X8V OXIDOREDUCTASE                           AKGEFEVQGHRIHEGDLVAAS 328  348
1Z8O OXIDOREDUCTASE                           AAEEVEIGGVAIPQYSTVLVA 295  315
2CD8 OXIDOREDUCTASE                           PVEPVDLDGTVIPAGDTVLVV 298  318
2OJD OXIDOREDUCTASE                           TSEDAVVRGYSIPKGTTVITN 383  403
2FR7 OXIDOREDUCTASE                           TTRDVELAGATIGEGEKVLMF 305  325
1ODO OXIDOREDUCTASE                           VTDIALPDGRTIARGEPILAS 298  318
2IJ2 OXIDOREDUCTASE                           KEDTVLGGEYPLEKGDELMVL 336  356
1JFB OXIDOREDUCTASE                           AKEDVMIGDKLVRANEGIIAS 294  314
2FDU OXIDOREDUCTASE                           VKKDTKFRDFFLPKGTEVYPM 374  394
1SUO OXIDOREDUCTASE                           VTKDTQFRGYVIPKNTEVFPV 371  391
1PQ2 OXIDOREDUCTASE                           VTTDTKFRNYLIPKGTTIMAL 370  390
1R9O OXIDOREDUCTASE                           VTCDIKFRNYLIPKGTTILIS 370  390
1NR6 OXIDOREDUCTASE                           VTRDVRFRNYFIPKGTDIITS 367  387
2F9Q OXIDOREDUCTASE                           TSRDIEVQGFRIPKGTTLITN 378  398
2HI4 OXIDOREDUCTASE                           TTRDTTLNGFYIPKKCCVFVN 390  410
1TQN OXIDOREDUCTASE                           CKKDVEINGMFIPKGVVVMIP 377  397

 1-> | 1|19| 9|34| 5|34|17|12|30| 4| 1| 3| 8|21|32| 4| 1| 3| 3| 2|35|
 2-> | 1|16|12|35| 5|35|14|15|27| 3| 2| 5| 9|19|31| 5| 2| 2| 2| 3|35|
 3-> | 0|20| 9|35| 5|35|12|16|26| 3| 3| 4| 9|20|30| 6| 1| 2| 2| 3| 1|
 4-> |35|19|11| 2| 2|35|10|17|27| 1| 1| 4| 9|21|31| 4| 3| 1|35| 1| 0|
 5-> | 0|19| 8|32| 6|34|17|12|29| 4| 1| 2| 8|20|31| 5| 0| 2| 3| 5|35|
 6-> | 0|19|10| 2| 4|35|12|16|24| 3| 4| 4| 8|20|29| 6| 1| 0| 5| 2|34|
 7-> | 1|18|11|33| 6|35|16|13|29| 4| 1| 3| 9|19|31| 6| 1| 2| 3| 3|35|
 8-> | 1|17|10|35| 4| 0|16|13|29| 4| 1| 4| 8|20|31| 4| 1| 3| 0| 2| 1|
 9-> | 1|18|11|35| 4| 0|16|12|30| 5| 0| 4| 8|21|30| 5| 1| 1| 2| 3| 0|
10-> | 2| 3| 2| 6|24| 9|29| 2| 2| 5| 2| 4| 7|21|29| 5| 3| 0| 0| 1|35|
11-> | 0|19|11| 0| 2| 0|13|15|27| 2| 1| 3| 9|22|31| 6| 1| 2| 2| 3|35|
12-> |35|21| 7| 0| 4|35|15|13|28| 4| 1| 3| 8|21|29| 4| 2| 2| 3| 2|35|
13-> | 0|18|11|35| 4| 0|15|13|29| 4| 2| 3| 9|20|31| 5| 1| 0| 1| 3| 0|
14-> |35|19|10| 1| 3|35|17|12|29| 4| 2| 5| 9|20|32| 5| 1| 1| 2| 3|35|
15-> | 3|19|11| 2| 2| 0|14|13|29| 2| 1| 5| 8|20|31| 7| 0| 1| 3| 4| 0|
16-> | 2|17|12| 2| 2|35|15|12|29| 4| 1| 4| 9|20|32| 6| 1| 2| 2| 3|35|
17-> | 0|20| 9|35| 4|35|14|13|29| 4| 0| 4| 9|21|32| 5| 1| 3| 2| 2|35|
18-> |18|11| 1| 5| 3| 7|23| 8|30| 5| 3| 4| 9|20|30| 6| 2| 1| 3| 1|34|
19-> |17|11|33| 6| 3|16|10|24|24| 3| 1| 4| 9|20|32| 5| 0| 2| 3| 2| 0|
20-> | 1|17|12| 0| 4| 0|12|16|26| 3| 2| 3|10|20|30| 5| 1| 1| 1| 3| 0|
21-> | 3|17|13| 1| 2|35|13|14|28| 4| 0| 4| 9|20|32| 5| 2| 2| 1| 3|35|
22-> | 3|18|11| 2| 3|34|15|13|28| 4| 1| 5| 9|19|32| 6| 1| 2| 2| 3| 0|
23-> | 3|17|12| 1| 4| 0|12|16|27| 3| 1| 5| 9|20|32| 7| 2| 0| 1| 4|35|
24-> | 3|17|12| 0| 4|35|12|16|27| 3| 1| 4| 9|20|32| 6| 2| 1| 2| 3|35|
25-> | 2|18|11| 0| 3| 1|12|15|26| 4| 1| 4| 9|20|32| 6| 2| 1| 1| 4|35|
26-> | 1|17|11|35| 5|35|16|12|30| 5| 1| 4| 9|20|33| 6| 1| 1| 2| 3|34|
27-> | 2|18|10|33| 5| 1|16|12|30| 5| 0| 4| 9|20|33| 5| 1| 1| 2| 3|34|
28-> | 1|18|11|33| 4| 0|14|15|26| 5| 1| 4| 9|20|31| 5| 3| 1| 2| 3|35|
```

Figure 5-6 Exemple de sortie de GAKUSA pour le bloc 9-10A ($\beta_{1\text{-}4}/\beta_{2\text{-}1}/\beta_{2\text{-}2}$). La première partie du résultat (en haut) correspond à la description des templates, leurs séquences respectives au niveau du bloc identifié ainsi que les positions du bloc sur les templates. La partie du bas correspond à l'alignement des angles α à chaque position du bloc (valeurs discrètes de 0 à 35 par tranches de 10° d'angle). Dans cet exemple, on peut voir l'effet de « moyenne » exercé dans la sélection des blocs, par exemple la dixième structure (2iag) montre à la position 5 un angle α très écarté de la moyenne de la colonne, de même dans d'autres positions du début de bloc. Pourtant, ce bloc est sélectionné parce que le score global sur tout le bloc reste favorable. Dans GOK, une position divergente comme cet exemple suffirait à provoquer une rupture de blocs, ce qui explique le morcellement et la taille des blocs dans les deux méthodes.

Ce qui est nouveau par rapport aux blocs identifiés par GOK, c'est la longueur trouvée pour chaque bloc : est-ce un artefact du logiciel (lié aux paramètres trop « permissifs » fournis par l'utilisateur) ? Peut-on attribuer une interprétation possible à cette observation ? Il existe en effet sous

GAKUSA des paramètres semblables à la maille de GOK mais ces paramètres sont réglés par défaut pour être « stringent » et GAKUSA ne permet pas la détection de nombreux blocs. Dans la version actuelle du programme, il n'est possible d'attribuer ce paramètre qu'une seule fois, au lancement du programme. Ainsi, j'ai dû utiliser un paramètre moins « stringent » que ceux définis par défaut, ce qui a permis de détecter plus de blocs, de taille plus importante. Par ailleurs, la similarité des trajectoires communes semble plus marquée lorsque l'on dispose de plus de *templates* : avec seulement deux *templates*, les différences dans les trajectoires sont facilement décelées tandis qu'avec un jeu plus conséquent de *templates*, il y a un effet de « moyenne » (lié au calcul de score) qui tend à atténuer les écarts des valeurs des angles (α,τ) trouvées (cf. Figure 5-6). Les *templates* pris séparément peuvent être un peu divergents au niveau de ces blocs, mais pris ensemble, présentent des régions structuralement conservées au niveau des blocs.

Ainsi, les blocs obtenus par GAKUSA soulèvent de nombreux points de discussion : les matrices PSSMs générées à partir de ces blocs seront-elles plus informatives que celles issues préalablement sous GOK ? On peut le penser dans la mesure où elles sont issues d'un jeu de *templates* global, à savoir toutes les structures non redondantes. Cependant, une telle longueur de bloc supposerait implicitement une même longueur de la zone concernée pour toutes les structures inédites de P450 (non référencées encore dans la PDB). Le problème de longueur ne pose en pratique aucun problème : *Caliseq* repose sur une programmation dynamique et cherche le meilleur positionnement pour les blocs, quitte à introduire des gaps dans la séquence cible. Ainsi, la réunion des blocs présente l'avantage d'éviter les alignements « imprécis » pour les blocs en N- et C-terminal par exemple. Le positionnement d'un long bloc sur la séquence cible pourrait être plus robuste que le positionnement de trois petits blocs. En effet, les trois blocs se positionneront au mieux sur la séquence cible, sachant que les gaps ne sont pas pénalisés pour leur glissement, pouvant conduire à de mauvais placements des blocs. Au contraire, pour positionner le bloc long, il faudra introduire des gaps dans la séquence cible, qui ont un coût et la longueur du bloc impose une délimitation naturelle de positionnement pour le bloc. Par exemple, lorsque deux blocs identifiés par GOK sont contigus au niveau d'une SSE, ces blocs peuvent être séparés entraînant la perte de la SSE présent à cheval sur les deux blocs. Ce problème n'existera pas en revanche sous GAKUSA, qui intègrera les deux blocs identifiés par GOK en un seul bloc.

5.2.5 Comparaison des blocs avec d'autres méthodes d'alignement 3D publiées

5.2.5.1 Les autres méthodes d'alignement structural

Outre la recherche de blocs par GAKUSA, d'autres logiciels d'alignement de structures (présentés en sections 2.4.3 et 2.4.4) ont été appliqués sur les 11 *templates* du jeu global (ceux de la Figure 5-5), afin de comparer les blocs obtenus par la méthode GOK aux alignements structuraux proposés par ces autres logiciels. En raison de leur nombre important, j'ai limité les comparaisons aux logiciels d'alignements structuraux de type multiple uniquement, et qui diffèrent tous par un descripteur particulier (C_α, SSE, géométrie…). Il est à noter que certains logiciels d'alignement structuraux ne permettent pas d'imposer les structures à aligner, mais utilisent un système de recherche automatique de *templates*, similaire à BLAST : ces logiciels ont été exclus de la comparaison. Enfin, il faut se rappeler que les logiciels utilisés pour la comparaison sont plus récents dans leur conception que le logiciel GOK. Ces différents logiciels ainsi que leurs caractéristiques sont présentés sur le Tableau 5.5.

Tableau 5.5 Liste des logiciels d'alignement structuraux utilisés pour évaluer les CSBs identifiés par GOK. Excepté le logiciel Matras qui n'accepte qu'une comparaison de 9 *templates*, les alignements structuraux par ces logiciels ont été opérés sur les 11 structures *template* du jeu global.

Logiciel	*Description*	*Année*	*Descriptif utilisé*	*Référence*	*Lien internet*
MultiProt	**Multi**ple Alignment of **Prot**ein Structures	2004	Géométrie	Shatsky et *al.*, 2004	http://bioinfo3d.cs.tau.ac.il/MultiProt
SSM	**S**econdary **S**tructure **M**atching	2003	SSE	Krissinel et Henrick, 2004	http://www.ebi.ac.uk/msd-srv/ssm/
Matras	**MA**rkovian **TRA**nsition of protein **S**tructure	2000	C_α et SSE	Nishikawa, 2000	http://biunit.aist-nara.ac.jp/matras/
Vorolign	Fast structure alignment using Voronoi contacts	2007	Cellules de Voronoï	Birzele et al., 2007	http://www.bio.ifi.lmu.de/Vorolign/
GAPS	**G**aussian-based **A**lignment of **P**rotein **S**tructures	2000	C_α et représentation Gaussienne	Mestres et al., 2000	NC

NC : Non communiqué

Contrairement à l'outil GOK qui ne fournit que les fragments de séquences structuralement conservées sur un ensemble de *templates*, la plupart des autres logiciels utilisés montre le résultat de l'alignement structural sous forme d'un alignement global de séquences. Pour effectuer des comparaisons avec les résultats de GOK, des blocs structuralement conservés ont été prélevés de ces alignements multiples résultants : on peut en effet assimiler à des blocs CSB les régions de l'alignement dépourvues de gap sur l'ensemble des *templates* issus des autres logiciels (du Tableau 5.5).

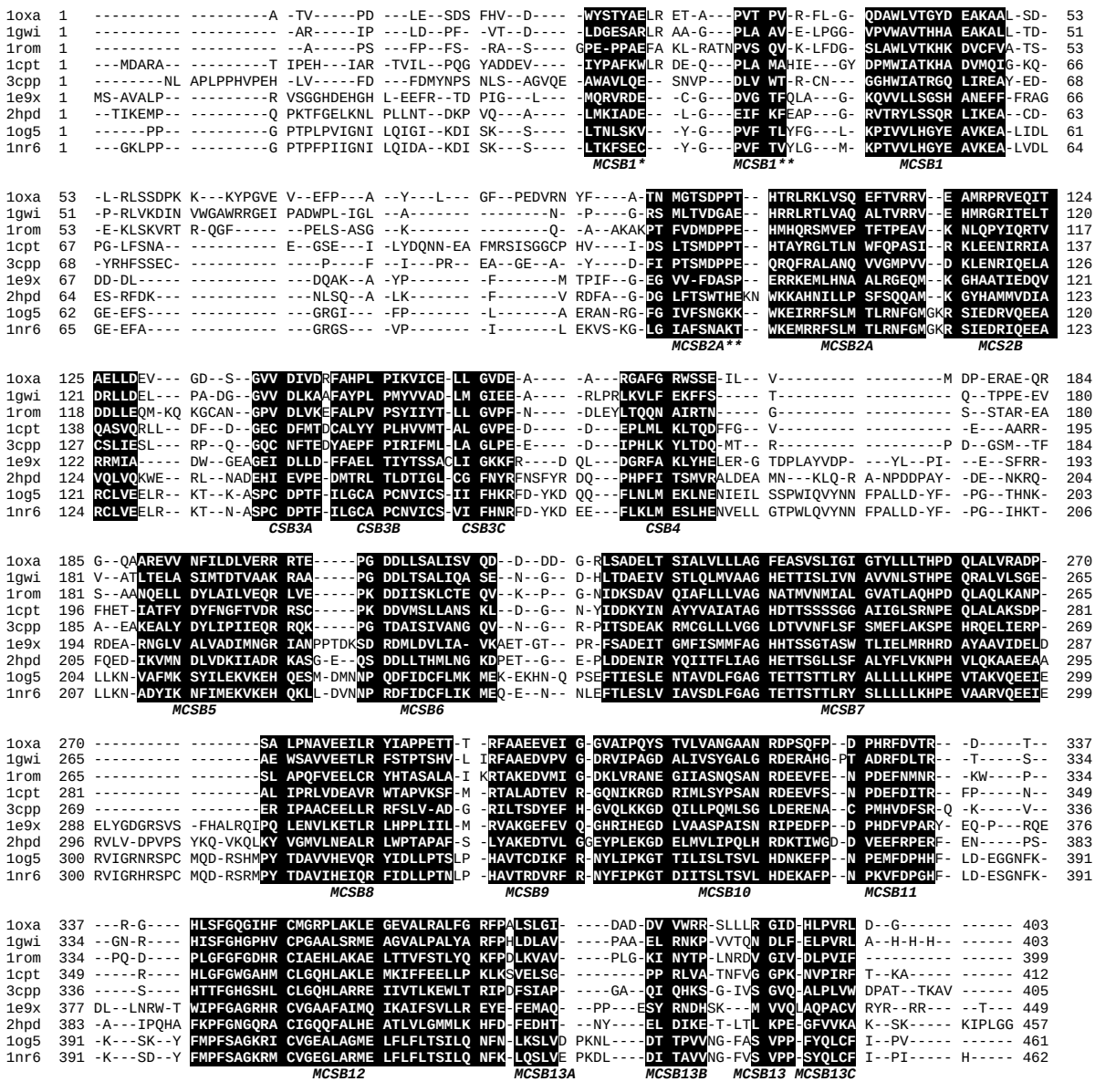


Figure 5-7 Identification des « blocs » conservés (MCSB) sur un alignement effectué sous Matras de 9 *templates* similaires à celui de la Figure 5-5 sans la structure du CYP 2C8 (1pq2) et du CYP 2C5 (1dt6) en raison de la limitation de nombre de structures à aligner sous Matras. La numérotation des MCSBs suit la nomenclature de P. Jean. Les structures retirées ont été choisies de façon à ne pas supprimer trop d'informations structurales : le CYP 2C9 (1og5) et le CYP 2C5 (1nr6) figurent encore dans l'alignement. Les MCSBs correspondent aux régions de l'alignement sans gap à l'exception du MCSB1*, où j'ai autorisé la présence d'un gap dans la structure du $P450_{nor}$ (1rom). Les régions identifiées sont situées non loin des CSBs identifiés par GOK, à quelques décalages près : elles portent en conséquence les mêmes désignations. Dans certains cas, il a fallu ajouter des numérotations, comme pour le MCSB3 qui est en trois « blocs » ici au lieu de un dans l'alignement sur la Figure 5-5.

Un exemple de cette identification de blocs est présenté à la Figure 5-7 où la méthode d'alignement utilisée est Matras. Les blocs identifiés sous Matras sont nommés MCSB et correspondent aux régions de l'alignement dépourvues de gap sur l'ensemble des templates utilisés.

Les *templates* utilisés sont les mêmes que ceux du jeu de blocs global à 11 *templates*. En raison de la limitation du serveur Matras à 10 *templates*[12], les structures 1dt6 (CYP 2C5) et 1pq2 (CYP 2C8) ont été écartées de l'alignement. Le choix s'est porté sur ces deux structures en raison de l'existence d'une autre structure de CYP 2C5 (1nr6) incluse dans l'alignement structural, ainsi que la présence d'une structure de CYP 2C9, très similaire à la structure du CYP 2C8.

En 2005, J. Mestres (Mestres, 2005) a publié un alignement séquentiel pour les P450s basés sur leurs informations structurales. L'information de similarité structurale a été évaluée cette fois-ci par une approche Gaussienne, implémentée par un programme maison d'alignement structural GAPS (pour Gaussian-based Alignment of Protein Structures) (Mestres, 2000). Dans ses travaux, J. Mestres a utilisé un total de 12 *templates* de P450s, dont 10 sont communs avec ceux du jeu global (à 11 *templates*) : cela facilite la comparaison avec notre alignement en comparant les « blocs » prélevés de son alignement multiple aux CSBs issus de GOK. Les *templates* utilisés par J. Mestres sont présentés au Tableau 5.6 et l'alignement est présenté en Figure 5-8.

Tableau 5.6 Liste des Cytochromes P450 utilisés par J. Mestres pour ses analyses (source Mestres, 2005).

CYP	*Classe*	*PDB ID*	*Résolution (Å)*	*Date*	*Source*
101 (cam)	I	1phc	1.60	31/10/1993	*Pseudomonas putida*
102 (bm3)	II	2bmh	2.00	31/07/1994	*Bascillus megaterium*
107 (eryF)	I	1oxa	2.10	07/12/1995	*Saccharopolyspora erythraea*
108 (terp)	I	1cpt	2.30	31/01/1994	*Pseudomonas* sp.
119	I	1io7	1.50	28/02/2001	*Sulfolobus solfataricus*
121	I	1n40	1.06	04/02/2003	*Mycobacterium tuberculosis*
51	I	1e9x	2.10	01/11/2000	*Mycobacterium tuberculosis*
55 (nor)	I	1rom	2.00	15/10/1997	*Fusarium oxysporum*
2C5	II	1dt6	3.00	27/09/2000	*Oryctolagus cuniculus*
2B4	II	1po5	1.60	07/10/2003	*Oryctolagus cuniculus*
2C8	II	1pq2	2.70	13/01/2004	*Homo sapiens*
2C9	II	1og2	2.60	17/07/2003	*Homo sapiens*

Note : Les classes utilisées ici, sont ceux de la nomenclature à 4 classes (cf. section 1.2.2.1, page 25)

[12] Seuls 9 *templates* sur les 10 autorisés par le serveur Matras ont été utilisés. Une place de réserve a été laissée à une dixième structure pouvant servir à « caler » l'alignement des *templates* sur la structure d'une séquence proche de la séquence cible, ou même, celle de la séquence cible, lorsqu'elle était disponible. Je reviendrai sur ce point ultérieurement.

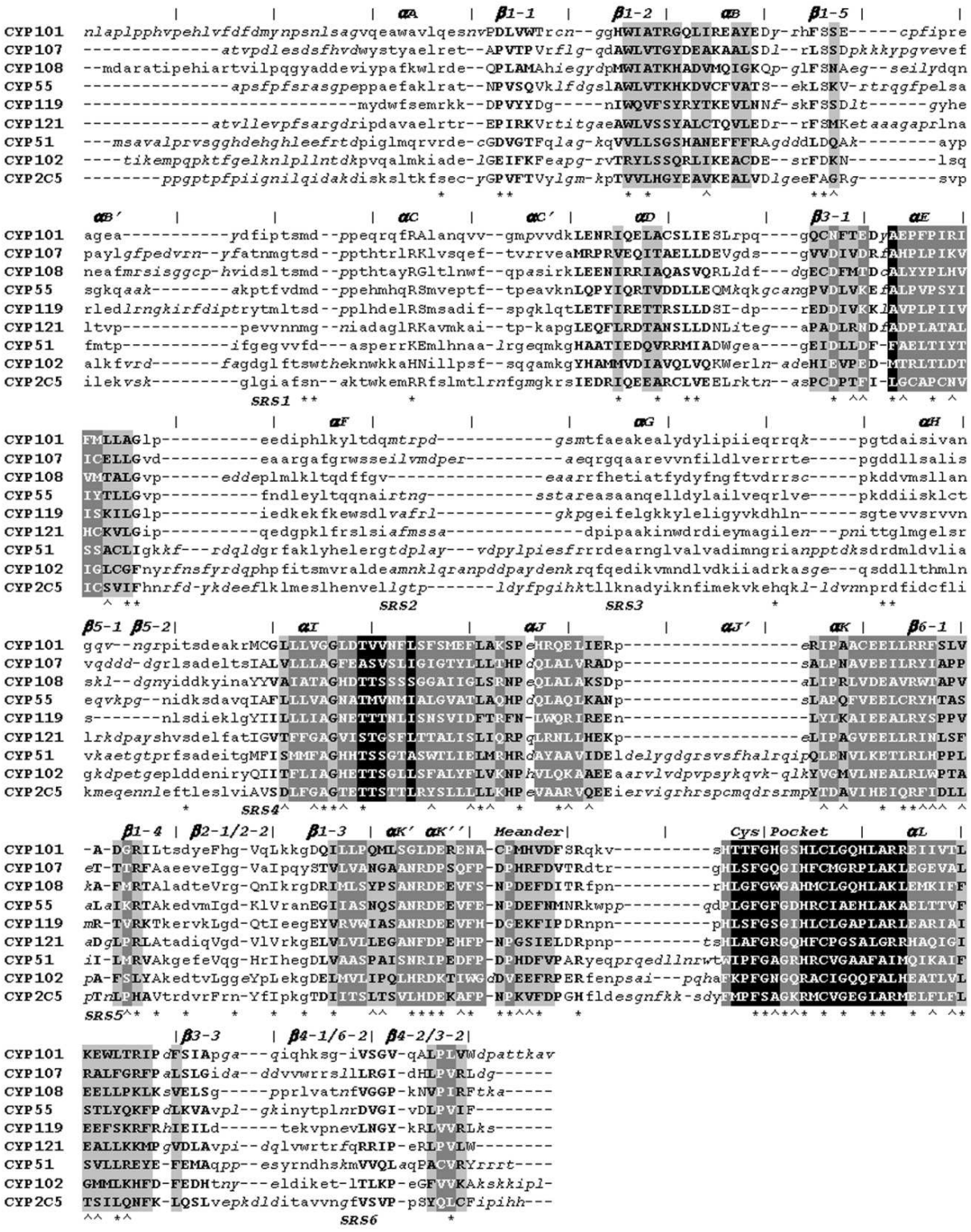


Figure 5-8 Alignement séquentiel de J. Mestres, basé sur des informations de structures de 9 P450s (source Mestres, 2005). Les noms des SSEs sont reportés au dessus des séquences et les SRS en dessous. Les surlignements correspondent à des valeurs de RMSD inférieures à 2 Å. Les caractères en minuscule correspondent à une valeur de RMSD > 3 Å, et en majuscule < 3 Å. Ainsi, les zones surlignées et en majuscule peuvent être assimilées à des blocs CSBs

5.2.5.2 Résultats de la comparaison avec les autres logiciels

Globalement, un nombre similaire de blocs sont identifiés, localisés pour la plupart aux mêmes endroits dans les séquences des *templates* (donc conduisant à des sous-alignements en séquences similaires à GOK). Les principales différences sont situées au niveau de la longueur des blocs et également dans le décalage d'une ou plusieurs séquences du sous-alignement. Aucune des méthodes citées dans le Tableau 5.5 ne donne un même alignement global, mais il existe en revanche une certaine tendance pour les sous-alignements des blocs : par exemple, les sous-alignements des blocs « centraux » (CSB3, CSB6 à CSB12) présentent peu de divergence d'un logiciel à un autre. En revanche, les blocs structuraux situés en N-terminal (CSB1* et CSB 1**), au niveau des hélices B' (CSB2A**), F (CSB4) et G (CSB5), et en C-terminal (série des 13) sont ceux qui font l'objet de plus d'interprétations divergentes au niveau du sous-alignement selon les méthodes utilisées.

On retrouve ces caractéristiques dans l'alignement de J. Mestres (cf. Figure 5-8). Dans ce dernier sont indiquées les régions à forte valeur de RMSD (séquence écrite en minuscule) et les régions à faible valeur de RMSD (séquence écrite en majuscule) qui correpondent donc à des régions structuralement conservées, comparables aux blocs identifiés sous GOK. Ainsi, la région correspondant au bloc CSB1* (hélice A) ne peut pas être assimilée à un bloc (RMSD > 3Å). En revanche, la région correspondant au bloc CSB1** est trouvée pour un RMSD de moins de 3Å dans le jeu de structures de J. Mestres, et peut donc être assimilée à un bloc, relativement court. La région comprenant les CSB2A (boucle B–C) n'est pas identifiable dans son alignement. Il en est de même pour les CSB correspondant aux hélices F, G et H ainsi que les régions C-terminales. On voit donc que les blocs qui posaient des problèmes de désaccord entre les différentes méthodes, ne sont pas assimilables à des blocs dans cet alignement. À noter que l'hélice H aussi n'est pas conservée dans cet alignement, comme dans le cas de GAKUSA, tandis que les autres méthodes parviennent à identifier cette hélice comme un bloc dont les sous-alignements sont assez en accord entre les différentes méthodes.

Encore une fois, les mêmes régions, sources de désaccord, posent problème : aucune des méthodes utilisées n'a donné le même sous-alignement pour ces régions. Pourtant, les éléments structuraux dans ces régions semblent conservés, puisque toutes les méthodes (même celle de J. Mestres) proposent un sous-alignement dépourvu de gap dans ces zones. Il est délicat ici de déterminer le meilleur alignement, dans la mesure où les alignements dépendent d'algorithmes différents.

En conclusion, mis à part GAKUSA, les logiciels d'alignement structuraux (présenté au Tableau 5.5) permettent d'identifier les mêmes CSBs que GOK, quel que soit le descriptif utilisé pour l'alignement de séquences. Avec environ 10 ans d'antériorité, GOK montre qu'il n'est pas contredit : la notion de CSBs (du moins chez les P450s) est donc maintenue et valide quelle que soit la méthode sollicitée. Ainsi, les logiciels disponibles aujourd'hui en ligne pourraient servir de méthodes alternatives à GOK dans la détection de CSBs, notamment pour valider un sous-alignement d'un CSB (dans le cas bien sûr où les blocs proposés sont en accord entre les méthodes, ce qui n'est pas le cas pour certaines régions des P450s).

5.3 *Caliseq* et les autres méthodes d'alignement

5.3.1 Obtention des alignements par *Caliseq* : rappel

Une fois les blocs structuraux définis, *Caliseq* réalise le positionnement des sous-alignements de ces blocs sur une séquence cible ou son pré-alignement avec des séquences homologues à fort taux d'identité. Il est d'ailleurs préférable d'utiliser un pré-alignement de séquences à une séquence cible unique[13] : en effet, même si les séquences de ce pré-alignement sont très semblables, une variabilité des résidus alignés à certaines positions peut exister. Cette variabilité est importante pour l'enrichissement des profils dans le programme *Caliseq*. Ainsi, *Caliseq* ne conduit pas toujours à un même positionnement des blocs selon qu'ils ont été alignés sur une séquence cible unique ou sur un pré-alignement de séquences. Il est d'ailleurs intéressant de constater que les blocs dont le positionnement diffère selon l'utilisation d'une séquence cible unique ou d'un pré-alignement, correspondent à certains blocs précédemment cités qui étaient source de désaccord entre les différentes méthodes d'alignement structural. Cette remarque souligne le caractère « flottant » de ces blocs.

L'étape de positionnement des blocs n'est en réalité qu'une étape préalable à l'obtention d'un alignement multiple final, qui servira de point de départ pour la reconstruction du modèle. En effet, le positionnement des blocs sur la séquence cible n'est qu'un alignement partiel : toute la partie inter-bloc doit être complétée par la suite, et ce, sans information spécifique. Selon l'importance que l'utilisateur portera sur ces régions inter-blocs, cette manipulation peut s'avérer délicate. D'ailleurs, c'est la principale contrainte de la modélisation par blocs.

[13] Sauf lors de la recherche de P450s dans les banques de séquences où les blocs sont positionnés sur une seule séquence, permettant ainsi de déterminer (selon le score d'alignement) si elle est reconnue ou non comme un P450. Voir la section 4.2.3.

5.3.2 Les différentes méthodes d'alignements pour la reconstruction 3D

5.3.2.1 Nomenclature

En vue de valider les alignements produits par *Caliseq*, on se propose de les évaluer par rapport à des alignements produits par d'autres logiciels académiques, pouvant servir eux aussi à la reconstruction de modèles de CYP. Pour des raisons de commodité d'explication, il convient d'établir une nomenclature pour tous les jeux de *templates*, qui sera utilisée jusqu'à la fin de ce chapitre. Chaque jeu diffère par la méthode d'alignement utilisée, méthodes qui seront présentés dans la section suivante. La nomenclature est présentée dans le Tableau 5.7.

Tableau 5.7 Nomenclature pour désigner les jeux de *templates* utilisés pour construire les modèles. Pour certains logiciels (Matras ou HHpred) il n'a pas été possible de reprendre exactement les *templates* de référence de la thèse (A,B,C ou D) : un sous ensemble a été alors utilisé ou même un nouveau jeu (jeu ***G*** pour HHpred)

Jeu	*Correspondance*	*Méthode d'alignement*	*Structures (ref)*
A	*6 Templates* de N. Loiseau	GOK (3D) / Clustalw (1D)	$P450_{eryF}$, $P450_{BM3}$, $P450_{cam}$, $P450_{nor}$, $P450_{terp}$, CYP 2C5
B	*4 Templates* de M. Cottevieille	GOK (3D) / Clustalw (1D)	$P450_{eryF}$, $P450_{BM3}$, CYP 51, CYP 2C5
C	*6 Templates* TA. NGUYEN	GOK (3D) / Clustalw (1D)	$P450_{eryF}$, $P450_{BM3}$, CYP 154C1, CYP 2C5, CYP 2C8, CYP 2C9
C'	*6 Templates* TA. NGUYEN	Matras (3D)	Jeu C
D	*11 Templates* communs de A,B et C	GOK (3D)	Jeu commun de A∪B∪C
D'	*9 Templates* (sous ensemble du jeu D)	Matras (3D)	Jeu D – {CYP 2C8}
E	28 *Templates* représentatifs des P450s	GAKUSA (3D)	$P450_{eryF}$, $P450_{BM3}$, $P450_{cam}$, $P450_{nor}$, $P450_{terp}$, $P450_{OxyC}$, $P450_{epok}$, $P450_{st}$, $P450_{PIKC}$, CYP 51, CYP 119, CYP 121, CYP 152A1, CYP 154A1, CYP 154C1, CYP 158A2, CYP 175A1, CYP 199A2, CYP 2A6, CYP 2B4, CYP 2C5, CYP 2C8, CYP 2C9, CYP 2D6, CYP 3A4, CYP 8A1, CYP 2R1, CYP 1A2
F	*4 Templates* (sous ensemble du jeu C)	HHpred (HMM)	Jeu C – {CYP 2C5, CYP 2C8}
G	*3 Templates* (sous ensemble du jeu B)	Hhpred (HMM)	Jeu B – {CYP 2C5}
H	*10 Templates* (variante de F∪G)	Hhpred (HMM)	CYP 2B4, CYP 2C9, CYP 51, CYP 154C1, CYP 154 A1, CYP 175A1, $P450_{BS}$, $P450_{BM3}$, $P450_{eryF}$, $P450_{terp}$

Note 1 : Le jeu D' correspond au 11 *templates* du jeu D privé de la structure du 1pq2 (CYP 2C8) et 1dt6 (CYP 2C5) en raison de la limitation des *templates* utilisables par le serveur pour produire l'alignement. Un $10^{ème}$ *template* peut être utilisé pour permettre l'alignement sur la séquence cible (le *template* pouvant être soit la structure d'une séquence proche de la séquence cible, soit le structure de la séquence cible, lorsqu'elle est disponible)

Note 2 : Le jeu E est différent de celui présenté au Tableau 3.2, page 141. La structure du $P450_{OxyB}$ a été retirée car elle comportait trop de régions non résolues.

Note 3 : Le jeu H est une variante du jeu commun F et G. Il correspond à l'union des *templates* de F et G et comprend également 6 autres structures proposées par HHpred. Ce jeu H offrait la possibilité d'avoir les alignements avec tous les *templates* de l'époque (2005). Il s'agit du cas où le CYP 3A4 a été soumis en requête.

5.3.2.2 Présentation des méthodes

Nombreuses sont les méthodes qui proposent un alignement de séquences permettant la reconstruction d'un modèle, comme il a été vu dans le chapitre 2 au paragraphe 2.4. Une comparaison exhaustive de ces méthodes n'est pas l'objet de cette thèse. Nous avons choisi les plus représentatives du moment pour pouvoir évaluer l'alignement obtenu à la suite du positionnement des blocs par

Caliseq, en le comparant à ceux obtenus avec les autres méthodes. Parmi celles que nous avons retenues, il y a les méthodes simples d'alignement de séquences comme Clustalw, des méthodes plus complexes basées sur les HMM comme HHpred, et des méthodes d'alignements basées sur l'information structurale, comme Matras. Pour tous les logiciels d'alignement, les *templates* de nos jeux de blocs ont été utilisés à chaque fois que cela était possible pour comparer l'alignement par *Caliseq* aux alignements effectués par les autres méthodes. Pour certains logiciels en revanche (HHpred ou Matras), il n'a pas été possible d'utiliser la totalité des *templates* d'un jeu donné (jeu ***D'***,***F***, ou ***G***) : à la place, un sous ensemble a été utilisé (respectivement, ***D***, ***C***, ou ***B***), et parfois même un jeu original a été créé (jeu ***G***).

Le logiciel **Clustalw** (cf. section 2.4.2.2 sur les méthodes heuristique, à la page 92), réalise un alignement basé exclusivement sur les informations de séquences primaires des *templates*. Sa grande simplicité et son accessibilité se révèlent suffisants et adaptés pour des analyses d'évolution et de construction d'arbres phylogénétiques.

Matras (section 2.4.3.3) correspond à une méthode d'alignement structural (à descriptif SSE) qui fournit en résultat un alignement de séquences. L'inconvénient lors de l'utilisation de Matras est d'une part sa limitation du nombre de structures *templates* utilisables pour l'alignement (10 *templates* au maximum), et d'autre part l'impossibilité d'incorporer dans l'alignement la séquence cible du P450 à reconstruire lorsque sa structure est inconnue. En effet, Matras utilise les structures disponibles dans la PDB pour réaliser l'alignement structural, il est donc normal que la structure à modéliser n'en fasse pas partie. Pour contourner cette contrainte, nous avons exploité l'alignement de Matras de deux manières : i) lorsque la structure de la séquence cible (ou la structure d'une séquence vraiment proche, à plus de 70% d'identité) était disponible, cette dernière était incorporée aux jeux ***C'*** ou ***D'***, au niveau du 10ème *template* autorisé par le serveur[14] et ii) lorsqu'aucune structure pouvant aider l'alignement sur la séquence cible n'était disponible, des blocs (les MCSBs) ont été prélevés de l'alignement structural de Matras des 9 *templates* et soumis à *Caliseq*. Cette deuxième manière « hybride » permet non seulement de contourner le problème de réalignement sur la séquence cible de l'alignement structural Matras, mais présente aussi l'avantage par rapport aux blocs provenant de GOK, d'avoir tout de suite à disposition les alignements inter-blocs entre les *templates*.

[14] Dans ce premier cas d'utilisation de Matras, l'alignement produit par Matras sur la séquence cible est largement favorisé, puisque que la structure de cette dernière a été alignée structuralement avec les autres *templates*

Le serveur **HHpred** (http://protevo.eb.tuebingen.mpg.de/hhpred) (Söding et *al.*, 2005) permet d'obtenir un alignement de séquences à la suite de la soumission de la séquence cible. La méthode d'alignement utilisée par ce seveur repose sur les HHMs. Dans cette méthode, les premières étapes correspondent à celles d'un PSI-BLAST où les SSEs, annotés par PSIPRED à la dernière itération sont pris en compte pour la construction d'un profil HMM. Ce profil HMM est alors confronté à ceux pré-calculés d'une base, contenant également des informations de SSE soit prédites par PSIPRED soit déterminées par DSSP, et permet d'identifier les séquences qui sont les plus proches de la séquence cible requête. Après sélection des structures de références, l'alignement peut être proposé sous forme de fichier d'entrée au programme Modeller. Il est à noter que les *templates* utilisés pour la recherche de blocs structuraux sous GOK, ne sont pas forcément les premiers à sortir lors de la recherche par HHpred : elle n'est basée que sur la similarité de la séquence, contrairement à nos *templates*, choisis en fonction des ligands communs reconnus, et de la forme du site actif. Dans certains cas, nos *templates* n'apparaissent pas dans les structures proposées par HHpred (la recherche a été effectuée sur la PDB70, construite sur des structures partageant au moins 70% d'identité en séquence). Il a donc fallu dans ces cas soit choisir une structure du même CYP mais sous un nom pdb différent (par exemple pour le CYP 2C9, utiliser le 1ro9 à la place du 1og5, la seule différence résidant dans le substrat co-cristallisé), soit avoir à se passer des *templates* non trouvés par HHpred. Ainsi, les jeux ***F*** et ***G*** sont dérivés des jeux ***B*** et ***C*** respectivement (à un ou deux *templates* près, cf. Tableau 5.7) pour rester au plus près des conditions expérimentales de GOK. Il est d'ailleurs surprenant qu'HHpred ne propose pas la structure du CYP 2C5 ni celle du CYP 2C8 pourtant proches de celle du CYP 2C9 quelles que soient les requêtes. Le jeu ***H***, lui, est proposé par HHpred en réponse à une requête de CYP 3A4 et englobe non seulement les structures des jeux ***F*** et ***G***, mais également un certain nombre de nouvelles structures d'origine bactérienne qui n'étaient pas disponibles lorsque nous avons travaillé avec les jeux ***A***, ***B*** et ***C***. Parmi les 10 structures du jeu ***H***, 6 sont communes avec le jeu ***D*** : il s'agit des CYP 2C9, CYP 154C1, CYP 51, $P450_{BM3}$, $P450_{terp}$ et $P450_{eryF}$. Ces *templates* identifiés comme proches de la séquence du CYP 3A4 confirme le choix des *templates* utilisés dans les jeux ***A***, ***B***, ***C*** et par conséquent ***D***. Outre l'absence des structures du CYP 2C5 et CYP 2C8, les $P450_{cam}$ et $P450_{nor}$ ne sont pas non plus proposés par HHpred. Ces dernières structures ($P450_{cam}$ et $P450_{nor}$) existaient dans le jeu historique ***A***. Ces structures pourraient donc, si on en croit HHpred, être une source de divergence pour tous les alignements qui les comportent. Une dernière remarque concerne la structure du CYP 2B4 : il s'agit de la forme ouverte 1po5, qui avait été exclue du jeu ***C*** lors de la recherche sur GOK, car trop divergente. Elle se retrouve pourtant dans le jeu ***H*** proposé par HHpred. La présence de cette structure ne devrait-elle pas compromettre la reconstruction du CYP 3A4, en dépit d'un bon alignement multiple ?

5.3.3 Comparaison des différents alignements

Évaluer une méthode d'alignement et déterminer si elle est meilleure qu'une autre n'est pas une tâche aisée. Il faut pour cela disposer d'un critère fiable, permettant la comparaison des alignements entre eux. On peut considérer qu'une fois les structures connues, le bon alignement correspond à celui qui relie au mieux la séquence à la structure quand celle-ci est connue. Ce critère doit être néanmoins utilisé avec précaution : il a été vu précédemment que les logiciels d'alignement structuraux ne fournissent pas toujours des résultats d'alignement en accord. Un moyen d'évaluer et de comparer les méthodes de reconstruction consiste donc à reconstruire des modèles d'une structure connue qui servira de référence pour la comparaison avec les modèles générés. Cette évaluation est donc postérieure à la reconstruction. Comme cette évaluation fait l'objet d'une section ultérieure, **la comparaison des alignements dans cette section sera purement descriptive**. Les alignements sont sensibles à plusieurs paramètres : l'algorithme de la méthode et le choix des *templates* utilisés pour l'alignement. Ainsi, un alignement différent peut être obtenu en ne retirant qu'une seule séquence du jeu de *templates* à aligner sur la séquence cible. Différentes séquences cibles à reconstruire ont été utilisées pour la comparaison : CYP 3A4, CYP 2C9, CYP 119, $P450_{cam}$ et CYP 1A2 (toutes sont présentes dans la PDB). Ce sont les alignements sur la structure la plus emblématique de la superfamille, le CYP 3A4 qui seront comparés dans cette section.

5.3.3.1 Caliseq : comparaison des alignements issus des différents jeux de templates

Examinons d'abord les alignements obtenus par *Caliseq* avec des jeux de *templates* différents. Ces alignements dépendent fortement du positionnement des blocs sur la séquence cible. La première comparaison a donc été effectuée sur les trois jeux de blocs ***A***, ***B*** et ***C*** (cf. résultats des alignements à la section 5.2.2.4). Cette comparaison est facilitée par le fait que chacun de ces jeux comporte au moins deux *templates* en commun avec les autres jeux de blocs.

La Figure 5-3 de la page 192, présente un exemple de résultat de positionnement des trois jeux de blocs par *Caliseq* sur la séquence cible CYP 3A4. Les sous-alignements, montrant déjà des divergences, ont résulté en un désaccord sur certaines positions de blocs sur la séquence cible, lequel entraîne un désaccord pour certaines régions inter-blocs. Les alignements séquentiels finaux diffèrent donc par suite d'un enchaînement : au premier niveau, des décalages présents (selon le jeu choisi A, B ou C) dans les sous-alignements lors de l'identification des blocs par GOK, entraînent au second niveau des désaccords de positionnement de ces sous-alignements sur la séquence cible, eux-mêmes se répercutant sur l'alignement inter-bloc.

Pour les premiers jeux historiques de *templates* (***A***, ***B*** et ***C***), *Caliseq* conduit à des alignements légèrement différents (en comparant les *templates* communs aux trois jeux avec la séquence cible). À noter que pour les jeux ***A*** et ***B***, il faut tronquer la séquence cible de 20 à 30 résidus en N-terminal lorsque celle-ci est microsomale, afin d'améliorer le positionnement des blocs dans cette région. Les régions conservées (au sens « d'absence de gap ») sont principalement observées au niveau des blocs. Parmi elles, celles qui sont localisées au niveau des blocs « fiables » correspondent à des zones fixes, retrouvées toujours aux mêmes endroits de la séquence cible sur les alignements des trois jeux. Il existe également des régions conservées qui sont décalées d'un alignement à l'autre : ces régions correspondent aux blocs « flous » sur lesquels aucune autre méthode d'alignement (ni Matras, ni Clustalw) n'a été en mesure de trancher. Concernant les régions variables, elles sont généralement localisées au niveau des zones inter-blocs. Différentes méthodes ont été essayées pour optimiser les alignements à ces endroits (cf. section 3.3.3.2) sans pour autant donner de résultat satisfaisant : elles ont été finalement alignées à la main, Clustalw n'étant pas en mesure de les réaliser sur des fragments isolés et trop divergents.

Lorsque le jeu de blocs global (jeu ***D***) a été conçu, un alignement nouveau a été observé par rapport à ceux obtenus à partir des jeux ***A***, ***B*** et ***C***. De manière similaire, il existe des régions en accord avec au moins deux des autres alignements. Comme toujours, ces régions « fixes » correspondent aux blocs « fiables ». Concernant la partie inter-bloc, les différences sont moindres, mais cela résulte d'un travail d'alignement manuel. Bien qu'étant nouveau, l'alignement à partir du jeu ***D*** est pourtant celui qui présente le moins de différences avec chacun des trois autres alignements pris séparément, ce qui est rassurant dans la mesure où les blocs de ce jeu global de *templates* résultent d'un compromis avec ceux des autres jeux.

Pour la méthode « hybride » (jeu ***C'*** ou ***D'***) où les blocs ont été prélevés dans l'alignement structural fourni par Matras, il existe déjà une différence dans les sous-alignements des blocs par rapport à ceux identifiés par GOK (jeu ***C*** et ***D***), qui se répercute ensuite sur tout l'alignement. Au final, on n'obtient pas un même alignement que ceux produits à partir des jeux ***C*** et ***D***. Néanmoins des régions fixes de l'alignement sont toujours observables, localisées aux mêmes endroits de la séquence cible que celles des alignements précédemment évoqués. Comme toujours, elles correspondent à des régions à haute conservation structurale (bloc « fiable »).

L'alignement des 28 *templates* (jeu ***E***) sur la séquence cible du CYP3 A4 n'a pas été effectué car la structure du CYP 3A4 figure parmi ces 27 *templates*. En revanche, le positionnement des blocs a été

réalisé, et semble correct : dans les sous-alignements des blocs, les « morceaux » de séquence correspondant à la structure du 1tqn (CYP 3A4) s'aligne parfaitement avec la séquence cible du CYP 3A4. Ce même constat de positionnement a été observé lorsque j'ai aligné sur une séquence cible de P450 bactérienne, le CYP 119, dont le *template* correspondant est le 1io7 et également sur une autre séquence cible de P450 microsomal, le CYP 1A2, dont le *template* est le 2hi4.

Pour les jeux de *templates* ***A*** à ***E*** alignés par *Caliseq*, on retiendra que des régions conservées fixes par rapport à la séquence cible sont observées sur tous les alignements, quel que soit le jeu utilisé (sauf le jeu ***E*** qui n'a pas été testé et qui comporte de surcroît des blocs plus longs). Ces régions correspondent systématiquement aux blocs « fiables », déjà discutés.

5.3.3.2 Alignements obtenus par HHpred

Le serveur HHpred offre l'avantage de proposer un alignement de séquences comprenant la séquence requête. Cette séquence requête correspond à la séquence cible dont le modèle 3D est à reconstruire. En revanche, les alignements obtenus par ce serveur présentent beaucoup de différences, ne serait-ce qu'entre eux, avec des jeux de *templates* différents (jeux ***F***, ***G*** et ***H***, cf. Tableau 5.7 page 206). Les alignements par HHpred sont étonnamment « aérés », résultant d'un nombre important d'insertions de gaps au niveau des régions variables. Les mêmes régions conservées fixes sont trouvées dans les alignements produits par HHpred. Elles sont en revanche de taille plus courte. Comme toujours, les régions de l'alignement correspondant aux SSEs de la boucle B–C, dans les régions des hélices F et H et dans le feuillet en C-terminal correspondent à des régions mal alignées. En revanche, la partie correspondante à l'hélice G ne pose pas de problème dans l'alignement, sans aucun gap sur toutes les séquences de l'alignement (aussi bien sur l'alignement des jeux ***F*** et ***H*** montrés en annexe que sur celui du le jeu ***G***). L'apport des structures bactériennes dans le jeu ***H*** n'apportent rien à l'alignement d'HHpred : c'est comme si les alignements des jeux ***F*** et ***G*** n'étaient que des sous-alignements de celui du jeu ***H***. C'est peut être ainsi d'ailleurs que les alignements sont obtenus : un alignement global est réalisé, donnant une multitude de sous alignements.

En changeant la séquence cible en requête, d'autres séquences *templates* sont proposées, conduisant alors à un alignement inévitablement différent. Au final, il n'est pas réellement évident d'effectuer des comparaisons d'alignement avec le logiciel HHpred, dans la mesure où les *templates* utilisés changent tout le temps en fonction de la séquence requête (le logiciel cherche les *templates* d'identité la plus proche de la séquence cible). Ces autres alignements réalisés à partir d'une séquence cible différente (CYP 119, CYP 2C9, $P450_{cam}$ ou CYP 1A2) semblent néanmoins présenter des régions

conservées fixes (par rapport aux jeux ***F***, ***H*** et ***G***), qui sont placées aux mêmes endroits selon la séquence requête demandée.

On retiendra donc pour les alignements d'HHpred que les régions conservées fixes sont les mêmes que celles précédemment repérées sur les alignements par *Caliseq* à partir des jeux ***A***, ***B***, ***C***, ***D***, ***C'*** et ***D'***. En changeant de séquence cible, les régions conservées « fixes » ne sont pas trop perturbées. Par extrapolation, il est possible de penser que ces régions conservées fixes sont localisées sur tous les P450s aux mêmes endroits. Comme ces régions conservées fixes coïncident avec les positionnements des blocs « fiables », nous avons là un nouvel argument en faveur de l'utilisation des CSBs.

5.3.3.3 Alignements obtenus par Clustalw

L'alignement par Clustalw ne dépend que de la séquence primaire, aussi subit-il d'énormes modifications selon les *templates* utilisés. Les alignements par Clustalw furent historiquement les premiers testés pour comparer à ceux obtenus soit par SmartConsAlign, soit par *Caliseq*. Dans le cas des jeux ***A***, ***B*** ou ***C***, un faible nombre de *templates* était proposé à Clustalw pour l'alignement, dont l'identité en séquence était extrêmement basse pour envisager une reconstruction par homologie : nous n'avions pas réellement de choix sur ces *templates*, car il s'agissait des seules structures alors disponibles dans la PDB. En raison des divergences en séquence des *templates*, les alignements présentaient alors de nombreux mésappariements et de désaccords entre les alignements des jeux ***A***, ***B*** et ***C***. En effet, contrairement aux autres méthodes, Clustalw semble préférer les mésappariements à l'introduction de gaps, même en diminuant le poids alloué aux ouvertures et aux extensions de gap dans le programme. En revanche, en combinant certains paramètres de poids et en utilisant la matrice de similarité BLOSUM30 (pour les séquences à moins de 30% d'identité), des alignements plus proches de ceux obtenus par *Caliseq* sont observés, avec la présence de régions conservées plus courtes. Comme pour *Caliseq*, cet alignement est très variable en N-terminal et C-terminal. L'alignement peut être amélioré en supprimant les quelques premiers résidus [15] N-terminaux, conduisant à une diminution des mésappariements en N-terminal lorsque peu de structures *templates* sont disponibles pour l'alignement (et surtout lorsqu'elles sont majoritairement d'origine bactérienne). Comme toujours, on note la présence de régions conservées (dépourvues de gap) au niveau central, lorsque les bons paramètres ont été ajustés pour l'alignement, mieux positionnées par rapport à ceux situés aux extrémités, et en accord avec celles observées dans les alignements des autres méthodes. Il

[15] Cette manipulation pour améliorer l'alignement n'a été possible qu'une fois en possession des structures de la séquence cible : l'alignement structural fournissait alors des informations sur l'alignement en N-terminale. Sans ce dernier, dans le cas d'un cytochrome P450 inconnu, il n'est pas aisé de décider si les premiers résidus doivent être supprimés ou non (tout dépend du nombre de *templates* d'origine microsomale dans les jeux d'alignement).

semblerait donc, en dépit d'une certaine variabilité, que les régions centrales, correspondant à l'hélice I et J, soient aussi bien conservées structuralement que séquentiellement, comme il a été remarqué sur les alignements des autres méthodes mais également celui fourni par Clustalw.

Avec les *templates* du jeu global (jeu ***D***), Clustalw produit des alignements proches de ceux des autres méthodes (aussi bien *Caliseq*, que des logiciels d'alignement structural) sans avoir à ajuster les paramètres de Clustalw. L'alignement par Clustalw des 11 *templates* du jeu ***D*** sur la séquence cible est nettement plus en accord avec ceux des autres méthodes que ne le sont les alignements des 4 ou 6 *templates* des jeux ***A***, ou ***B*** et ***C*** respectivement. Il existe toujours des désaccords dans les appariements en N- et C-terminal et dans certaines régions correspondant aux SSE des hélices B', F et G.

Enfin, avec un jeu de tous les *templates* disponibles de P450s (jeu ***E***), un alignement étonnamment « propre » est observable, en accord avec les alignements structuraux et permettant de bien séparer les séquences de *templates* d'origine bactérienne des séquences de *templates* d'origine microsomale. Cet alignement (jeu ***E***) est légèrement différent de ceux qu'on pourrait obtenir par *Caliseq*, mais plus en accord avec ceux des autres méthodes comme Matras ou HHpred par exemple. Un tel résultat est assez choquant, et remet en question le travail d'alignement par *Caliseq* : à quoi bon développer une méthode aussi complexe lorsqu'un simple alignement par Clustalw fournit un alignement plus proche de ceux obtenus par des logiciels d'alignement structuraux ? Une hypothèse a été formulée : la qualité de cet alignement fourni par Clustalw résulte en fait du nombre important de *templates* utilisés (actuellement disponibles dans la PDB). Ce nombre a augmenté rapidement ces cinq dernières années (cf. Figure 1-14, page 39). L'alignement Clustalw remet en question l'intérêt de l'outil spécifique pour l'alignement à bas taux d'identité.

Afin de confirmer cette hypothèse, la manipulation suivante a été opérée : plusieurs alignements ont été réalisés itérativement sous Clustalw (dans son fonctionnement par défaut), avec le jeu de 28 *templates*, où des séquences sont chaque fois retirées de façon aléatoire. Après quelques itérations, le nombre de séquences retirées est augmenté, et le processus d'alignement itératif, relancé. J'ai procédé ainsi jusqu'à observation d'un changement radical dans l'alignement. L'idée de cette manipulation est de connaître le nombre minimal de séquences nécessaire pour l'obtention d'un alignement de séquences proche de l'alignement structural. Comme je l'ai fait remarquer précédemment, pour chaque suppression de séquences de *templates*, un léger changement dans l'alignement est observé. Celui-ci est d'ailleurs plus important lorsque la séquence retirée correspond à certaines structures de

P450 d'origine bactérienne. Il semblerait selon cette observation que c'est la présence de structures issues de mammifère qui imposeraient une certaine pression sur l'alignement. Néanmoins, au fur et à mesure que des *templates* sont retirés, il est possible d'observer l'évolution des alignements par Clustalw. Il fut très difficile de délimiter un seuil minimal, sachant que la comparaison d'alignement est relativement difficile : à quel moment dit-on qu'un alignement est différent du précédent ? J'ai dû faire le choix de considérer que l'alignement devenait réellement différent lorsque les régions censées conservées n'étaient plus homogènes avec celles observées sur les alignements structuraux. Selon ce critère, les alignements des P450s ne semblent pas fiables à moins de huit *templates*. C'est d'ailleurs pour cette raison qu'avec le jeu global (jeu ***D***) aucun paramétrage de Clustalw n'a été nécessaire pour proposer un alignement relativement proche de ceux des autres méthodes.

5.3.3.4 Alignements obtenus par Matras

Pour Matras, deux approches ont été testées. Pour la première approche, l'alignement structural comporte non seulement les *templates*, mais aussi la structure de la séquence cible à reconstruire. En procédant ainsi, il semble évident que l'alignement obtenu est largement favorisé par rapport aux autres méthodes dans la mesure où Matras utilise l'information structurale. Dans l'alignement proposé par Matras sur la structure 1tqn du CYP 3A4, environ 80% de l'alignement correspondent à des régions conservées, à savoir des régions dépourvues de gap. En revanche, dans les régions variables (en N-terminal principalement) un nombre important de gaps est observé dans l'alignement. À noter que la suppression ou le remplacement d'un *template* par un autre n'entraînent qu'une légère modification, principalement sur la taille des régions conservées : ainsi, contrairement aux alignements séquentiels, les alignements structuraux semblent mieux préservés (du moins avec cette méthode pour la famille des P450s) reflétant la caractéristique de repliement commun de cet enzyme. Les différences entre les alignements obtenus de cette première approche par Matras et ceux par *Caliseq* sont les mêmes que ceux déjà décrits pour la différence entre les blocs (cf. section 5.2.5.2).

La première méthode n'est pas applicable lorsque la séquence à modéliser est de structure inconnue. Lorsqu'aucun *template* n'est proche en identité de la séquence à modéliser, il s'avère difficile d'exploiter l'alignement des *templates* obtenus par Matras. C'est ainsi que j'ai eu l'idée d'utiliser les régions conservées dans l'alignement de Matras comme des blocs, que j'ai positionnés sur la séquence cible à l'aide de *Caliseq*. Il s'agit donc d'une méthode « hybride » utilisant l'information structurale fournie par Matras et l'alignement par profils réalisé par *Caliseq*. Une expérience intéressante a consisté justement à comparer l'alignement obtenu par la méthode « hybride » avec celui obtenu uniquement par Matras, lorsque la structure de la séquence cible est

disponible (pour le cas du CYP 3A4 par exemple, j'ai pris la structure d'1tqn). Pour cela, après avoir aligné les structures *templates* (jeu ***D'***) avec la structure du CYP 3A4 (1tqn), les « blocs d'alignement » des régions conservées ont été récupérés pour être positionnés par *Caliseq*, en supprimant préalablement les séquences du 1tqn de tous les sous-alignements. Les différences de positionnements des « MCSB » par les deux logiciels sont montrées en Figure 5-9.

Les différences de positionnements des MCSBs ne sont pas nombreuses. En considérant que Matras fourni un alignement structural fiable, sur les 18 MCSBs *Caliseq* ne s'est trompé que sur 4 : deux d'entres eux sont en C-terminal et ne correspondent pas à des décalages significatifs, en revanche, les deux autres (MCSB2A* et MCSB4) ne sont pas du tout bien positionnés. À noter que les blocs en N-terminal n'ont pas été identifiés par l'alignement de Matras comme régions conservées. Du fait des décalages sur le positionnement de ces blocs initiaux, il y a donc également quelques décalages dans l'alignement global, principalement aux niveaux des régions inter-blocs. Enfin, comme les MCSBs ne correspondent pas tout à fait aux CSBs identifiés par GOK, au niveau des sous-alignements, l'alignement par la méthode « hybride » diffère légèrement de ceux obtenus par *Caliseq* avec le jeu ***D***.

```
CYP3A4(c)   1   MALIPDLAME TWLLLAVSLV LLYLYGTHSH GLFKKLGIPG PTPLPFLGNI LSYHKGFCMF DMECHKKYGK VWGFYDGQQP  80
CYP3A4(m)   1   MALIPDLAME TWLLLAVSLV LLYLYGTHSH GLFKKLGIPG PTPLPFLGNI LSYHKGFCMF DMECHKKYGK VWGFYDGQQP  80
                                                                          MCSB1*       MCSB1**

CYP3A4(c)  81  VLAITDPDMI KTVLVKECYS VFTNRRPFGP VGFMKSAISI AEDEEWKRLR SLLSPTFTSG KLKEMVPIIA QYGDVLVRNL  160
CYP3A4(m)  81  VLAITDPDMI KTVLVKECYS VFTNRRPFGP VGFMKSAISI AEDEEWKRLR SLLSPTFTSG KLKEMVPIIA QYGDVLVRNL  160
                 MCSB1                MCSB2A*                MCSB2A                MCSB2B

CYP3A4(c) 161 RREAETGKPV TLKDVFGAYS MDVITSTSFG VNIDSLNNPQ DPFVENTKKL LRFDFLDPFF LSITVFPFLI PILEVLNICV  240
CYP3A4(m) 161 RREAETGKPV TLKDVFGAYS MDVITSTSFG VNIDSLNNPQ DPFVENTKKL LRFDFLDPFF LSITVFPFLI PILEVLNICV  240
                    MCSB3A   MCSB3B     MCSB3C                  MCSB4

CYP3A4(c) 241 FPREVTNFLR KSVKRMKESR LEDTQKHRVD FLQLMIDSQN SKETESHKAL SDLELVAQSI IFIFAGYETT SSVLSFIMYE  320
CYP3A4(m) 241 FPREVTNFLR KSVKRMKESR LEDTQKHRVD FLQLMIDSQN SKETESHKAL SDLELVAQSI IFIFAGYETT SSVLSFIMYE  320
                       MCSB5              MCSB6                              MCSB7

CYP3A4(c) 321 LATHPDVQQK LQEEIDAVLP NKAPPTYDTV LQMEYLDMVV NETLRLFPIA MRLERVCKKD VEINGMFIPK GWVVMIPSYA  400
CYP3A4(m) 321 LATHPDVQQK LQEEIDAVLP NKAPPTYDTV LQMEYLDMVV NETLRLFPIA MRLERVCKKD VEINGMFIPK GWVVMIPSYA  400
                                              MCSB8        MCSB9      MCSB10

CYP3A4(c) 401 LHRDPKYWTE PEKFLPERFS KKNKDNIDPY IYTPFGSGPR NCIGMRFALM NMKLALIRVL QNFSFKPCKE TQIPLKLSLG  480
CYP3A4(m) 401 LHRDPKYWTE PEKFLPERFS KKNKDNIDPY IYTPFGSGPR NCIGMRFALM NMKLALIRVL QNFSFKPCKE TQIPLKLSLG  480
                     MCSB11                MCSB12                          MCSB13A  MCSB13B

CYP3A4(c) 481 GLLQPEKPVV LKVESRDGTV SGA 503
CYP3A4(m) 481 GLLQPEKPVV LKVESRDGTV SGA 503
          MCSB13C MCSB13
```

Figure 5-9 Positionnement des MCSBs sur la séquence du CYP 3A4 par Caliseq (c) et par Matras (m). Dans cet alignement, quatre blocs sont en désaccord, dont deux importants : MCSB2A* et MCSB4 correspondant aux SSE B' et F respectivement.

5.3.4 Bilan de la comparaison

Comparer les alignements provenant de différentes méthodes a été assez délicat, surtout pour juger de la fiabilité relative de chaque alignement, étant donné qu'ils sont tous différents. *A priori*, les alignements structuraux sont censés être les plus « fiables » selon les concepts de la biologie structurale. Néanmoins, comme il a été vu précédemment, les alignements structuraux – qu'ils soient multiples ou « pairwise » – dépendent énormément de la méthode utilisée ainsi que des paramètres choisis par l'utilisateur. Ainsi, aucune des méthodes utilisées n'a donné exactement le même alignement. De ce fait, il est difficile de savoir réellement la place occupée par *Caliseq* parmi les autres méthodes à la seule vue de l'alignement. Seule la reconstruction d'un modèle peut aider à déterminer si *Caliseq* produit des alignements de qualité.

En revanche, pour chaque méthode, des régions conservées fixes et des régions variables sont observées dans les alignements, quelle que soit la nature de la méthode, aussi bien séquentielle que structurale. Il existe également des régions que j'appellerai « semi-conservées », à savoir celles ne présentant aucun gap dans l'alignement, mais jamais positionnées à un même endroit selon les alignements (généralement un décalage de quelques résidus) : elles correspondent aux SSE des boucles B', F, G et la boucle B–C et le feuillet en C-terminale. Dans tous les alignements, ces trois types de régions sont observés, quelle que soit la nature de la méthode. Elles semblent d'ailleurs renseigner sur les propriétés structurales des P450s. En effet, les régions « semi-conservées », celles qui ont posé problème à la fois pour leur identification structurale et leur positionnement, se situent dans des régions SRS identifiées par Gotoh en 1992 (cf. Figure 1-18 de la page 44) : SRS-1, SRS-2, SRS-3 et SRS-6, correspondant respectivement aux hélices B', F, G et au feuillet C-terminal. Sur la structure des P450s, ces SSEs bordent la cavité du site actif[16]. Dans la littérature, ces SSEs de P450 ont été décrits comme super variables : l'hélice B' par exemple, supposée flexible pour laisser passer les substrats, a été observée aussi bien structurée que déstructurée (Poulos et *al*., 1987). Il n'est pas surprenant que séquentiellement, des problèmes de positionnement apparaissent dans ces régions. La bonne nouvelle concerne les régions « vraiment » conservées dans tous les alignements qui reflètent bien cette caractéristique de conservation de repliement sur toutes les structures de CYP. Celles-ci ont bien été identifiées dans les alignements par *Caliseq*, Matras, et HHpred à un niveau moindre, et

[16] En fait, ce n'est pas exactement le feuillet 3 qui borde le site actif, mais plutôt la structure en « beta hairpin ». Il s'agit du coude plus ou moins long selon des P450s qui est formé par les deux brins β_4 antiparallèles et relié à la base par le feuillet 3

difficilement sous Clustalw, sauf pour ce dernier en augmentant le nombre de séquences dans l'alignement.

5.4 Construction de modèles de P450s, quelle méthode adopter ?

5.4.1 Comparaison des modèles reconstruits à une structure connue : CYP 3A4 (1tqn)

En dessous de 30% d'identité en séquence, la modélisation comparative n'est pas la méthode la plus adaptée pour construire un modèle (cf. Figure 2-19 de la page 113) et on lui préfère généralement les méthodes d'enfilage ou « *threading* » (cf. section 2.5.2). Par certains aspects, la méthode de reconstruction par blocs structuraux conservés est proche des concepts des méthodes d'enfilage, dans la mesure où un squelette protéique est créé à partir des blocs, et où les parties inter-blocs sont construites sans information *a priori*, et donc de façon *ab initio*. Cependant, dans la méthodologie développée au laboratoire pour reconstruire les P450s à faible identité en séquence, ce n'est pas une banque de repliements qui est utilisée pour générer la structure du squelette peptidique mais des structures de P450s homologues structuralement, qu'on impose comme *templates*. Le modèle est reconstruit alors à la manière d'une modélisation par homologie. Il s'agirait donc d'une méthode « hybride » entre l'enfilage et la modélisation comparative, du moins pour le cas de protéines à repliement extrêmement conservé. Pour confirmer ces dires, l'idée a consisté ici aussi à comparer notre méthode à différentes méthodes existantes. À la manière du concours CASP, les modèles des séquences cibles sont évalués par rapport à la structure connue des séquences cibles utilisées. J'ai choisi ici le critère de RMSD pour la comparaison entre les structures et les modèles. Plusieurs tests de comparaison ont été effectués, à la fois sur des CYPs d'origine bactérienne ($P450_{cam}$, CYP 119) et sur des CYPs d'origine microsomale (CYP 2C9, CYP 1A2, CYP 3A4). Les résultats étant relativement en accord, je ne présenterai ici que les résultats de comparaison pour les meilleurs modèles du CYP 3A4 (selon les critères d'évaluation de ProsaII, Anolea, ProQ, et de la fonction objective de Modeller), dont les premières structures sont apparues au début de ma thèse (cf. Tableau 5.8)

Tableau 5.8 Comparaison des modèles de CYP 3A4, construits selon différentes méthodes. Le RMSD a été calculé par rapport au squelette peptidique de la structure X (1tqn) par le logiciel Profit. Pour les modèles générés à partir de notre stratégie, un RMSD au niveau des blocs a été également calculé. Le modèle par le serveur SwissModel n'a pas pu être obtenu car son logiciel d'alignement n'était pas en mesure d'aligner les structures imposées au programme. Modeller n'a pas pu construire le modèle Clustalw avec le jeu C en raison également d'un alignement trop mauvais. Pour le dernier cas, on a utilisé des blocs tirés de l'alignement structural de Matras. Pour HHpred, un psi-blast est utilisé : il n'a pas toujours été possible d'utiliser tous les templates d'un jeu donné.

Année	***Modèle***	***Nombre de templates***	***Sur la base du jeu***	***RMSD (Å)***	***RMSD (Å) au niveau des blocs***
2002	Ancienne stratégie	6	*A*	5,83	4.95
2004	Clustalw	6	*C*	échec	-
2004	Nouvelle stratégie	6	*C*	5,06	4.09
2004	Nouvelle stratégie	6	*C'*	6,76	6.04
2005	Clustalw	6	*A*	9,70	-
2005	Clustalw	4	*B*	7,59	-
2005	Nouvelle stratégie	4	*B*	3,53	3.28
2005	HHpred	4	*F*	5,14	-
2005	HHpred	3	*G*	5,53	-
2005	HHpred	10	*H*	5,04	-
2005	SwissModel	6	*C*	échec	-
2006	Nouvelle stratégie	11	*D*	4,19	3.83
2006	Clustalw	11	*D*	4,31	-
2007	Sybyl	2	-	7,30	-
2007	Sybyl	12	-	9,99	-
2007	Matras	10	*D*	3,33	-
2007	Nouvelle stratégie	10	*D'*	5,07	4.50

Ancienne stratégie :	CSBs + SmartConsAlign + Dyana/Xplor
Nouvelle stratégie :	CSBs + Caliseq + Modeller
ClustalW :	Clustalw + Modeller
HHPred :	HHpred + Modeller
SwissModel :	Serveur
Matras :	Matras (avec 1tqn.pdb en template) + Modeller
Sybyl :	Fugue/Orchestrar + Composer

Le Tableau 5.8 retrace l'historique des principaux modèles construits au cours de ma thèse pour la construction du CYP 3A4. La structure de ce cytochrome P450 ayant été publiée pendant ma première année de thèse, les expériences de modélisation avaient pour but unique de valider la méthode mise au point au laboratoire. Plusieurs méthodes ont été utilisées pour le comparatif : l'une d'elle est une version améliorée des méthodes d'enfilage (Fugue/Orchestrar), une autre correspond à un serveur automatique de reconstruction (SwissModel), mais la plupart des méthodes propose uniquement des alignements, qui nécessitent l'utilisation du logiciel Modeller pour la reconstruction 3D. Ces dernières sont celles présentés à la section précédente. Concernant la méthode Fugue/Orchestrar, il s'agit d'une méthode très proche de notre méthode dans sa façon de construire les modèles, puisque Fugue effectue d'abord une recherche sur la base HOMSTRAD des structures qui serviront de *templates*, en comparant pour cela la séquence cible (ou un pré-alignement cible) à des profils structuraux de chaque famille dans la banque de données. Orchestrar, une suite d'applications, prend ensuite le relais, commence à déterminer les régions structuralement conservées entre les *templates* et la séquence cible et construit le modèle (uniquement le squelette peptidique) en fonction de ces informations. Les

boucles sont ensuite reconstruites à partir d'une banque de repliement et les chaînes latérales sont enfin ajoutées en accord avec une banque de rotamères. Cette méthode ressemble beaucoup à la méthode de reconstruction utilisée au laboratoire par N. Loiseau et M. Cottevieille.

Les RMSDs ont été calculés en utilisant l'algorithme de McLachlan (McLachlan, 1982) par le logiciel Profit (Martin, A.C.R., http://www.bioinf.org.uk/software/profit/), sur les atomes du squelette peptidique pour centrer la comparaison sur les repliements. Ainsi, pour le CYP 3A4, aucune méthode n'a pu produire de modèle à moins de 3 Å de la structure 1tqn utilisée. Il faut néanmoins garder en mémoire que pour deux structures d'une même isoforme, une différence de RMSD existe. Ainsi, rien qu'en comparant la structure nue du CYP 3A4 (1tqn) avec une structure disposant d'un substrat comme l'érythromycine (2j0d) ou le ketaconazole (2v0m), des RMSD de 1,21 et 1,74 Å respectivement sont observés. Plus impressionnant encore, le RMSD entre la conformation fermée (1suo) et la conformation ouverte (1po5) du CYP 2B4 est trouvé à 5,53 Å montrant l'énorme flexibilité de la molécule. De ce fait, une différence de 0,5 à 1 Å pour les modèles n'est pas réellement significative, et peut être considérée comme un bruit de fond.

Ainsi, un modèle proche de 3 Å de RMSD peut être considéré comme un modèle relativement correct. De tous les modèles produits, seuls ceux obtenus par Matras (avec utilisation de la séquence d'1tqn pour guider l'alignement structural ce qui introduit le biais en faveur de Matras) et par la nouvelle méthode de reconstruction (combinaison de GOK, *Caliseq* et Modeller) avec les *templates* de M. Cottevieille (jeu ***B***) s'en approchent le plus, avec respectivement des RMSDs de 3,33 Å et 3,58 Å. Ce dernier constat est plutôt surprenant dans la mesure où le jeu ***B*** comporte peu de *templates*. Les plus « mauvais » modèles ont été obtenus par Sybyl, selon la méthode Fugue/Orchestrar, avec des RMSD de 7,30 Å à 9,99 Å. Les méthodes « hybrides » (***C'*** et ***D'***) n'ont pas donné d'excellents modèles non plus, avec 6,76 Å et 5,07 Å respectivement. À noter qu'une légère amélioration du modèle est perceptible lors de l'utilisation d'un nombre plus important de *templates* dans l'alignement.

Aucun modèle de CYP 3A4 n'a été obtenu par le serveur SwissModel en raison des *templates* imposés au serveur (ceux du jeu ***C***) : SwissModel n'était pas en mesure d'aligner les *templates* fournis. Toutefois, ce résultat doit être considéré avec précaution : il n'est désormais plus possible d'imposer ses propres *templates* sur le serveur. Par ailleurs, il n'est plus possible d'utiliser ce programme à des fins de comparaison avec notre méthode dans la mesure où il imposera forcément la structure de la séquence cible (1tqn dans le cas du CYP 3A4) dans son alignement.

À l'image des modèles obtenus par Fugue/Orchestrar, les alignements de Clustalw conduisent à de mauvais modèles lorsque peu de *templates* sont utilisés (jeu B et C, avec respectivement 4 et 6 *templates* dans chaque jeu). Dans un cas notamment (lors de l'utilisation des *templates* du jeu ***C***) Modeller n'a pas réussi à produire de modèle tant les distances entre les résidus des séquences de l'alignement étaient importantes, et ce en raison d'un mauvais alignement. En revanche, avec un nombre de *template*s plus important, une amélioration notable du modèle est observée, à peu près équivalent (voire sensiblement meilleur) au modèle obtenu par l'alignement effectué par *Caliseq* avec le même nombre de *templates*.

Justement, qu'en est-il des modèles produits par la méthode de reconstruction à bas taux d'identité développée au laboratoire ? Le modèle obtenu par N. Loiseau avec l'ancienne méthode (Combinaison de GOK, SmartConsAlign et Dyana/X-plor) a un RMSD plutôt élevé (5,83 Å) par rapport à la vraie structure du CYP 3A4 lorsqu'il est calculé sur toute la longueur de l'enzyme, et descend à 4,95 Å, soit un gain de près de 1 Å lorsque le RMSD n'est calculé qu'au niveau des CSBs. Cette observation souligne le fait que la reconstruction des régions inter-blocs posait déjà un sérieux problème et n'était pas si bien résolue par le recuit simulé opéré par Xplor. Concernant les modèles obtenus par la nouvelle méthode (combinaison GOK, *Caliseq* et Modeller), outre le modèle produit à partir du jeu ***B***, les RMSDs semblent corrects, situés aux alentours de 4-5 Å. Au niveau des blocs, un gain de 1 Å environ est aussi observé.

Pour des jeux de *templates* plus réduits, HHpred donne également des modèles assez proches de ceux obtenus par notre méthode, aux alentours de 5 Å.

5.4.2 Choix de la méthode idéale ?

5.4.2.1 Discussion et évaluation

Pour toutes les méthodes, les RMSDs semblent fluctuer énormément en fonction des *templates* utilisés : typiquement, les modèles du CYP 3A4 semblent moins bons que lorsque les *templates* sont dérivés du jeu ***C*** (jeux ***C*** et ***F***) alors qu'ils ont tendance à s'améliorer lorsque des dérivés du jeu ***B*** sont utilisés (jeux ***B*** et ***G***) : cela est observé aussi bien sur la nouvelle méthode que sur Clustalw ou HHpred. Ce constat va à l'encontre des idées que j'avais eues lors des choix de mon jeu de *templates*. En effet, dans le jeu ***C***, j'avais mis autant de structures issues de P450s bactériens que de structures issues de P450s microsomaux. Je pensais alors que les structures de P450s microsomaux m'aideraient à me rapprocher de la structure du CYP 3A4. Par ailleurs, même si les résultats ne sont pas montrés dans ce manuscrit, le jeu ***C*** est celui qui a donné les plus mauvais modèles avec les autres tests ($P450_{cam}$, CYP

119, CYP 1A2) excepté le modèle du CYP 2C9. Une explication probable est la présence en nombre trop important de *templates* de la famille des 2C dans ce jeu, qui orienterait toutes les reconstructions vers un modèle proche des structures de CYP de la famille 2C. En revanche, cette information pourrait être diluée lorsque d'autres *templates* sont ajoutés au jeu, comme dans le cas du jeu global ***D***.

Les alignements proposés par Clustalw et par *Caliseq*, desquels dépend la reconstruction du modèle, sont d'ailleurs meilleurs avec un nombre de *templates* plus important dans l'alignement (chose qui avait été déjà démontré pour Clustalw) : un gain d'au moins 1 Å est observé ainsi. En outre, les imprécisions de reconstruction des régions inter-blocs semblent également s'amoindrir avec la nouvelle méthode et lorsque le nombre de *templates* est plus important. En dépit des problèmes rencontrés pour aligner les régions inter-blocs, il s'avère au final que Modeller est parvenu à « corriger » les régions approximativement alignées en partie inter-bloc. Ainsi, toutes les méthodes visant à optimiser l'alignement inter-bloc n'ont finalement pas été exploitées comme évoqué plus haut, et les méthodes visant à optimiser structuralement les parties inter-blocs, comme la méthode de minimisation par blocs développée par K. Zimmermann ou la reconstruction *ab initio* de boucles par Modeller, non plus. Dans aucun cas, ces dernières méthodes n'ont permis de diminuer le RMSD du modèle à la structure 1tqn du CYP 3A4.

Il est difficile d'expliquer en revanche le bon RMSD attribué au modèle issu de la nouvelle méthode à partir du jeu ***B*** qui n'a nécessité qu'un alignement inter-bloc manuel. Est-ce en raison du nombre de blocs plus important dans ce jeu, ou encore de la nature des *templates* utilisés dans ce jeu ? Nous n'avons pour le moment pas d'explication à proposer (à noter que ce même alignement produit des modèles de qualité reproductible, avec une variation de quelques dixièmes d'Angström).

Partant d'un bon concept, Fugue/Orchestrar n'a pu fournir de modèles réellement convaincants. Le principal problème est survenu lors du choix des *templates* proposés par Fugue : le logiciel conseillait de ne prendre que deux structures (d'origine bactérienne qui plus est) pour pouvoir par la suite identifier les régions structuralement conservés (nommé SCR dans le programme). À la place j'ai imposé au programme le jeu des 12 *templates* disponibles dans le programme pour un essai de reconstruction du CYP 3A4. Autant les SCRs ont été simples à identifier pour Orchestrar dans le jeu de 2 *templates*, autant le programme a eu du mal pour le jeu des 12 *templates*. Cela s'est ressenti lors de la production du modèle : un modèle déstructuré à 9,99 Å de la véritable structure du CYP3A4. Une cause possible peut être liée à la banque de structure HOMSTRAD, peut être trop obsolète.

Néanmoins, je doute qu'avec plus de structures le programme ne s'en serait mieux sorti, étant donné les difficultés qu'il a eues pour identifier des SCRs sur les 12 *templates*.

Le serveur HHpred qui fournit des alignements pour Modeller a généré des modèles de qualité moindre que ceux générés par notre méthode, avec moins de *templates*. Il nécessite également moins de manipulations pour le modélisateur. Néanmoins, les *templates* utilisés n'ont pas été nécessairement les mêmes que ceux utilisés dans notre méthode. Parmi les *templates* utilisés, surtout pour le jeu ***H***, de nombreuses structures n'étaient pas disponibles lors de l'analyse sous GOK. Idéalement, il aurait fallu refaire une analyse GOK avec les *templates* du jeu ***H*** pour pouvoir réellement comparer les modèles.

5.4.2.2 Verdict

Au final, quelle serait la méthode la plus adaptée pour la reconstruction d'un CYP inconnue à basse identité de séquence ? En fait, tout dépend du moment de la reconstruction sur l'échelle de temps. Il semble évident qu'à ce jour, le nombre de *templates* disponibles dans la PDB est tel qu'un alignement simple sous Clustalw permet de produire des modèles convenables et meilleurs que ceux qu'on pourrait obtenir avec un enfilage. HHpred semble être également un bon candidat dans la mesure où les modèles obtenus sont généralement de qualité constante quel que soit le nombre de *templates* utilisés. D'autant plus qu'HHpred fournit directement le fichier d'alignement pour Modeller. En revanche, lorsque peu de structures sont disponibles et donc exploitables en tant que *templates*, il semblerait bien qu'HHpred et la méthodologie mise au point au laboratoire soient à égalité. HHpred serait plus rapide et moins complexe d'utilisation. Néanmoins, il n'est pas sûr que l'alignement produit avec moins de *templates* soit aussi fiable : celui-ci dépend du profil HHM qui a été calculé en prenant en compte toutes les séquences des structures de P450s disponibles au moment où j'ai effectué le test de comparaison. En bref, la nouvelle méthode semble aujourd'hui un peu dépassée, mais demeure probablement une bonne méthode de reconstruction par homologie à basse identité de séquence, et lorsque les *templates* disponibles sont en nombre limité. Pour vérifier cette hypothèse, il faudrait donc dans l'idéal expérimenter la méthode sur une autre famille de protéines à repliement très conservé.

5.5 Vers l'obtention d'une banque de P450s

5.5.1 Données initiales

5.5.1.1 Les CSBs, un nouveau critère de sélection

Nous avons signalé que l'information de repliement conservé des P450s contenue dans les CSBs pouvait être exploitée pour la reconstruction de modèles 3D de CYPs inconnus. Il était alors intéressant de savoir si cette information pouvait également servir de « signature » structurale pour identifier spécifiquement les P450s dans des banques de données ou des génomes nouvellement séquencés. L'utilisation des CSBs dans la génomique exploratoire n'a nécessité qu'une légère modification du programme *Caliseq*, lui offrant ainsi la possibilité de positionner les blocs sur toutes les séquences d'un fichier qui correspond à une banque de séquences. Grâce au score d'alignement pour chaque séquence et à un seuil défini, *Caliseq* détermine si la séquence testée appartient à la superfamille des P450s. Ainsi, en traitant toute une banque de séquences, une banque spécifique de P450s peut en être extraite et exploitée : combinée avec notre méthode de reconstruction, il serait envisageable d'obtenir une banque de sites actifs pour des études d'arrimage par exemple.

5.5.1.2 Méthodes déjà existantes de sélection des P450s

Il existe de nombreuses ressources dédiées aux CYPs sur Internet, comme par exemple « the Cytochrome P450 Homepage » (http://drnelson.utmem.edu/CytochromeP450.html), « the P450 Knowledgebase » (http://cpd.ibmh.msk.su/) (Lisitsa *et al.*, 2001), « the Arabidopsis P450 database » (http://www.p450.kvl.dk/), « the Insect P450 Site » (http://p450.antibes.inra.fr/) etc. Étonnamment, aucune banque globale de P450s (de séquences ou de structures) n'a été mise au point. Pourtant les outils ne manquent pas : un certain nombre de méthodes existent déjà permettant d'attribuer le caractère « P450 » à une séquence donnée. Parmi ces méthodes, deux nous parraissent intéressantes : celles utilisées par Prosite et Pfam.

Ainsi, Prosite (cf. section 2.3.4.1) est une base construite sur une méthode qui reconnaît un motif (ou pattern) particulier présent sur la séquence des protéines. Dans le cas des P450s, ce motif est appelé « *Cytochrome P450 cysteine heme-iron signature* ». Pfam (cf. section 2.3.4.3) est une banque de domaines. Dans cette base, les domaines de P450 correspondent à des alignements de P450s, codés dans des profils HMM. Par commodité, je ferai dans la suite de ce chapitre l'amalgame entre le nom de la base et la méthode associée.

J'ai utilisé ces deux méthodes pour construire des bases spécifiques de séquences de P450s qui me serviront d'éléments de comparaison avec la base que j'ai construite avec *Caliseq* et un jeu de blocs. Pour les trois méthodes à comparer, les banques de P450s sont obtenues à partir des bases de séquences SptrEmbl (supposées non redondantes) de 2005 et de 2007. À noter que les premiers essais de *Caliseq* avec l'ancien système de score ont été opérés sur la base de 2005, tandis que le nouveau score a été expérimenté sur la dernière version 2007 de SptrEmbl. Le nombre de séquences de P450s identifiées par les deux méthodes sur les bases de séquences SptrEmbl2005 et SptrEmbl2007, est présenté en Tableau 5.9

Tableau 5.9 Nombre de séquences de P450s identifiées sur SptrEmbl par Pfam et Prosite.

Année	*Nombre de séquences dans SptrEmbl*	*Nombre de CYPs identifiés par Pfam[1]*	*Nombre de CYPs identifiés par Prosite[2]*
2005	2345429	5166	3392
2007	3169275	6478	4859

[1] les données de Pfam ont été récupérées à partir d'un unique fichier SptrEmblPfam distribué par Pfam
[2] les données de Prosite ont été récupérées à partir de deux fichiers (uniprot_swissprot et uniprot_trEmbl) distribuées par la banque Uniprot

À première vue, la méthode Pfam permet d'identifier plus de P450s que la méthode Prosite. En revanche, entre 2005 et 2007, Prosite semble avoir subi une légère amélioration : alors que Pfam identifie 1312 nouvelles séquences, Prosite en identifie 1487. La différence, soit 175 séquences, correspond aussi bien à des nouvelles séquences dans SptrEmbl2007 que Pfam n'a pas identifiées, mais également des séquences préexistantes dans SptrEmbl2005, présentes dans Pfam et que Prosite n'avait pas identifiées à l'époque. En réalité, il semblerait qu'entre ces deux années, l'algorithme de Prosite ait légèrement évolué, et la sélection ne se fait plus exclusivement sur un motif séquentiel.

5.5.2 Création des bases de P450s par Caliseq

Caliseq a été lancé sur les deux bases SptrEmbl2005 et SptrEmbl2007, en faisant varier à chaque essai un paramètre différent : l'idée était alors de déterminer les paramètres idéaux pour la détection de séquences de P450. Chaque criblage sur banque complète par *Caliseq* prend environ une heure de calcul[17]. Ces résultats sont montrés au Tableau 5.10.

[17] La mesure a été réalisée en utilisant seulement un processeur Xeon 3 Ghz d'un seul nœud d'un cluster de PC.

5.5.2.1 Processus d'obtention du paramétrage optimal

Les paramètres optimaux ont été affinés de la manière suivante : chaque fois qu'une valeur de paramètre a permis d'obtenir un résultat jugé convenable, il était conservé pour faire varier un autre paramètre et ainsi de suite. De faibles incréments du paramètre ont été choisis pour l'exploration. Il s'avère que les résultats sont peu sensibles dans une certaine zone (pas d'augmentation ou de diminution significative du nombre de séquences récupérées), et basculent rapidement dès qu'un certain seuil est franchi. Durant cette phase de recherche, j'ai choisi de favoriser les faux négatifs par rapport aux faux positifs pour la banque de P450s : le choix qui a été privilégié était de ne pas détecter des séquences de P450s plutôt que d'attribuer une séquence en tant que P450 alors qu'elle ne l'est pas.

5.5.2.2 Évolution du ratio T

La détermination du seuil d'indentification de P450 par *Caliseq* a évolué au cours de la thèse. Ainsi, dans les premiers essais sur la SptrEmbl2005, le seuil de sélection reposait sur un ratio (cf. section 4.3.2.1), que j'ai appelé T. Dans les premiers criblages, j'ai fait évoluer T d'une valeur de 0,05 à 0,40 par pas de 0,05. Selon la formule de ce seuil, plus le ratio (ou coefficient) est bas, plus le seuil sera proche d'un score attribué à un alignement des blocs sur une séquence aléatoire. À l'inverse, plus le ratio est élevé, plus le seuil sera sélectif et proche d'un score attribué à un alignement parfait des blocs sur eux-mêmes. Ainsi, dans le Tableau 5.10, le jeu ***A*** (*templates* de N. Loiseau) combiné à des ratios T de valeur faible entraîne l'attribution d'un nombre très élevé de séquences à des P450s par *Caliseq* (*Nombre de séquences récupérées* dans le Tableau 5.10). Sachant que Pfam et Prosite n'en décelaient au maximum que 5566 (cf. Tableau 5.9) la présence d'un nombre important de faux positifs était fort probable. Cela est d'autant plus vrai qu'à un seuil de 0,05, *Caliseq* identifie presque la moitié de la base StprEmbl2005 ! Puis, de 0,05 à 0,25, le nombre de séquences récupérées décroît relativement rapidement, passant de la moitié de la banque récupérée au dixième.

À partir d'un seuil de 0,30, la sélection passe un palier significatif, du dixième de la SptrEmbl récupérée au cinquantième. Passé le seuil de 0,35 pour ce ratio (toujours avec le jeu ***A***) Caliseq commence à donner des résultats plus proches de Pfam et Prosite, dans les 3000 séquences identifiées comme P450s.

Tableau 5.10 Comparatif des méthodes de sélection de P450s par Caliseq/Pfam/Prosite. Pour tous les calculs, le poids de gap dans *Caliseq* a été fxé à 2. Dans certain cas (lorsque la limite L a été mise à 350), un jeu de 10 blocs a été utilisé à la place de la vingtaine de blocs des jeux A, *B* ou *C*. Les meilleurs paramètres (pour l'ancien seuil et le nouveau seuil) sont indiqués en gras.

Jeu de bloc	*base*	*Méthode*	*seuil*	*limite*	*nb séquences récupérées*	*Exclusif à Caliseq*	*Exclusif à Pfam*	*Exclusif à Prosite*	*Exclusif à Caliseq et Pfam*	*Exclusif Caliseq et Prosite*	*Exclusif Pfam et Prosite*	*Commun*	*% faux positif*
A	sptrEmb2005	T-ratio	0,05	1000	926721	922313	654	0	1296	98	202	3014	99,52
A	sptrEmb2005	T-ratio	0,05	800	874849	870480	673	0	1277	90	214	3002	99,50
A	sptrEmb2005	T-ratio	0,10	1000	907779	903376	659	0	1291	98	202	3014	99,51
A	sptrEmb2005	T-ratio	0,10	800	855908	851544	678	0	1272	90	214	3002	99,49
A	sptrEmb2005	T-ratio	0,15	1000	807466	803106	688	0	1262	92	210	3006	99,46
A	sptrEmb2005	T-ratio	0,15	800	755596	751275	707	0	1243	84	222	2994	99,43
A	sptrEmb2005	T-ratio	0,20	1000	612523	608225	728	0	1222	81	221	2995	99,30
A	sptrEmb2005	T-ratio	0,20	800	560669	556410	747	0	1203	73	233	2983	99,24
A	sptrEmb2005	T-ratio	0,25	1000	301691	297543	837	0	1113	56	237	2979	98,63
A	sptrEmb2005	T-ratio	0,25	800	249985	245876	856	0	1094	48	249	2967	98,36
A	sptrEmb2005	T-ratio	0,30	1000	42760	38790	936	0	1014	19	279	2937	90,72
A	sptrEmb2005	T-ratio	0,30	800	16428	12491	955	0	995	17	291	2925	76,03
A	**sptrEmb2005**	**T-ratio**	**0,35**	**1000**	**3680**	**34**	**1141**	**0**	**809**	**11**	**390**	**2826**	**0,22**
A	sptrEmb2005	T-ratio	0,35	800	3633	15	1157	0	793	11	402	2814	0,00
A	sptrEmb2005	T-ratio	0,40	1000	2558	5	1488	0	462	4	1129	2087	0,00
A	sptrEmb2005	T-ratio	0,35	350	2845	14	1282	0	668	9	1062	2154	0,00
A	sptrEmb2005	T-ratio	0,40	350	2057	3	1514	0	436	5	1603	1613	0,00
B	sptrEmb2005	T-ratio	0,35	1000	2280	0	1566	0	384	0	1320	1896	0,00
C	sptrEmb2005	T-ratio	0,35	1000	1610	1	1708	0	242	0	1849	1367	0,00
A	sptrEmb2005	T-ratio	1,50	1000	3123	10	1344	0	606	7	716	2500	0,00
A	sptrEmb2005	T-ratio	1,25	1000	3489	13	1208	0	742	8	490	2726	0,00
A	sptrEmb2005	T-ratio	0,50	1000	3664	23	1144	0	806	11	392	2824	0,14
A	sptrEmb2005	T-ratio	0,25	1000	3664	23	1144	0	806	11	392	2824	0,14
A	sptrEmb2005	T-ratio	0,05	1000	3664	23	1144	0	806	11	392	2824	0,14
A	sptrEmb2005	T-ratio	0,05	1200	3713	58	1133	0	817	11	389	2827	0,78
A	sptrEmb2007	Z-score	0,05	1000	4508	22	1262	264	636	15	745	3835	0,02
A	sptrEmb2007	Z-score	0,05	1200	4534	36	1260	264	638	15	735	3845	0,20
D	sptrEmb2007	Z-score	0,05	1000	4417	16	1272	266	626	13	818	3762	0,00
D	sptrEmb2007	Z-score	0,05	1200	4436	24	1269	266	629	13	810	3770	0,09
E	sptrEmb2007	Z-score	0,05	1000	4906	28	1158	252	740	27	469	4111	0,00
E	**sptrEmb2007**	**Z-score**	**0,05**	**1200**	**4914**	**28**	**1156**	**252**	**742**	**27**	**463**	**4117**	**0,00**

A : Jeu de 6 *templates* de N. Loiseau (GOK)
B : Jeu de 4 *templates* de M.Cottevieille (GOK)
C : Jeu de 6 *templates* TA.Nguyen (GOK)
D : Jeu de bloc global 11 *templates* (alignement)
E : Jeu de 6 *templates* de TA.Nguyen (GAKUSA)

Nb : Nombre
Base : Base de séquences, utilisée pour les identifications de P450s
Limite : Distance maximale autorisée premier bloc/dernier bloc
Méthode : Méthode pour déterminer le seuil d'identification de P450s

%faux positif : Nombre de faux positifs dans les séquences inédites de *Caliseq*

5.5.2.3 *Évolution du seuil Z (Z-score)*

Une nouvelle fonction de score basée sur un « log-odd » pour coder les profils (cf. section 4.3.1.2), ainsi que l'utilisation du calcul d'un Z-score comme critère de sélection (cf. section 4.3.2.3), ont été implémentés dans le but d'améliorer la sensibilité de la méthode lors de la recherche sur banque (réduction des faux positifs dans les séquences récupérées et réduction des faux négatifs non identifiés). Ils ont été testés sur une banque mise à jour, la SptrEmbl2007. En raison du nombre important des séquences contenues dans cette banque, et du fonctionnement de *Caliseq* lié au calcul du Z-score –*Caliseq* effectue un double passage : le premier pour récupérer les meilleurs scores d'alignement et le second pour sélectionner ceux dont le Z-score est au dessus du seuil souhaité– il faut un peu plus de temps pour obtenir les résultats qu'avant, mais cela demeure raisonnable pour un algorithme de programmation dynamique (de 1 heure ½ à 2 heures sur un nœud du clusteur). À l'instar du seuil de T-ratio, j'ai fait évoluer la valeur seuil pour le Z-score de 0,05 à 0,20 par pas de 0,05. La nouvelle méthode est réellement plus sensible : même avec de mauvais paramètres, elle récupère dès le départ moins de faux positifs. Dans tous les cas, le nombre est limité par la structure de l'arbre qui mémorise les meilleures séquences. Ainsi, si l'utilisateur décide de récupérer seulement 10000 séquences au premier passage de *Caliseq*, le maximum de séquences pouvant être récupérées au second passage est également de 10000 séquences. Quoiqu'il en soit, avec cette fonction de score en « log-odd » et le filtre sur les z-score, il s'agissait cette fois de trouver les bons paramètres pour augmenter le nombre de séquences récupérées plutôt que de le faire diminuer, comme c'était le cas avec la précédente méthode avec le T-ratio. Finalement, la meilleure valeur de seuil a été trouvée pour 0,05. Cette valeur est très proche de la moyenne, comme présentée à la Figure 4-3, page 170 : on récupère donc quasi toutes les valeurs positives de z-score, à savoir, quasi toutes les séquences identifiées par *Caliseq*, P450 ou non (le seuil est vraiment très bas).

5.5.2.4 *Influence de la limitation sur la distance entre les blocs*

La limitation de la distance entre le premier résidu du premier bloc et le dernier résidu du dernier bloc a été mise en place pour limiter le nombre de faux positifs récupérés par *Caliseq*. En effet, le score d'alignement dépend du positionnement des blocs sur la séquence cible. Quelle que soit la distance séparant ces blocs, tant que le positionnement des blocs est correct, le score est élevé, puisque les pénalités de glissement des blocs ne sont pas comptées. Imposer une limitation de distance à 800-1000 permet de supprimer tous les faux positifs de grande taille dont le score d'alignement est favorable. Bien que ne figurant pas dans le tableau, cette limitation a permis de réduire d'1/5 le nombre de séquences récupérées lors de la sélection par un T-ratio par *Caliseq* (pour la plupart des

faux positifs). En revanche, lorsque le Z-score a été utilisé, la limitation a pu être augmentée légèrement, pour récupérer plus de séquences : la méthode combinant le Z-score et la nouvelle fonction de score étant plus sélective, il fut possible de récupérer plus sélectivement des séquences qui ne pouvaient être récupérées par l'ancienne méthode sans entraîner avec elles des faux positifs.

5.5.2.5 Jeu de templates utilisé

Un total de cinq jeux de *templates* a été utilisé pour construire une base de P450 à l'aide de *Caliseq*. Le jeu ***A*** est celui qui a servi pour tester les effets des variations de paramètres. Lorsque le meilleur compromis a été trouvé pour les paramètres, les autres jeux de templates ont été utilisés : ***B*** et ***C*** pour la base SptrEmbl2005, et ***D*** et ***E*** pour la base SptrEmbl2007.

5.5.3 Comparaison avec les autres méthodes : Prosite et Pfam

Sur le Tableau 5.10, sont présentés également les recouvrements des séquences récupérées avec ceux de Prosite et ceux de Pfam. Pour obtenir ces recouvrements, les numéros d'accession de chaque séquence ont été utilisés : il devenait alors possible de vérifier si une séquence était identifiée par toutes les méthodes. Ainsi, pour chaque ligne du tableau, la composition des séquences récupérées par *Caliseq* est renseignée, à savoir combien de séquences sont exclusives à i) *Caliseq*, ii) à Pfam, iii) à Prosite, iv) combien sont trouvées uniquement dans *Caliseq* et Pfam, v) *Caliseq* et Prosite, vi) Pfam et Prosite, et enfin vii) combien sont trouvées en commun par les trois méthodes. Une information supplémentaire a été indiquée pour les séquences identifiées comme P450s exclusivement par *Caliseq* : il s'agit du taux de faux positifs comptabilisés dans cet ensemble (*exclusif à Caliseq*). Le dénombrement des faux positifs résultent d'une expertise manuelle des séquences quand cela est possible (nombre de séquences *exclusif à Caliseq* < 200). Dans les cas où les séquences identifiées par *Caliseq* sont trop nombreuses, toutes ont étés considérées comme faux positifs (vérification manuelle impossible). Pour les séquences récupérées de la SptrEmbl2005, il était en effet possible d'avoir une idée sur la nature de ces séquences grâce à la SptrEmbl2007. En revanche, dans le cas de la SptrEmbl2007, j'ai été amené à vérifier l'annotation de la séquence (annotée comme CYP, putatif ou hypothétique, etc.). J'ai également eu recours à d'autres serveurs comme SMART ou ProDom qui pouvaient m'indiquer selon leurs critères si ces séquences sont des P450s. En dernier recours, il était possible de vérifier après une recherche PSIBLAST si des séquences de P450s proches apparaissent en résultat.

5.5.3.1 Quelques remarques générales

Examinons d'abord les différences globales entre les trois ensembles de séquences identifiées comme P450 par les trois différentes méthodes. Dans le Tableau 5.9, Pfam identifiait plus de P450s que Prosite aussi bien sur SptrEmbl2005 que sur SptrEmbl2007. Pourtant, toutes les séquences identifiées par Prosite ne sont pas incluses dans ceux identifiés par Pfam et *vice versa* (colonnes 10 et 11 sur le tableau). Néanmoins Pfam comporte plus de séquences inédites que Prosite.

Par ailleurs, Prosite semble avoir connu une amélioration entre sa version 2005 et sa version en 2007 : sur la colonne des séquences uniquement identifiés par Prosite, aucune séquence n'a été trouvée exclusivement par Prosite dans la SptrEmbl2005 par rapport aux deux autres méthodes, tandis qu'environ 260 séquences sont exclusives à Prosite dans la SptrEmbl2007. Même avec les meilleurs paramètres pour *Caliseq* et l'utilisation du meilleur jeu de blocs, une dizaine seulement de ces séquences ont pu être détectées par *Caliseq*, et environs 250 séquences restent propres à Prosite.

Un détail amusant pour *Caliseq* est que même en récupérant la moitié des séquences de SptrEmble2005 (première ligne du tableau), il n'a pas été en mesure de récupérer 856 séquences présentes dans Pfam, dont 202 en commun avec Prosite. Ce constat est donc fort surprenant

5.5.3.2 Obtention des meilleurs paramètres pour Caliseq

Avec l'ancienne méthode, l'augmentation de la valeur du seuil T-ratio a permis de diminuer drastiquement le nombre de séquences récupérées par *Caliseq*, dont la majeure partie correspondait à des faux positifs. Au fur et à mesure que le nombre de séquences récupérées atteint un seuil convenable, il est possible parallèlement de voir « réapparaître » des séquences parmi les séquences exclusivement déterminées par Pfam ou Pfam/Prosite. À l'inverse, le nombre de séquences exclusivement communes entre *Caliseq* et Pfam, entre *Caliseq* et Prosite, ou encore communes aux trois méthodes, tend à diminuer. Ceci caractérise en fait les faux négatifs non détectés par *Caliseq*, à savoir des séquences de P450s qui ne sont plus identifiées par *Caliseq*.

Partant du principe qu'il est préférable d'avoir des faux négatifs que des faux positifs, un paramétrage de *T-ratio=0,35* et *L=1000* avec le jeu ***A***, fut un bon compromis (cf. Figure 5-10 A) : 3680 séquences ont été ainsi détectées par *Caliseq*, et seulement 1531 séquences (1147 de Pfam + 390 de Pfam et Prosite) identifiés par les deux autres méthodes n'ont pu être détectées par *Caliseq*. Parmi les 3680 séquences, 34 sont trouvés exclusivement par *Caliseq*, avec seulement 8 faux positifs (détecter comme étant des transporteurs et des kinases par les autres serveurs), soit un taux de faux

positifs égal à 0,22% (en considérant que les 3646 autres séquences détectées en accord avec les deux autres méthodes sont des vrais P450s). De manière encourageante, parmi ces 26 séquences (34 –8 faux positifs), l'une d'elles est inconnue de Prosite et de Pfam sur la StprEmbl2005, mais identifiée comme P450 par Pfam sur la SptrEmbl2007.

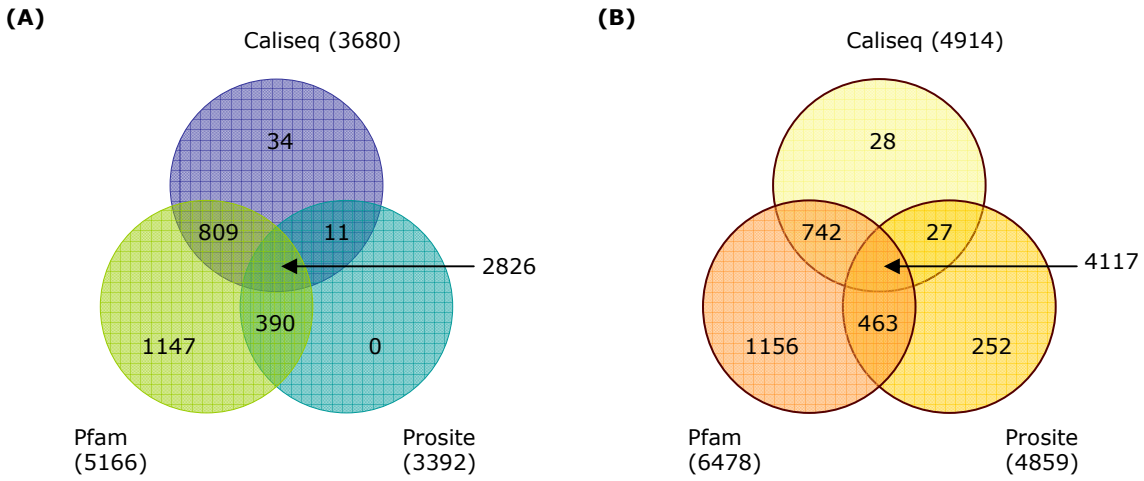


Figure 5-10 Représentation en diagramme de Venn des séquences de P450s de la SptrEmbl, identifiées selon les trois méthodes (Caliseq, Pfam et Prosite). En (A) figurent les résultats pour la SptrEmbl2005 où Caliseq a été utilisé avec le jeu de blocs de N. Loiseau avec le premier système de scoring (T-ratio) et en (B) sont présentés les résultats pour la SptrEmbl2007 où Caliseq a été utilisé avec le jeu de blocs déterminé par GAKUSA avec le nouveau système de scoring (Z-score).

Le **T-ratio** permet donc de **trouver le bon seuil d'élimination** quasi-totale des faux positifs, tandis que le **paramètre L** permet **d'ajuster la sélection**. Par exemple, si on resserre le paramètre L de 1000 à 800, le nombre total de séquences passe de 34 à 15, et ces 15 séquences sont incluses dans les 34. De plus, il existe dans la base Prosite de P450s des faux positifs connus, issus des séquences de la Swissprot (une sous partie de la SptrEmbl). Ces séquences annotées par Prosite comme faux positifs ne font pas partie des 3680 séquences récupérées par *Caliseq*. Les jeux ***B*** et ***C*** n'ont pas donné de meilleurs résultats, ce qui n'est pas surprenant pour le jeu ***C***, mais plus pour le jeu ***B*** qui contient moins de *templates*. Une plus grande diversité du jeu ***A*** par rapport au jeu ***B*** pourrait éventuellement être un élément de réponse. Quoiqu'il en soit, *Caliseq* semblerait être une bonne méthode à l'interface entre Pfam et Prosite : les CSBs sont parfaitement adaptés pour servir de signature structurale dans des expériences de génomique exploratoire.

Lors des essais avec la fonction de score en « log-odd » et du Z-score, la recherche du paramétrage a été opérée à l'inverse de la première méthode : au lieu d'essayer d'éliminer les faux positifs, nous avons cherché à abaissé le taux de faux négatifs. À titre d'exemple, le paramètre sur la distance entre les blocs, L, n'affecte plus les séquences uniquement trouvées par *Caliseq*. En passant d'une valeur de *L* de 1000 à 1200, quelques séquences seulement, qui étaient uniquement identifiées soit par Prosite, soit par Pfam, sont désormais reconnues par *Caliseq*. Avec les jeux ***A*** et **D**, *Caliseq* récupérait moins de séquences que Pfam (6478) ou Prosite (4859). En revanche, avec le jeu **E** (jeu des 28 *templates*), *Caliseq* retrouve sa position à l'interface des deux autres méthodes alors que le jeu de blocs ***E*** diffère franchement en nombre et longueur de bloc (cf. Figure 5-10B).

Tableau 5.11 Identité des séquences identifiées par Caliseq comme CYP sur la SptrEmbl2007 avec le jeu de bloc issu de GAKUSA. Toutes ces séquences sont en réalité des P450s non identifiées par les autres méthodes.

Numéro d'accession	***Désignation de la séquence***
Q27DW3	Conserved hypothetical heme-thiolate monooxygenase.
Q28N78	Cytochrome P450 family protein.
Q25S09	Hypothetical protein.
Q3GYB6	Probable cytochrome P450 125 Cyp125.
Q3W4R4	Putative cytochrome P450.
Q3W4R5	Putative cytochrome P450-family protein.
Q70AR6	Putative cytochrome P450 (EC 144).
Q93PA6	MS125, putative cytochrome P450.
Q9F840	TDP-4-keto-6-deoxyhexose 3,4-isomerase.
P95746	Hypothetical NDP hexose 3,4 isomerase.
Q2P9Y8	Cytochrome P450-like.
Q3WMN4	Similar to Cytochrome P450.
Q3WV35	Fatty acid alpha hydroxylase.
Q4NF21	Fatty acid alpha hydroxylase.
Q5SFA9	Putative NDP-hexose 3,4-isomerase.
Q67G14	CvhE.
Q9RN55	SnogN.
Q5YNS9	Cytochrome P450 monooxygenase.
Q93RT1	Putative cytochrome P450.
Q9L141	Putative cytochrome P450-family protein.
Q73XP8	Hypothetical protein.
Q82GL6	Cytochrome P450 hydroxylase.
Q9RJQ6	Putative cytochrome P450.
Q2J4W2	Putative cytochrome P450.
Q3J3U8	Putative fatty acid beta hydroxylase (Cytochrome P450) (EC 14.-.-).
Q47KL4	Putative cytochrome P450.
Q47L36	Putative cytochrome P450-family protein.
Q2G529	Cytochrome P450 family protein.

Ainsi, le meilleur paramétrage de Caliseq pour identifier des P450s sur la SptrEmbl2007 fut *Z=0.05*, *L=1200* avec le jeu ***E*** (cf. Figure 5-10 B et deuxième ligne en gras du Tableau 5.10). *Caliseq* détermine 4914 séquences comme étant des P450s, et ne détecte pas un total de 1871 séquences identifiées par les deux autres méthodes. Comme pour la première version de *Caliseq*, ces 4914 séquences ne contiennent aucun faux positif connu de Prosite, et comprennent également 28 séquences inconnues des deux autres méthodes. Ces 28 séquences sont montrées dans le Tableau 5.11.

Après vérification de ces séquences sur les autres serveurs (SMART ou ProDom) il s'avère que 100% sont des séquences de P450s. *Caliseq* se montre donc très sensible et très spécifique (mais peut être trop sélectif…) pour l'identification de séquences de P450s dans les banques de séquences.

5.5.3.3 Les faux négatifs ?

Une attention particulière a été portée aux séquences identifiées par les deux autres méthodes et non identifiées par *Caliseq*. Après examen de quelques unes manuellement, j'ai pu constater que ces séquences correspondaient pour la plupart à des fragments de séquences (moins de 200 résidus, ce qui est trop court pour être une P450). Ainsi, une statistique sur les longueurs de ces séquences (PFAM et Prosite) a été réalisée (cf. Figure 5-11)

Sur cette figure, on peut constater qu'un pic de P450 non reconnu par *Caliseq* est observé pour des séquences courtes de moins de 200 résidus. Par nature, ce genre de séquence n'est pas repérable par *Caliseq* : les blocs recouvrent environs 60 à 70% de la longueur totale d'un P450. Sachant qu'un P450 fait en moyenne une longueur de 400 à 500 résidus, la longueur minimale de séquences identifiables par *Caliseq* est donc d'environ 300 résidus. En effet, *Caliseq* n'est pas en mesure de positionner des blocs de séquences sur une séquence cible plus courte que la longueur totale des blocs. Ainsi, une grande partie des séquences situées en dessous de ce seuil limite n'est pas identifiable par *Caliseq*. Il en est de même pour les séquences trop longues en raison de la limite imposée par le paramètre *L*.

Ces séquences courtes n'ont pas de pertinence dans le cadre d'un projet de reconstruction de modèle 3D et d'une banque de sites actifs de CYP.

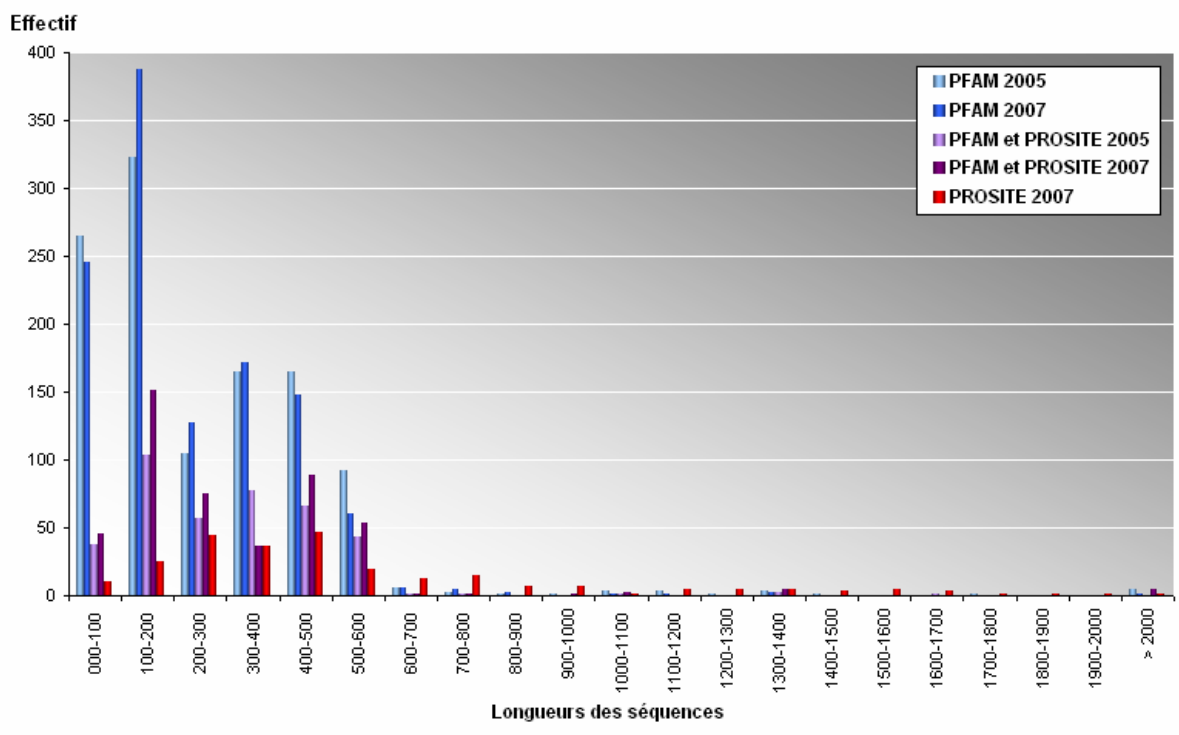


Figure 5-11 Statistique des longueurs de séquences non identifiées par *Caliseq* sur SptrEmbl2005 et SptrEmbl2007. Le jeu de blocs de N. Loiseau a été utilisé sur la SptrEmbl2005 avec une limite de distance de 1000 résidus, et le jeu de blocs issu de GAKUSA a été utilisé sur la SptrEmbl2007 avec une limite de distance de 1200 résidus. En bleu, le nombre de P450s identifiés seulement par Pfam et non trouvé par *Caliseq* ni Prosite, en mauve, le nombre de P450s identifiés uniquement dans Pfam et Prosite à la fois, et en rouge le nombre de P450s exclusifs à Prosite. Beaucoup de faux positifs pour *Caliseq* correspondent en fait à des fragments de P450s.

5.5.4 Essais complémentaires

À la suite de ces expériences de génomique exploratoire avec les CSBs identifiés par GOK ou GAKUSA, deux autres manipulations ont été effectuées.

La première a consisté à utiliser à la place des blocs de GOK ou GAKUSA, des « blocs » (autrement dit, les régions conservées de l'alignement) issus d'un alignement de Clustalw ou de Matras. Quels que soient les paramétrages utilisés, le nombre de séquences récupérées avec ces deux jeux de blocs n'ont pu dépasser celui obtenu avec le jeu ***E***. Néanmoins, il est toutefois possible de récupérer sur la SptrEmbl plus de 4000 séquences, ce qui n'est pas négligeable. À noter également que les blocs (MCSB) de Matras donnent de meilleurs résultats que ceux prélevés de Clustalw. L'information structurale contenue dans les blocs n'est donc pas négligeable dans la génomique exploratoire.

La deuxième expérience a consisté à utiliser le logiciel PSI-BLAST en utilisant les blocs pour construire le profil pour PSI-BLAST. La construction de cette matrice a de plus été bien pensée, dans la mesure où il est possible de restreindre la construction de la matrice à certaines positions et utiliser une matrice de similarité de type BLOSUM à d'autres positions de l'alignement fourni pour construire la matrice consensus. L'expérience a été lancée sur la base NR (Non Redondante). Deux inconvénients ont été trouvés pour cette méthode : i) la recherche des séquences ne dépend pas uniquement du profil, mais également (et surtout) de la séquence requête utilisée (la méthode est donc trop « requête dépendante » et la séquence « pivot » influence le résultat) et ii) beaucoup de faux positifs sont récupérés lors de cette recherche. PSI-BLAST, ne semble donc pas adapté pour utiliser l'information des CSBs en génomique exploratoire.

5.6 Conclusion et perspectives

Les résultats montrés au cours de ce chapitre ont souligné l'importance de l'utilisation de l'information structurale à la fois dans l'amélioration des alignements multiples de séquence pour une éventuelle reconstruction par homologie, et dans la recherche et la reconnaissance des séquences de protéines recherchées dans une banque de séquences.

Cette information structurale peut être exploitée sous forme de blocs structuraux conservés (CSBs) dans une approche qui associe justement la reconstruction de modèle protéique à bas taux d'identité, et la recherche sur banque par utilisation des CSBs comme signatures structurales spécifiques de la protéine dont sont issus les CSBs. Cette méthode à double fonction a su donner des résultats convenables et encourageants, dans ces deux champs d'études pour les P450s. Bien qu'étant conduites uniquement pour le moment sur la superfamille des P450s, il serait fort intéressant d'appliquer cette méthode sur d'autres familles de protéines à repliement conservé.

Les alignements obtenus suite aux positionnements des CSBs par *Caliseq*, ne semblent pas à première vue se démarquer de ceux obtenus par des méthodes plus classiques, ou plus complexes, compte tenu de l'état des connaissances actuelles. Néanmoins, placé dans le contexte historique, il s'agit pourtant d'une méthode fiable et innovante pour produire des alignements satisfaisants pour la construction de « bons » modèles lorsque l'information structurale, à savoir le nombre de structures qu'il est possible d'utiliser en *templates*, est limitée.

Concernant l'exploration génomique, les CSBs ont montré leur capacité à être utilisés en tant que signature structurale spécifique de P450s. Même si la méthode identifie moins de séquences que les méthodes homologues, elle montre en revanche une sensibilité et une spécificité très surprenante : aucun faux positif n'a été comptabilisé parmi les séquences identifiées par l'association des CSBs et de *Caliseq*. Par ailleurs, il a été également vu que seule une utilisation de l'information de type structurale est en mesure de proposer un tel résultat.

Par ailleurs, au cours de ces expériences d'exploration génomique, des phénomènes de palier ont été observés lors du filtrage par le z-score : des sous-ensembles de la banque sont en effet récupérés « par paquet » en fonction de l'ajustement du seuil de sélection. Il serait alors intéressant de vérifier si ces sous-ensembles sont identiques selon le jeu de blocs utilisé (qui sont fonction des *templates* qui le composent). Des expériences de clustérisassions pourraient peut être mettre en évidence une relation entre ces sous-ensembles et le jeu de blocs utilisés, conduisant au choix des meilleurs *templates* à utiliser pour la construction d'un modèle d'une cible donnée.

Cette méthode offrira donc une réelle innovation à partir du moment où la reconstruction par homologie, à savoir, toutes les étapes comprises entre le positionnement des blocs et l'obtention des modèles, sera effectuée de manière automatisée, sans intervention humaine. En effet, il deviendra alors envisageable d'exploiter en même temps les deux aspects différents et pourtant complémentaire de la méthode : en combinant efficacement l'exploration génomique à la reconstruction par homologie aux moyens des CSBs, il sera éventuellement possible de mettre en place une banque de modèles, voire seulement de sites actifs à disposition de la communauté scientifique. Une telle base peut servir d'outil pour des tests de pharmacologie et de prédiction d'interactions substrat-P450. En outre, par comparaison globale du devenir d'un même substrat utilisé sur tous les sites actifs de P450s de la banque, il sera peut être possible de fournir des éléments nouveaux dans la compréhension des mécanismes de reconnaissance des substrats.

CHAPITRE 6

Du virtuel au concret : applications sur le CYP2B6

« Le monde de la réalité a ses limites ;
le monde de l'imagination est sans frontières.»
Jean-Jacques Rousseau (1712 – 1778 ap JC)

6.1 Introduction

C'est au cours de ma seconde année de thèse que l'équipe de l'INSERM UMR517 (I. de Waziers et P. Beaune) en étroite collaboration avec P. Dansette (CNRS UMR8601), a fait appel à nos compétences en modélisation moléculaire. Cette équipe travaille dans le domaine de la pharmacologie dont les sujets d'études sont étroitement liés à la cancérologie et l'oncologie. L'une des méthodes expérimentales qu'ils ont développées, consistant à utiliser les CYP comme des activateurs de prodrogues, implique un cytochrome P450 non cristallisée à ce jour : le CYP 2B6. Ce cytochrome, comme la plupart de ses homologues impliqués dans la dégradation des composées exogènes, se retrouve principalement concentré au niveau du foie humain. Il est entre autre responsable du métabolisme spécifique d'un composé, le cyclophosphamide (CPA) dont le produit de dégradation (gaz moutarde) est cytotoxique pour les cellules en s'attaquant directement à leur ADN. En raison de ces propriétés cytotoxiques, le CPA est largement utilisé en tant qu'agent chimiothérapique, mais la difficulté majeure de son utilisation réside dans le non contrôle de son ciblage. En effet, lorsque le CPA est administré aux patients, ce dernier est dégradé au niveau du foie. Le produit de cette dégradation, très cytotoxique, se propage alors dans le corps à travers le flux sanguin. Il n'atteint pas que la tumeur, mais également des tissus non tumoraux. Afin de limiter son champ d'action, une stratégie particulière, la GDEPT (pour gene-directed enzyme prodrug therapy) a été adoptée. Celle-ci consiste à introduire dans un vecteur le gène d'un CYP 2B6, et à l'exprimer dans les cellules tumorales avant d'ajouter le CPA. Cette stratégie souffre toutefois d'un inconvénient majeur : l'affinité faible du CPA pour le CYP 2B6. La collaboration entre nos deux équipes avait donc pour objectif d'expliquer et d'améliorer l'affinité du CPA pour le CYP 2B6, par diverses approches prédictives de la modélisation moléculaire. L'équipe d'I. de Waziers nous a contactés pour produire un modèle fiable du CYP 2B6 afin de repérer les résidus du site actif candidats à des mutations susceptibles de modifier l'affinité du substrat à l'enzyme.

Disposant des outils nécessaires à leur traitement *in silico*, nous avons également proposé de construire virtuellement les mutants du CYP 2B6 et de conduire des simulations de dynamiques moléculaires sans contrainte de quelques nanosecondes, pour étudier, comprendre et expliquer le comportement du CPA dans le site nouvellement modifié. Cette approche a donné des résultats forts prometteurs, avec de bonnes convergences entre prédictions et expériences. L'ensemble de résultats fait l'objet d'un article en cours de soumission, ayant pour titre : « Analysis of CYP 2B6

cyclophosphamide activation by molecular modeling and site-directed mutagenesis : from *in silico* to *ex vivo*. ».

Dans ce chapitre, je ne décrirai pas toutes les techniques déjà présentées dans l'article. Au contraire, j'aborderai des points qui n'ont pas été détaillés, ou qui ont été réalisés après la rédaction de l'article.

6.2 Différentes approches exploitées pour un seul modèle final

La construction d'un modèle de CYP 2B6 était l'occasion d'expérimenter les différentes stratégies de reconstruction par homologie disponibles, et de les comparer à la méthodologie mise en place au laboratoire. À part SwissModel, qui propose un modèle final à partir de références de structures de P450s, les méthodes ne proposent que des alignements : c'est à l'utilisateur d'exploiter cet alignement, en le fournissant à des logiciels de reconstruction comme Modeller. Dans tous les cas, la première tâche consiste en la sélection des structures de référence.

6.2.1 Choix des structures de référence par différentes méthodes

La soumission de la séquence du CYP 2B6 à BLASTP sur les séquences des protéines contenues dans la PDB donne les résultats présentés au Tableau 6.1, par ordre d'identité (requête d'octobre 2005).

Tableau 6.1 Résultats de la recherche d'homologues du CYP 2B6 par un BLASTP sur la PDB

PDB	***Taux d'identité avec CYP 2B6***	***Recouvrement avec CYP 2B6***	***Nom du CYP***
1SUO / 1PO5	78%	351/447	CYP 2B4[1]
1Z11 / 1Z10	52%	234/447	CYP 2A6
1PQ2	50%	237/447	CYP 2C8
1DT6 / 1NR6 / 1N6B	49%	221/446	CYP 2C5
1OG5 / 1OG2 / 1R9O	48%	217/446	CYP 2C9
1TQN / 1W0G / 1W0F / 1W0E	25%	103/427	CYP 3A4
1Q5E / 1PKF	26%	44/164	$P450_{epok}$
1JPZ / 1BU7 / 2BMH / 1BVY	24%	49/203	$P450_{BM3}$

[1].1SUO et 1PO5 ne présentent pas tout à fait les mêmes valeurs, dans la mesure où 1P05 comporte des résidus non résolus. Par ailleurs, 1PO5 est une forme sans substrat et ouverte.

Dans ce tableau, les structures des CYP 2B4 présentent une identité de 78% avec le CYP 2B4. A ce niveau d'identité, il est admis qu'une reconstruction en *monotemplate* (une seule structure de référence utilisée) est suffisante, voire même fortement conseillée. Comme mon idée était de tester les

différentes méthodes, j'ai quand même entrepris l'essai des autres méthodes, juste par curiosité : allaient-ils me donner un modèle équivalent à celui obtenu en *monotemplate* ?

Les structures de référence et les méthodes employées pour aligner les séquences des structures de références sur la séquence cible ont été les suivantes :

a) Pour la reconstruction du CYP 2B6 en monotemplate, la forme fermée de la structure du CYP 2B4 a été choisie (1SUO) : comme détaillé dans l'article, l'avantage que présentait la structure 1SUO du CYP 2B4 est de comporter un substrat, le 4-(4-chlorophenyl)imidazole (CPI) co-cristallisé dans son site actif. De plus, ce dernier étant structuralement proche du CPA, sa position dans le site actif du CYP 2B4 pouvait fournir un bon point de départ pour le positionnement du CPA. L'alignement quant à lui est aisément réalisé à l'aide du logiciel Clustalw.
b) Le serveur SwissModel a choisi pour sa part, les structures 1SUO, 1PO5, 1Z11 et 1Z10 (respectivement CYP 2B4 et CYP 2A6) pour construire le modèle. Il est en effet possible d'imposer ses propres structures de référence, mais par défaut, SwissModel effectue un BLASTP et choisit les meilleures séquences des structures présentant la meilleure identité avec la séquence cible soumise.
c) Sous le serveur Matras (cf. 2.4.4.3), j'ai fourni le même set de structures que celui de SwissModel : Matras se charge alors de retourner un alignement de ces quatre structures de référence entre eux. Il me reste alors à aligner les séquences de ces quatre structures sur la séquence du CYP 2B6. La réalisation de cet alignement est triviale dans la mesure où la séquence du CYP 2B4 est très proche de celle du CYP 2B6.
d) Un autre serveur, HHPred (Söding et *al.*, 2005) a été utilisé à l'occasion. Pour mémoire, ce serveur utilise un programme basé sur les HMM pour réaliser l'alignement de séquences en recherchant les homologues du CYP 2B6. L'avantage que présente ce serveur est de fournir en résultat un alignement pré formaté pour le logiciel Modeller : il ne reste alors plus qu'à préparer le fichier de commande. Les structures retenues en référence par HHPred sont le 1X8V (CYP 51), 1PO5 (CYP 2B4), 1R9O (CYP 2C9), 1Z10 (CYP 2A6), 1IZO ($P450_{BSbeta}$), 1UE8 ($P450_{st}$), 1IO7 (CYP 119), 1N4O (CYP 121), IUED ($P450_{OxyC}$) et 1LFK ($P450_{OxyB}$). Mises à part les structures des CYP 2C9 et du CYP 2A6, toutes les autres structures sont celles de bactéries. Ce résultat est assez surprenant dans la mesure où HHPred a proposé pour *templates* de nombreuses structures en dessous du seuil d'identité de 26%, indiqué par le BLASTP (non présentées dans le Tableau 6.1).

e) Enfin, la méthode de blocs structuraux, proposée au laboratoire a aussi été testée, en n'utilisant que mon jeu de blocs, qui contient un plus grand nombre de P450s microsomaux. En fait, les trois jeux de blocs ont été testés, et mon jeu de blocs a été celui qui a fourni les résultats les plus convaincants au niveau de l'alignement. Il est à noter que le jeu global de blocs n'a été instauré que vers la fin de ma deuxième année de thèse, je n'ai pas pu le tester à ce moment-là. Pour mémoire, mon jeu de blocs possède six structures : 1OXA, ($P450_{eryF}$), 2HPD ($P450_{BM3}$), 1PQ2 (CYP 2C8), 1OG5 (CYP 2C9), 1NR6 (CYP2C5) et 1GWI (CYP 154C1).

6.2.2 Choix de la méthode finale

6.2.2.1 Départage sur l'alignement

De manière surprenante, les alignements sont très proches, et ce, en dépit de la diversité des *templates* utilisés – certaines méthodes incluent des structures bactériennes à taux d'identité faible par rapport au CYP 2B6 –. Une explication possible de cette similitude entre les alignements, est la présence d'au moins une séquence de la famille des 2C dans chacun des jeux de structures pour chaque méthode. En effet, l'identité de séquences pour les structures de cette sous-famille est forte (~50%) vis-à-vis du CYP 2B6 comme le montre le Tableau 6.1. En conséquence, la présence de ces structures a permis un « ancrage » favorable des séquences des *templates* sur la séquence du CYP 2B6, aboutissant à un alignement similaire pour toutes les méthodes. Il est à noter que cette observation de « similarité » d'alignement n'a pu être effectuée que sur les séquences de structures communes à toutes les méthodes, qui sont généralement les structures de P450s microsomaux.

6.2.2.2 Départage sur le modèle 3D

La différence entre les méthodes s'observe davantage pour les modèles obtenus : le logiciel Modeller prenant en compte toutes les contraintes spatiales des *templates*, le résultat final dépend fortement des structures utilisées comme références. Comme celles-ci diffèrent d'une méthode à l'autre, il n'est donc pas étonnant d'obtenir des modèles divergents. Le meilleur modèle, selon la fonction objective de Modeller a été évalué pour chaque stratégie par des logiciels d'évaluation de structures RX (Prosa II, ProQ et Anolea). Tous les modèles générés répondent correctement aux critères d'évaluation. Les RMSDs de ces meilleurs modèles calculés sur le squelette peptidique, ont été calculés par rapport au modèle du CYP 2B6 (qui sert ici de référence) obtenu en monotemplate (Tableau 6.2).

Tableau 6.2 RMSD calculé sur le backbone entre les modèles de CYP 2B6 générés par les différentes méthodes d'alignement. La structure de référence pour le calcul de RMSD est le modèle obtenu en monotemplate. Pour information, le modèle du CYP 2B6 monotemplate a un RMSD de 1,77 Å de la structure RX du CYP 2B4

Modèle obtenu par la méthode d'alignement	RMSD avec le modèle obtenu en monotemplate
Matras	0,38 Å
SwissModel[1]	0,63 Å
Caliseq	1,85 Å
HHpred	2,69 Å

[1] SwissModel est la seule méthode entièrement automatisée, pour les 3 autres, nous avons utilisé Modeller 7

Sur ce tableau, les deux modèles qui semblent diverger le plus par rapport à celui construit à partir de 1SUO sont ceux générés par l'alignement HHPred et Caliseq. Encore une fois, ce résultat n'est pas surprenant : les méthodes HHPred et Caliseq comportent en effet plus de structures de référence bactériennes, alors que les autres n'en disposent pas. À constater également qu'à partir d'un jeu identique de *templates*, la méthode Matras et SwissModel ne donnent pas tout à fait le même résultat : le programme de reconstruction utilisé est différent (Modeller pour Matras, PromodII pour SwissModel), mais la différence n'est pas significative. D'ailleurs, à une simple minimisation près, les modèles obtenus pour SwissModel, Matras et en *monotemplate* sont équivalents.

Le choix final de la méthode pour reconstruire le CYP 2B6 s'est porté sur la stratégie en *monotemplate* pour les raisons évoquées précédemment (~78% d'identité en séquence). Les autres constructions ont été réalisées en vue de comparer les différentes méthodes. À noter enfin que de toutes les méthodes, la méthode de recontruction par blocs structuraux est la plus défavorisée pour l'alignement : elle est la seule qui ne comporte pas la séquence d'un CYP 2B4. Toutefois, ce désavantage est compensé par la structure du CYP 2B4 utilisée par les autres méthodes comme *templates* pour la reconstruction : 1PO5 est une forme ouverte du CYP 2B4, à 5 Å de RMSD de la structure du 1SUO –le RMSD est calculé sur les atomes des squelettes peptidiques–. Utiliser cette forme ouverte du CYP 2B4 pourrait ainsi expliquer (en plus du nombre de structures bactériennes utilisées) la divergence avec le modèle construit en *monotemplate*.

6.2.2.3 Obtention du modèle final

Comme expliqué plus en détail dans l'article, le modèle *monotemplate* du CYP 2B6 a été généré à partir d'une structure minimisée du CYP 2B4 1SUO et relaxée par 2 ns de dynamique moléculaire libre. Cette opération de pré-reconstruction a été réalisée dans le but de limiter les défauts structuraux

qui pourraient être présents sur la structure X initiale. Cette opération peut sembler inutile, dans la mesure où Modeller est censé relaxer les structures *templates* avant d'utiliser leurs contraintes spatiales. Mais la relaxation par MD pendant une longue durée (2 ns) et dans une boîte de solvatation explicite permet de rechercher une meilleure structure à l'équilibre. Cette étape permet en outre de préparer le protocole GROMACS qui sera appliqué sur le modèle final du CYP 2B6.

Le modèle, une fois construit et validé par les logiciels d'évaluation de structures (Prosa II, ProQ et Anolea), a été comparé à la structure du CYP 2B4. De l'évolution des structures lors de la dynamique, aux canaux d'accès suggérés par le logiciel CAVER (Petrek et *al.*, 2006) en passant par le volume du site actif déterminé par le logiciel Voidoo (Kleywegt et *al.*, 2006), tout a été passé en revue comme cela est décrit dans l'article. Le but de ces manœuvres était de vérifier si le modèle du CYP 2B6 n'était pas une simple copie de la structure du CYP 2B4 et d'observer de vraies variations dans le site actif. Le RMSD final entre le modèle du CYP 2B6 relaxé et équilibré et la structure du CYP 2B4 est de 1,77 Å, ce qui suggère que ces deux protéines sont structuralement distantes.

6.3 Docking manuel et mutations *in silico*

La seconde étape a consisté à placer le substrat dans le site actif du CYP 2B6 afin de déterminer les résidus en contact étroit avec le CPA. Ceux-ci seront alors proposés à des mutations en vue de modifier le comportement du CPA envers son enzyme.

6.3.1 Docking du CPA

6.3.1.1 Obtention du fichier PDB du CPA

Ne disposant pas de structure cristallographique du CPA, il a fallu le construire, en partant de sa simple formule chimique (cf. Figure 6-1) récupérée à partir du serveur de composés chimiques ChemIdPlus (http://chem.sis.nlm.nih.gov/chemidplus/). Cette formule a été ensuite soumise au serveur Dundee PRODRG2 (http://davapc1.bioch.dundee.ac.uk/programs/prodrg/) (Shuettelkopf et Aalten, 2004) qui fournit à la fois un fichier de topologie exploitable par GROMACS ainsi que le fichier de structure du CPA au format PDB.

Le composé est chiral au niveau du phosphore, or PRODRG ne propose qu'un seul des deux énantiomères : le S. Sachant que l'équipe d'I. de Wazier utilisait un mélange racémique pour les mésures d'activités, il fallait donc construire son énantiomère R pour disposer d'une reproduction

complète des données expérimentales en machine. Pour cela, j'ai utilisé le logiciel SYBYL (Tripos inc, St Louis, USA) (fonction « *invert chirality* »).

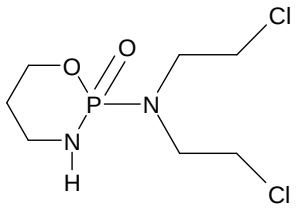


Figure 6-1 Formule chimique du Cyclophosphamide (CPA), chiral au niveau du P.

6.3.1.2 Positionnement du CPA dans le site actif

La procédure de positionnement du CPA dans le site actif du CYP 2B6 est expliquée dans l'article, dans la partie Materials and Methods ainsi que dans la partie Results « Initial positionning of CPA ». Brièvement, j'ai donc profité du fait que le CPI, une molécule structuralement proche du CPA, a été co-cristallisé avec le CYP 2B4 dans 1SUO. Après superposition des modèles, j'avais donc un point de départ pour positionner le CPA : les deux atomes faisant face à l'hème pour les deux molécules ont été superposés grâce à un script TCL sous VMD[18]. Ce placement du CPA dans le site actif du CYP 2B6 ne s'effectue pas sans conflit stérique, c'est pourquoi un léger réajustement manuel suivi d'une minimisation locale a été nécessaire. Les paramètres de la minimisation locale sont décrits dans l'article, tandis que le réajustement a nécessité l'utilisation d'un autre logiciel de visualisation : SPdbViewer (http://www.expasy.org/spdv/). Ce logiciel offrait en effet l'avantage de figer la protéine et permettait de ne manipuler que le substrat par utilisation de deux fichiers distincts pour chacune des entités. Il existe toutefois un inconvénient à manipuler deux fichiers indépendants sous SPdbViewer : le logiciel ne peut pas tenir compte des interactions entre les atomes des deux fichiers. C'est pourquoi, lorsqu'une position de CPA semblait correcte, les coordonnées cartésiennes du CPA étaient enregistrées et traitées sous SYBYL qui se chargeait de la minimisation locale du CYP 2B6 sur les résidus avoisinant le CPA, hème exclu. Il est à noter que la dernière partie de la procédure, la minimisation locale autour du CPA, a été réalisée pour chaque mutant du CYP 2B6 construit ultérieurement.

[18] VMD est un logiciel de visualisation de structures, qui peut recevoir des directives au moyen de scripts écrits en langage TCL. C'est l'outil dont je me suis le plus servi pour visualiser mes résultats, mais pas pour générer les images : elles ont été réalisées sous PYMOL.

6.3.2 Mutations *in silico*

Le placement du CPA dans le site actif du CYP 2B6 a permis de repérer les résidus en contact étroit avec le CPA (cf. article). Ces derniers ont été alors proposés pour des expériences de mutations ponctuelles, en vérifiant préalablement si ces mutants avaient déjà été signalés dans la littérature. Les premiers mutants opérés par l'équipe d'I. de Waziers ont porté principalement sur le remplacement des résidus proposés par des résidus à caractère polaire : I. de Waziers et coll. pensaient en effet exploiter le caractère polaire du groupement P=O du CPA en remplaçant des résidus hydrophobes du site actif par des résidus polaires. Ces résultats n'ont pas été convaincants comme on peut le constater sur le Tableau 6.3, dérivé du Tableau 5 de l'article avec les caractéristiques des résidus en plus. Avec l'appui du modèle du site actif, d'autres combinaisons ont été essayées portant sur le caractère « encombrement stérique» des résidus. Trois mutants (V477F, I114V, V477W) ont révélé des résultats intéressants au niveau de l'affinité (Km) et de la vitesse de réaction (Vmax). Ces trois mutants ont été testés *in silico* afin d'observer le comportement du CPA placé à l'intérieur. L'analyse des simulations des trois mutants a conduit à la réalisation d'un double mutant (I114V/V477W) qui a montré en *ex vivo* des résultats très encourageants, interprettables par le comportement du substrat dans les simulations de MD *in silico*.

Pour la réalisation de chaque mutant, le modèle du CYP 2B6 vide, équilibré et relaxé par 2 ns de simulation de MD a été repris comme point de départ, et soumis au logiciel SYBYL. Pour chaque mutant, une série de rotamères pour les chaînes latérales est proposée : ceux-ci résultent d'une observation statistique de rotamères dans la PDB. Ainsi, on choisira le rotamère le plus « naturel » à savoir celui dont la fréquence est la plus importante dans la PDB. Afin de supprimer les « bumps » éventuels (ou contacts non désirés) survenus lors de la mutation, une minimisation locale est appliquée. Le CPA est ensuite soumis à une minimisation locale, toujours conduit sous SYBYL. Suite à cet ajustement, une simulation de 2 ns de MD a été réalisée. Les résultats de la simulation sont ensuite analysés.

Tableau 6.3 Analyse cinétique du métabolisme du CPA par les levures exprimant les CYP 2B6 wt et mutant. Les mutants sont répartis en trois groupes : polymorphisme, mutations canines et mutations du site actif des résidus prédits *in silico*. (source : I. de Wazier)

Mutation	Comparaison par rapport au CYP2B6wt			Vmax		Km		Vmax/Km	
	sauvage	muté	taille de la chaîne latérale	*min^{-1}*	*% of wt*	mM	*% of wt*		*% of wt*
CYP 2B6wt				63	100	5	100	13	100
Q172H				33	*53*	4	86	8	*62*
F107V			-	96	**153**	6	*133*	15	**115**
L199M			+	125	**200**	5	98	26	**205**
S207A				70	112	6	113	13	99
K236N				47	*75*	56	*1153*	1	*7*
M365I			-	13	*21*	4	**77**	3	*27*
C475I				2	*3*	5	96	0	*3*
I114V			-	65	104	2	**47**	28	**221**
G366V			+	1	*2*	NQ	-	-	-
G366E +V367L			+	NQ	-	NQ	-	-	-
V367L			+	90	**144**	9	*181*	10	*80*
V367F			++	27	*43*	13	*264*	2	*16*
V367S				5	*8*	NQ	-	-	-
V367T				24	*39*	6	*131*	4	*30*
V367H				2	*2*	NQ	-	-	-
V477S			-	NQ	-	NQ	-	-	-
V477T				37	*59*	9	*182*	4	*33*
V477Y			++	29	*46*	6	*121*	5	*38*
V477N			+++	NQ	-	NQ	-	-	-
V477D				3	*4*	NQ	-	-	-
V477E				2	*3*	NQ	-	-	-
V477I			++	77	**124**	3	**66**	24	**189**
V477F			++	81	**130**	3	**68**	25	**191**
V477W			+++	100	**160**	3	**57**	36	**278**
G478A			+	28	*45*	6	*129*	4	*35*
G478V			++	36	*58*	3	**53**	14	**110**
G478S				7,51	*12,0*	NQ	-	-	-
G478E				5,4	*8,6*	NQ	-	-	-
I114V +V477W			- / +++	58,47	93,6	1,11	**22,793**	52,6757	**411**

Hydrophobe
Polaire, non chargé
Polaire, chargé négativement
Polaire, chargé positivement

NQ: non quantifiable
+ / - Taille de la chaîne latérale par rapport au wt

Les mutations conduisant à une meilleure activité catalytique sont indiquées en grisé. À l'inverse, ceux conduisant à une diminution de l'activité enzymatique sont indiqués en italique.

6.4 Simulation de MD

Les simulations de dynamique moléculaire ont été réalisées sous deux champs de force différents : GROMOS87 pour toutes les simulations présentées dans l'article, et AMBER7 pour quelques simulations postérieures à l'article. L'idée sous-jacente était de vérifier la conformité des résultats selon deux méthodes différentes. Le choix de GROMOS87 (ou plus exactement, GMX, un champ de force dérivé de GROMOS87) a été guidé d'une part par l'utilisation du serveur PRODRG2 qui fournissait des topologies pour logiciel GROMACS exclusivement, et d'autre part par le fait que GROMACS incorporait la topologie de l'hème et permettait sa fixation à l'enzyme sans instructions supplémentaires particulières. La différence majeure entre les deux champs de force vient de leur traitement des atomes d'hydrogène : le champ de force GROMOS est de type « tout uni » et non tout atome, accélérant énormément les temps de calculs. Ainsi, en réalisant des simulations avec NAMD, qui utilise le champ de force AMBER (tout atome), je voulais vérifier si les résultats obtenus sous GROMACS étaient reproductibles.

6.4.1 Utilisation de GROMACS

6.4.1.1 Procédure pour les complexes substrat-enzyme

Pour réaliser la dynamique moléculaire du CYP 2B6 (de l'un de ses mutants, ou du CYP 2B4) avec son substrat placé dans son site actif, une procédure en sept étapes, dérivée du tutorial de J.E. Kerrigan (http://www2.umdnj.edu/~kerrigje/pdf_files/trp_drug_tutor.pdf), a été adoptée :

1. Création des deux fichiers GROMOS pour l'enzyme (un fichier *gro* contenant les coordonnées cartésiennes des atomes de la protéine et un fichier *top* qui décrit toute la topologie de la protéine, à savoir distances, angles etc.) et ligation de l'hème par la fonction *pdb2gmx*.
2. Création des deux fichiers GROMOS pour le substrat par requête au serveur PRODRG2 (on un fichier *gro* de coordonnées cartésiennes et un fichier *itp*, de similaire au fichier *top* de l'enzyme sont alors récupérés).
3. Renumérotation manuelle du fichier *gro* du substrat puis incorporation en fin de fichier *gro* de l'enzyme. Utilisation de la commande *#include* dans les fichiers *top* de l'enzyme pour prendre en comptes ceux du substrat (fichier *itp*).
4. Création de la boîte d'eau par solvatation explicite et neutralisation des charges par des contre-ions (de 1 à 3 dans le cas des CYP 2B6, et 4 dans le cas du CYP 2B4).
5. Minimisation du système (protéine, substrat, solvant et contre ions). La minimisation du système s'arrête lorsqu'il y a convergence du système (Fmax < 1000) ou lorsqu'un nombre de pas fixé par l'utilisateur est atteint (5000 pas).

6. Equilibration du système (20 ps) pour permettre à l'eau de bien diffuser.
7. Dynamique Moléculaire du complexe enzyme-substrat solvaté avec ou sans contraintes.

L'avantage de cette méthodologie est multiple : il n'y a pas besoin de créer le fichier de topologie pour le substrat, puisque c'est le serveur PRODRG qui s'en occupe, ni de définir la topologie de l'hème déjà présente dans le champ de force GMX, ni de l'attacher au CYP 2B6 car la ligation est réalisée automatiquement lors de l'utilisation du script *pdb2gmx*. Néanmoins, dans cette stratégie, la commande *#include* présente l'inconvénient de traiter l'enzyme et le substrat dans deux fichiers séparés : l'application d'une contrainte de distance entre l'hème présent dans un des fichiers de topologie et le substrat présent dans l'autre fichier de topologie n'était pas réalisable, car le programme ne peut pas gérer les interactions entre les deux systèmes de coordonnées. L'utilisation de contraintes de distance est pourtant intéressante lorsque l'on désire favoriser une interaction particulière (par exemple maintenir le CPA dans l'environnement de l'hème). Sous GROMACS, cette contrainte de distance correspond en fait à une pénalité énergéique ajoutée à l'énergie potentielle, lorsque la distance entre deux atomes d'une paire spécifiée excède un certain seuil. Pour pouvoir utiliser les contraintes de distances dans mes simulations, j'ai été amené à modifier le protocole de J.E. Kerrigan, en réalisant une fusion des informations topologiques du CPA et de celles l'enzyme, dans un seul et même fichier de topologie. Deux scripts python ont été écrits à cet effet : *Addgro2gro.py* permet la renumérotation et la fusion des fichiers *gro*, *Additp2top.py* permet de fusionner le fichier *itp* de topologie du substrat au fichier *top* de topologie de l'enzyme. Non seulement la troisième étape de la stratégie devenait désormais automatisée, mais aussi les contraintes de distances ont pu être appliquées.

Les différents paramètres utilisés pour la minimisation, l'équilibration et la dynamique sont détaillés dans l'article.

6.4.1.2 Résultats et analyses des simulations

Un total de 25 simulations a été réalisé selon le nouveau protocole établi, comprenant également celles du CYP 2B4 en présence et en absence de son substrat. Pour la plupart des mutants, deux types de simulations ont été calculées : un premier essai de 500 ps pour vérifier si la dynamique s'exécutait sans erreur, puis une autre simulation exhaustive de 2 ns. Pour le CYP 2B6wt, et les mutants V477F et V477W, des essais de contraintes de distances appliquées entre le carbone à oxygéner et le fer de l'hème ont été réalisés. Nous n'avons pas pu observer une différence avec la simulation sans contrainte, ce qui suggère que le substrat est dans une position d'équilibre favorable dans la poche du

site actif. Enfin, pour des contraintes de temps de calcul, j'ai réalisé le docking des deux énantiomères uniquement pour le CYP 2B6wt, le double mutant I144V/V477W et le mutant V477Y. Pour ce dernier mutant, il s'agissait d'expliquer par la dynamique la différence observée lors du remplacement de la valine en position 477 par un résidu encombrant : une meilleure efficacité catalytique est observée lors d'un remplacement par une phénylalanine ou un tryptophane, tandis que le remplacement par une tyrosine entraîne une chute de l'activité (cf. Tableau 6.3). Pour chaque simulation, la trajectoire de celle-ci est alors observée puis analysée.

Outre l'observation de la trajectoire de la simulation sous VMD, le logiciel GROMACS offre une panoplie d'outils d'analyse des simulations, permettant de détecter d'éventuelles anomalies énergétiques au cours de la dynamique, ou vérifier si la durée de cette dernière est suffisante pour que le système atteigne une position d'équilibre. Ces outils permettent également d'analyser des distances, ou l'évolution des liaisons hydrogène créée entre le substrat et l'enzyme au cours de la trajectoire.

Stabilité du modèle. Toutes les simulations sous GROMACS ont été construites dans l'ensemble NPT (Nombre de molécule constant, Pression constante et Température constante). Pour s'assurer du respect de ces conditions, la fonction *g_energy* de GROMACS a été utilisée pour fournir des renseignements sur la pression, température, l'énergie du système à chaque « frame » (ou temps[19]) de la simulation. En dehors de l'énergie potentielle qui évolue logiquement selon la qualité du modèle du complexe de départ, les courbes sont similaires pour toutes les simulations réalisées.

[19] Dans les simulations de MD conduites sous GROMACS, chaque frame correspond à 2 femtosecondes.

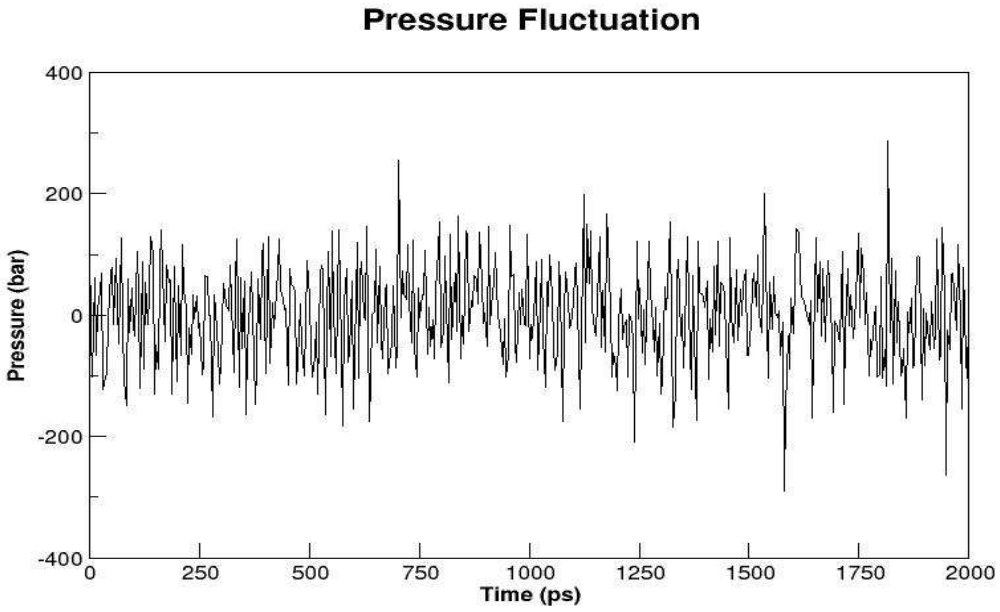


Figure 6-2 Exemple de contrôle de la pression durant la dynamique du double mutant CYP 2B6 (I114V/V477F). La courbe de pression montre une oscillation normale du comportement de la pression autour de 1 bar. Pour plus de clarté, un frame sur 20 est montré.

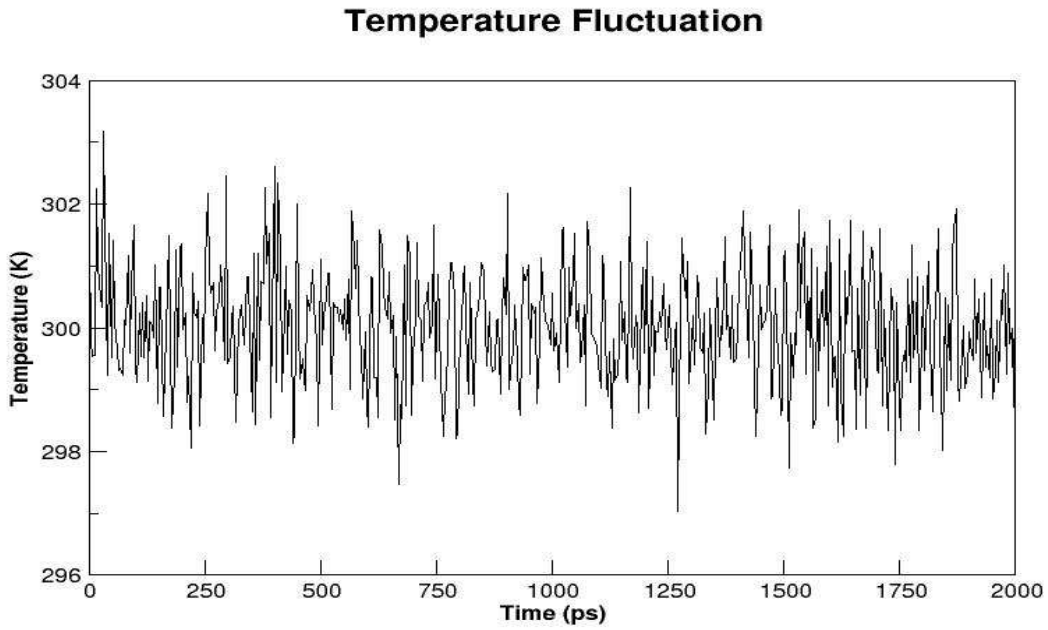


Figure 6-3 Exemple de contrôle de la température durant la dynamique du double mutant CYP 2B6 (I114V/V477F). La courbe de température montre une oscillation normale du comportement de la température autour d'une moyenne de 300K, la température de consigne.

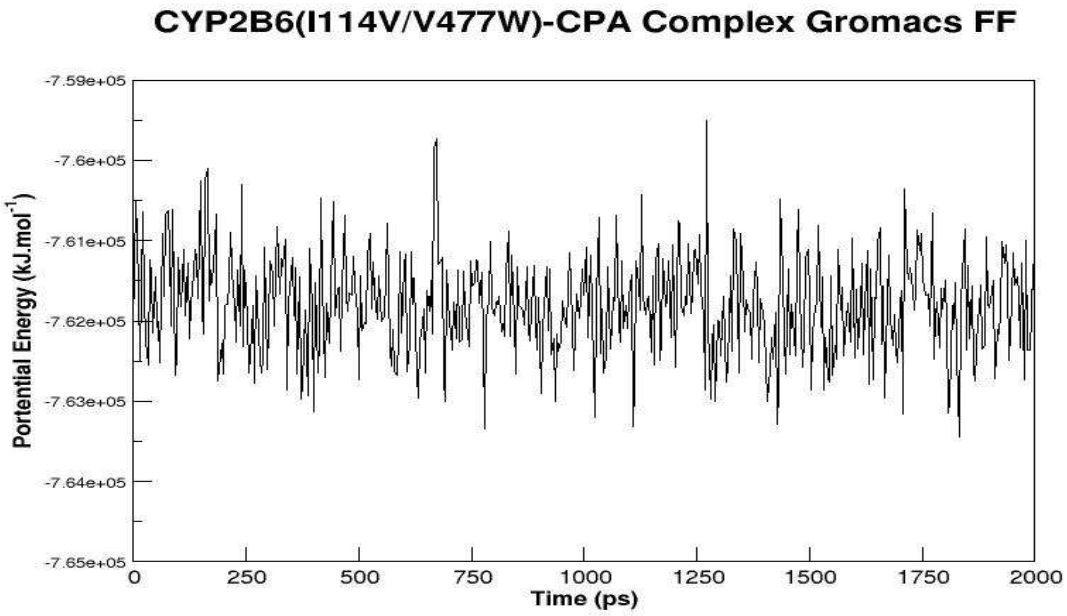


Figure 6-4 Évolution de de l'énergie potentielle pour le complexe double mutant (I114V/V477W) / CPA. Dans le cas présent, l'équilibration du système a été obtenue rapidement (~ –761 500 kJ/mol).

Mouvements des atomes durant la simulation. Pour contrôler les fluctuations du squelette du modèle ainsi que de son substrat, la fonction *g_rms* a été utilisée. Celle-ci calcule à chaque temps, le RMSD entre le squelette peptidique du modèle (ou les atomes lourds du CPA) à un temps *t* et sa (leurs) position(s) à l'instant initial t_0. L'ensemble de ces RMSD forme une courbe comme celles montrées en figures 1 et 7 de l'article. Ces courbes permettent de voir à partir de quel moment le modèle a atteint un point d'équilibration, et si la dynamique est suffisamment longue. La plupart du temps, cette position d'équilibre était atteinte aux environs de 500 ps. Les premières simulations atteignent cette « limite » des 500 ps : elles ont donc été étendues à 2 ns (soit 4 fois le temps d'atteinte à l'équilibre). Par ailleurs, la courbe de RMSD du CPA (figure 7 de l'article) rend compte des mouvements du CPA, observés lors de la simulation. En effet, au cours de la trajectoire dynamique, il a été observé que le CPA était moins mobile dans le site du double mutant, ou dans le site du mutant V477F par rapport au CYP 2B6wt. L'évolution du RMSD au cours du temps est un moyen commode de rendre compte des mouvements observés sur la vidéo.

Liaisons H formées entre le substrat et l'enzyme. Afin de connaître les interactions impliquées entre le CPA et le CYP2B6, nous avons cherché à caractériser le nombre de liaisons hydrogènes formées entre les deux molécules au cours de la dynamique. La fonction *g_hbond* dans GROMACS a

permis cette analyse. Par défaut, cette fonction prend une distance de r_{DA} = 2,5 Å entre l'atome donneur D et l'atome accepteur A, ainsi qu'un angle de α=60° autour des atomes pour trouver leurs partenaires potentiels (cf. Figure 6-5). Étonnamment, aucune liaison H n'a été observée (même transitoirement) entre le CPA et le CYP2B6 quelle que soit la dynamique et quel que soit le mutant considéré, et ce même en modifiant les paramètres par défaut du programme. Ce résultat peut expliquer l'absence d'effets positifs sur l'affinité pour le CPA dans la série des différents mutants testés sur la base de la polarité.

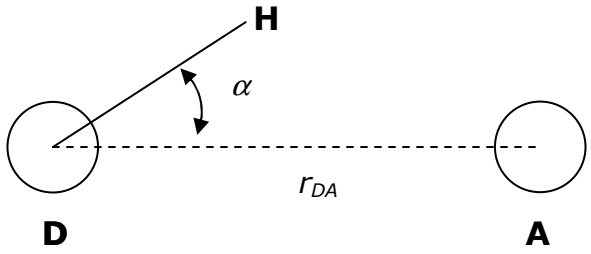


Figure 6-5 Relations géométriques utilisé par g_hbond. D correspond à l'atome Donneur et A à l'atome Accepteur. H représente l'atome d'hydrogène. Par défaut, la distance entre les deux atomes est réglée à r_{DA} = 2,5 Å et α = 60°.

Recherche des interactions dans le site actif. Une autre fonction de GROMACS qui a été très utile lors de l'étude d'interaction CPA-CYP 2B6, est la fonction *g_dist*. Cette dernière permet de calculer la distance entre deux atomes tout le long de la dynamique. Cette information de distance s'est révélée intéressante lorsque j'ai cherché à connaître l'évolution de la distance entre l'atome d'oxygène du groupe P=O du CPA avec les résidus polaires les plus proches dans le site actif. Même si aucune liaison H ne se formait, il y avait peut-être une interaction possible avec les résidus avoisinant en dehors de la répulsion stéarique exercée par la mutation de la valine 477 par un résidu « encombrant ». Sous VMD[20], cinq résidus polaires intéressants ont été trouvés à environ 10 Å du CPA : la serine 294, le glutamate 301, la thréonine 302 et les phénylalanines 206 et 297 (cf. Figure 6-6). L'évolution de ces distances est montrée sur la Figure 6-7.

[20] La Figure 6-6 a été obtenue à l'aide du logiciel Pymol.

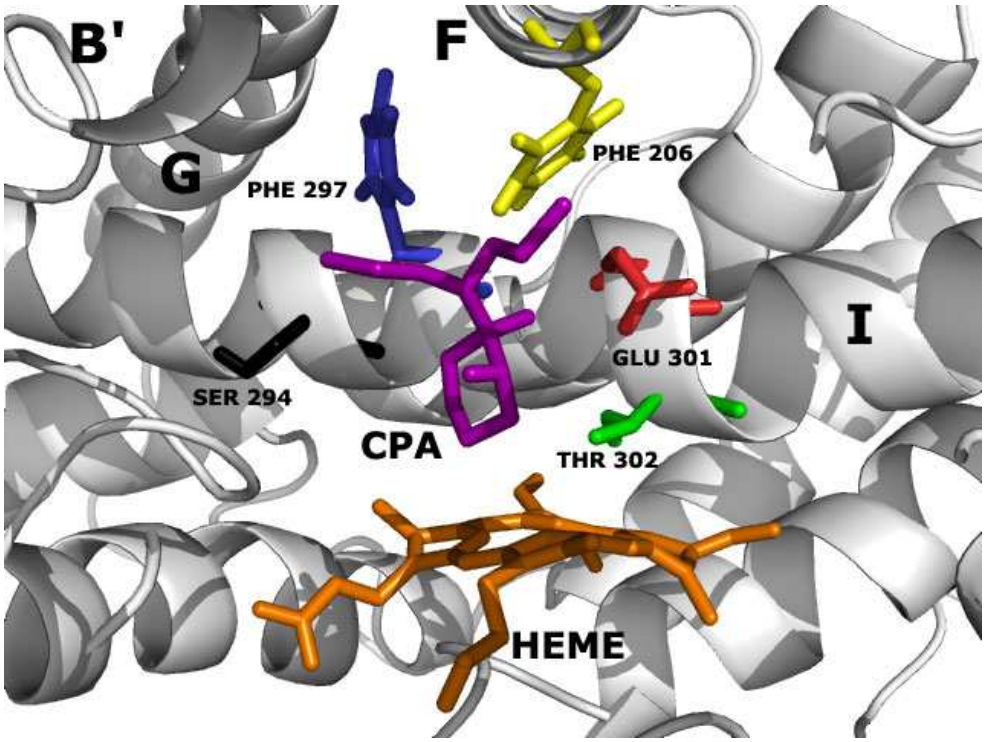


Figure 6-6 Représentation du site actif avec les 5 résidus à 10 Å du groupe P=O du CPA : Ser294 (noir), Glu301 (rouge), Thr302 (vert), Phe206 (jaune) et Phe297 (bleu). Les résidus, l'hème et le CPA sont représentés en bâtonnet, tandis que le reste de la protéine est représenté en rubans (logiciel Pymol). Les hélices I, G, F et B' sont mentionnées à titre indicatif.

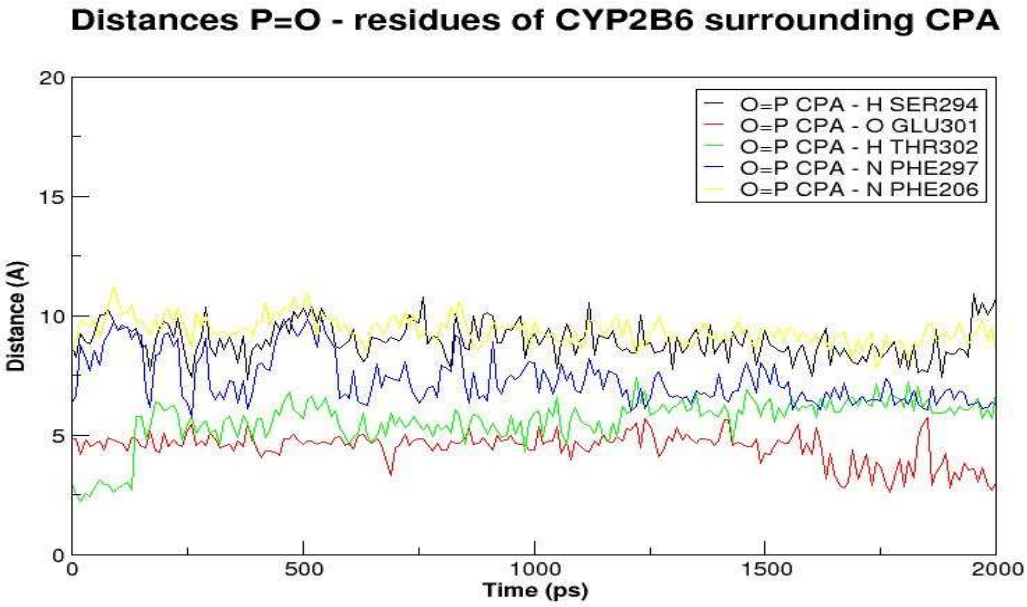


Figure 6-7 Distances entre l'atome d'oxygène du groupe P=O du CPA avec un atome des résidus avoisinants : Ser294 (noir), Glu301 (rouge), Thr302 (vert), Phe206 (jaune) ou Phe297 (bleu).

Cette information nous a permis de révéler l'existence d'un contact polaire (défavorable) entre le groupe P=O du CPA et l'un des oxygènes du carboxylate du résidu Glu301. Comme montré sur la Figure 6-7, la distance entre les atomes d'oxygène reste relativement constante durant toute la simulation, et tend même à diminuer en fin de simulation. La présence de ce contact polaire défavorable est liée au contexte environnemental du Glu301 qui se retrouve encapsulé entre des résidus à caractère hydrophobe (cf. article, section discussion). C'est pourquoi, l'atome d'oxygène du P=O s'oriente plus volontiers vers le Glu301, plus favorable que son voisinage. Cette interaction peut par ailleurs être favorisée en présence d'une molécule d'eau au voisinage (cf. Figure 6-8), qui n'a malheureusement pas été observée au cours de la dynamique.

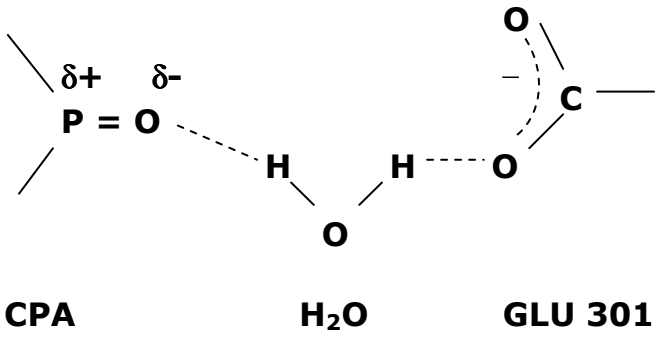


Figure 6-8 Interaction CPA – GLU301 favorisé par la présence d'une molécule d'eau. Les formules du CPA et du GLU 301 sont partiellement représentées. Les interactions sont représentées par des traits en pointillés.

En résumé, GROMACS a été utilisé pour les expériences de MD dans l'article, en raison de sa facilité d'utilisation, sa rapidité d'exécution par rapport aux autres outils disponibles, et aussi pour l'ensemble des outils qu'il propose pour l'étude des simulations. Néanmoins, GROMACS reste sensible à la stabilité du modèle de départ. Le programme peut diverger à la suite de « clashs » survenant lors des processus d'équilibration ou de dynamique moléculaire, et ce, en dépit d'une minimisation préalable.

Souvent, il a fallu modifier légèrement la position initiale du CPA dans le site actif avant de pouvoir relancer correctement l'ensemble de la dynamique. J'avais d'ailleurs écrit un script pour générer différentes positions initiales pour le substrat : mais l'étroitesse du site rend cette approche souvent non productive, il n'a que peu de marge possible dans le placement initial du substrat dans le site actif, non rattrapable dans GROMACS.

6.4.2 Utilisation d'AMBER7 sous NAMD

Dans le but de vérifier la pertinence et la reproductibilité des résultats de simulation, les simulations ont été portées sous un autre champ de force, celui d'AMBER7 utilisé par le logiciel NAMD. La procédure à suivre pour générer une MD avec cet outil est plus complexe que celle à suivre pour GROMACS. Néanmoins, il était intéressant de l'expérimenter, dans la mesure où le champ de force AMBER, contrairement à GROMOS87, est un champ de force tout atome.

6.4.2.1 Procédure pour les complexes substrat-enzyme

À l'instar de GROMACS, il faut convertir les fichiers de structures des protéines ou modèles en fichiers de topologie (.top) et de coordonnées (.crd) afin qu'ils soient lus et traités par NAMD. La création de ceux-ci nécessite préalablement un travail sur le fichier de structure de la protéine (définition des extrémités N-terminales et C-terminales de la protéine, réattribution des noms des atomes et des résidus selon le dictionnaire des librairies d'AMBER …) ainsi que l'obtention du fichier de topologie (.prep ou .frcmod) de la molécule, non présente dans les librairies d'AMBER.

NAMD ne possède pas de serveur automatique dédié au calcul de topologie des petites molécules. À la place, il faut utiliser *antechamber*, un package dédié à la création de topologies pour le champ de force AMBER. Il existe deux méthodes pour créer la topologie des petites molécules sous *antechamber* selon la manière de calculer et d'attribuer les charges à la molécule (cf. Figure 6-9). La première méthode est réalisée par RESP qui effectue un calcul quantique pour reproduire des potentiels électrostatiques afin d'attribuer les charges à la molécule. La seconde méthode dite AM1BCC est de nature semi-empirique : elle consiste à calculer les charges partielles dérivées de la fonction d'onde AM1, qui sont ensuite corrigées par BCC afin de générer et d'attribuer les charges partielles.

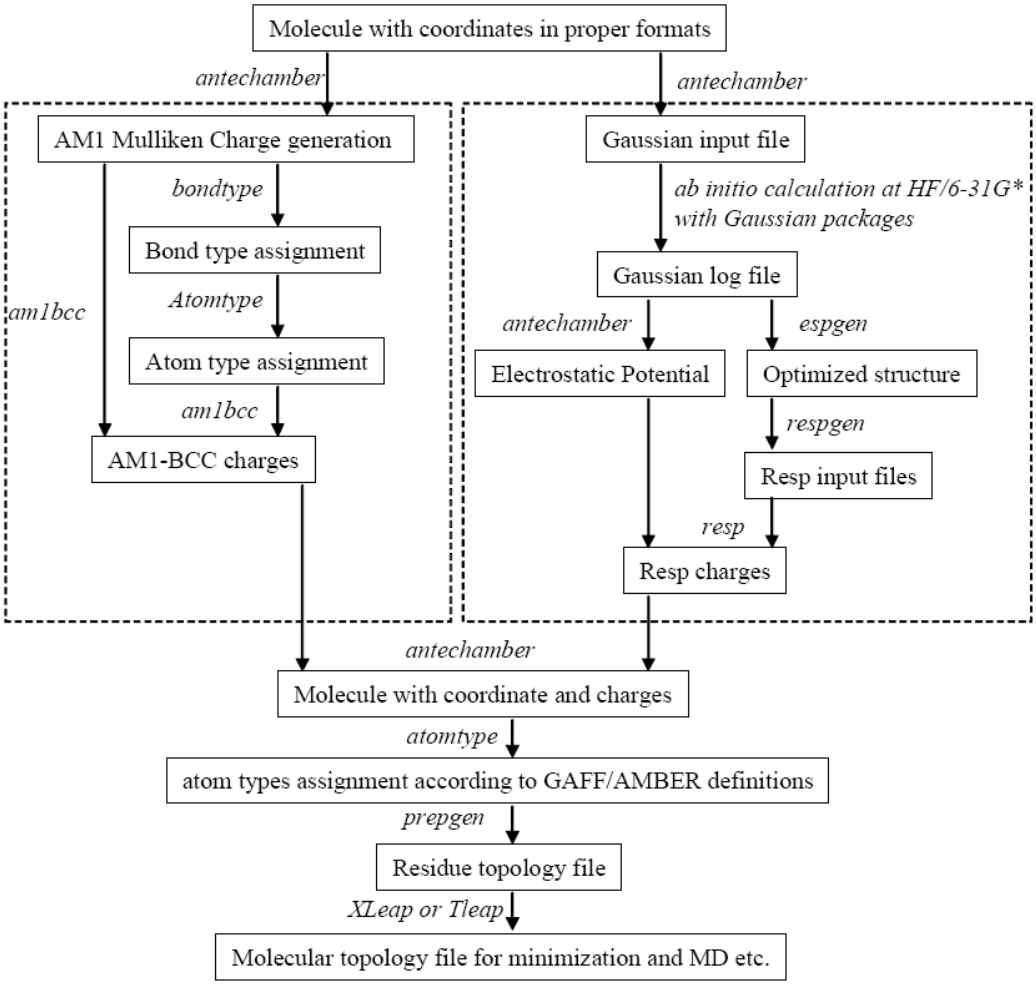


Figure 6-9 Organigramme de la procédure à suivre pas à pas pour générer une topologie AMBER d'une molécule sous *antechamber*. Les procédures basiques de calcul des charges sont montrées dans les rectangles en pointillés (AM1-BCC à gauche et RESP à droite) (Source : Manuel d'*antechamber*)

Nous avons appliqué la méthode AM1-BCC pour construire la topologie du CPA. Par ailleurs, NAMD ne connaît pas non plus la topologie de l'hème, contrairement à GROMACS. Une création de topologie de l'hème sous *antechamber* se voyait donc également nécessaire. À la suite d'une publication parue en 2005 (Oda et *al.*, 2005), il a été possible d'utiliser une topologie déjà existante et disponible sur le net.

Lorsque les deux fichiers de topologies (.prep ou .frcmod) sont construits, ils peuvent être utilisés pour créer les fichiers de topologies et de coordonnées du complexe entier (fichier .top et .crd). Ces derniers sont construits à l'aide de l'utilitaire *Leap*, utilisable sous forme de ligne de commandes. Ainsi, c'est sous *Leap* que l'utilisateur demande l'attachement de l'hème à la protéine (en précisant sur

quels atomes cette liaison doit s'opérer) mais aussi la création de la boîte de solvatation périodique et son remplissage par de l'eau et des contre-ions.

Lorsque les fichiers de topologie et de coordonnées sont générés, NAMD peut effectuer la simulation de MD. Contrairement à GROMACS, rien n'est automatisé ici : l'utilisateur doit définir lui-même comment les forces vont être appliquées au système, renseigner sur la taille de la boîte ainsi que les conditions périodiques (par calcul du volume sous VMD) etc. NAMD impose donc un contrôle complet des paramètres de la simulation. Les étapes de la simulation sont les mêmes que celles adoptées sous GROMACS : d'abord une étape de minimisation, puis d'équilibration (principalement en fonction de la température dans un premier temps) et une phase de dynamique moléculaire. Les paramètres sont également identiques à ceux utilisés lors des simulations de MD sous GROMACS : température identique à l'équilibre (300K), solvant explicite, pression constante (1 bar), nombre de pas et durée de simulation (2 ns) etc. NAMD travaillant en « tout atome » le temps de calcul pour une même trajectoire est plus long. Pour cette raison, seules 3 trajectoires ont été simulées sous NAMD pour comparer avec les résultats de simulations sous GROMACS (avec les deux énantiomères) : le modèle sauvage du CYP2B6, le simple mutant V477Y, et le double mutant, I114V/V477F.

6.4.2.2 Résultats et Analyses des simulations NAMD

Visuellement, les simulations obtenues sous NAMD ne semblent pas si différentes de celles obtenues sous GROMACS. L'hème semble se déformer un peu plus, mais cette déformation reste minime. Pour une analyse plus quantitative, nous avons entrepris d'analyser les résultats de simulation sous NAMD à l'aide des outils de GROMACS présentés plus haut (section 6.4.1.2). Pour cela, il fallait préalablement transformer les fichiers de trajectoires AMBER (.dcd) en format lisible par GROMACS (.trr) et également construire un fichier de topologie GROMOS à partir de la structure du système. Pour les trajectoires, cette transformation peut se faire par le logiciel VMD. En revanche, la topologie GROMOS ne s'obtient qu'en appliquant le protocole de GROMACS (utilisation du programme *pdb2gmx*).

Ces manipulations une fois réalisées, on obtient alors les résultats présentés dans les Figure 6-10 et Figure 6-11.

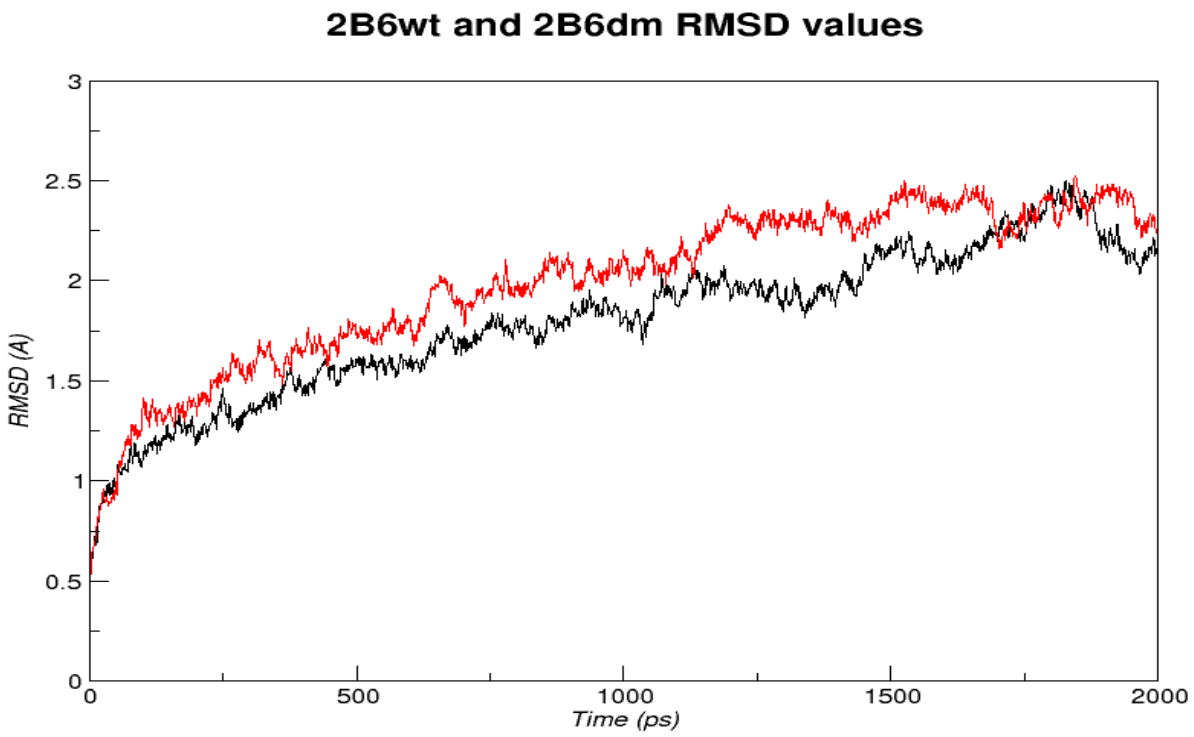


Figure 6-10 Évolution du RMSD du CYP 2B6 wild type (en rouge) et CYP 2B6 I114V/V477W (en noir) durant les 2ns de MD sous NAMD. Dans les deux simulations, un plateau aux alentour de 2 Å est atteint après 1 ns de simulation. Ce plateau est plus haut par rapport à celui obtenu sous GROMACS, et est atteint également plus tard.

Sous le champ de force AMBER, les modèles semblent atteindre une conformation stable après seulement 1,2 ns, ce qui est nettement plus tardif par rapport aux simulations sous GROMACS où les modèles atteignaient leurs conformations stables à 500 ps (cf. Figure 1 de l'article). Par ailleurs, sous AMBER, le plateau semble moins bien marqué, aux alentours de 2,3 Å, alors que sous GROMOS, il était de 1,7 Å. Néanmoins, les simulations sont relativement similaires, dans la mesure où les modèles atteignent au final une conformation stable, quel que soit le champ de force utilisé. En fin de simulation de MD, le RMSD calculé sur le squelette peptidique pour le CYP 2B6wt est de 2,44 Å par rapport à son homologue sous GROMACS, tandis que celui du CYP 2B6dm est de 2,41 Å : on obtient à peu près les mêmes structures d'arrivée pour les deux méthodes.

Ce premier résultat est très encourageant, puisqu'il confirme que l'approche par dynamique moléculaire en solvant explicite est fiable, et indépendamment du champ de force utilisé. Le fait aussi que les simulations s'équilibrent autour de la même configuration dénote aussi d'une certaine reproductibilité des modèles construits pour le CYP2B6 et ses mutants.

Après avoir vérifié la stabilité des modèles, la seconde vérification porte sur le comportement du CPA dans le site actif du CYP 2B6 sauvage et du CYP 2B6 double mutant I114V/V477W. La Figure 7 de l'article montre justement par l'évolution des RMSDs du CPA lors de la simulation de MD la restriction de mobilité du CPA dans le site actif du double mutant par rapport au site actif sauvage. Cette perte de mobilité pouvait alors expliquer l'augmentation de l'efficacité de son métabolisme dans le double mutant.

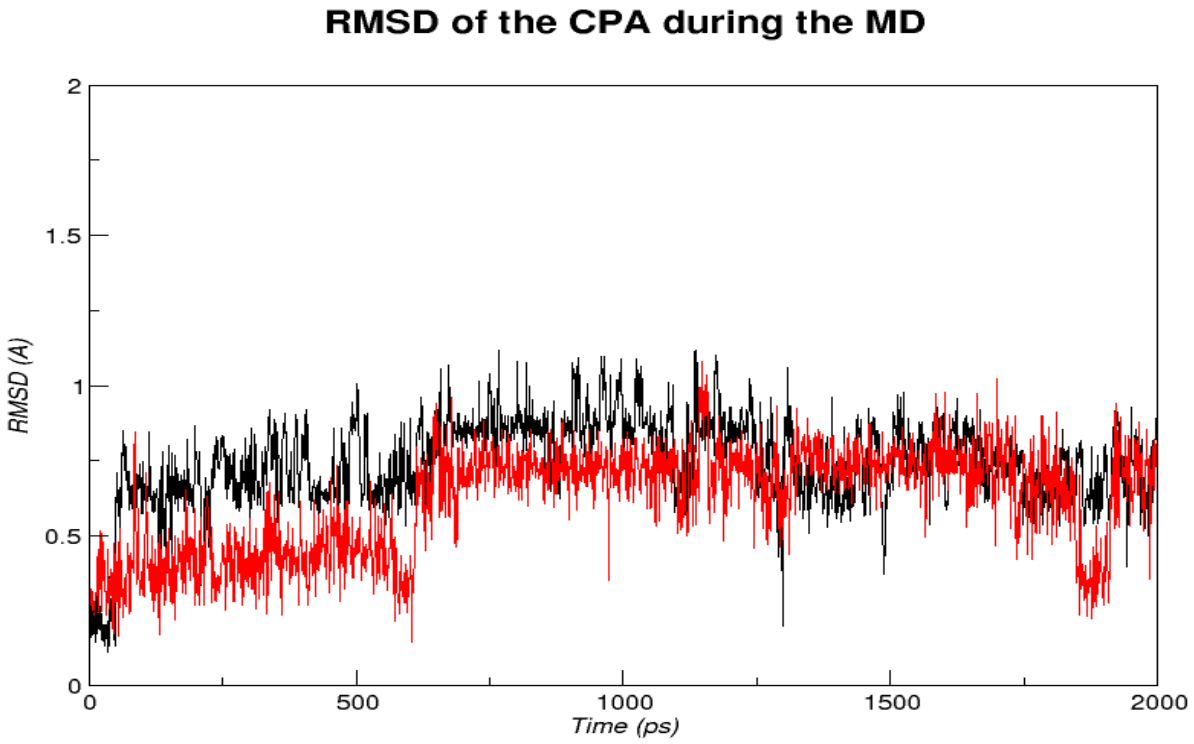


Figure 6-11 Comportement lors de la dynamique sous NAMD du CPA (énantiomère S dans cet exemple) dans le site actif du CYP 2B6 wt (en rouge) et CYP 2B6 I114V/V477W (en noir). Comme sous GROMACS, la tendance des amplitudes est en faveur d'une restriction de mobilité du CPA dans le site actif du double mutant. Cette tendance est cependant moins marquée que sous GROMACS.

Sous NAMD, les mêmes fluctuations du RMSD sont retrouvées, avec différence entre le double mutant et le sauvage un peu moins marquée que dans l'expérience sous GROMACS. L'allure générale est conservée : le CPA semble également dans ces simulations plus contraint, moins mobile dans le CYP 2B6 I114V/V477W que dans le CYP 2B6 sauvage. Comme pour les simulations sous GROMACS, le « décrochage » observé en début de simulation dans le cas du CYP 2B6 double mutant et les niveaux de RMSD correspondent au repositionnement du CPA. Les différents paliers dans le RMSD du CPA (S) du CYP 2B6 sauvage (à 700 ps et à 1800 ps) correspondent à des changements d'orientations du groupe P=O du CPA : par moment pointant vers l'hélice I, par moment pointant vers

le côté opposé. Ceci peut trouver une explication dans le fait qu'à part le glutamate en position 301, les résidus voisins de l'hélice I sont à caractère hydrophobe, sans affinité particulière pour le groupe polaire P=O du CPA : le groupe P=0 subit donc moins de répulsion en face de l'O du carboxyle du GLU 301 que lorsqu'il fait face aux groupements hydrophobes voisins. De ce fait, ce dernier peut se trouver soit en position défavorable (sauf lorsqu'une molécule d'eau se trouve dans le voisinage, mais cela n'a pas été observé non plus sous NAMD), tourné vers le GLU 301 soit dans une position opposée à l'hélice I où il se trouve la plus grande partie du temps. Dans les deux cas, aucun des deux positionnements ne semble favoriser une position particulière du métabolisme (Carbone près du groupement NH) plus que l'autre. À noter enfin que ce basculement du CPA est moins prononcé sur l'énantiomère R, mais la simulation n'est peut être pas suffisamment longue pour pouvoir observer ce phénomène.

6.5 Conclusion

Dans le cas particulier de cette collaboration entre l'équipe d'I. de Waziers et la nôtre, les méthodes de la bioinformatique structurale ont pu apporter des réponses concrètes aux préoccupations des cliniciens impliqués dans cette approche de prodrogue activée sous thérapie génique. La confrontation des simulations et des paramètres cinétiques mesurés a permis en particulier de fournir des éléments prédictifs pour la conception de nouveaux mutants. L'issue la plus notable de cette collaboration fut l'obtention d'un double mutant du CYP 2B6 qui, selon les expériences *ex vivo* de cytotoxicité sur des cellules cancéreuses humaines, présente une meilleure affinité ou du moins une meilleure efficacité sur le CPA : le CYP 2B6 double mutant est donc un candidat à fort potentiel pour des expériences de thérapie génique qui utilisent le CPA comme agent chimiothérapique.

Par ailleurs, d'autres simulations ont été opérées sur les modèles, visant à confirmer les premiers résultats. Celles-ci ont été effectuées sous un champ de force et un environnement très différents. Les résultats obtenus sont en accord avec ceux décrits dans l'article, prouvant la fiabilité des premiers résultats obtenus.

L'ensemble de tout ce travail, à la fois expérimental, prédictif, et reproductible permet donc de dresser un « profil » de site actif de CYP 2B6 adapté au métabolisme du CYP 2B6. Ainsi, les mutations pour lesquelles une amélioration de l'efficacité catalytique a été observée, correspondent dans la majeure partie des cas à des résidus positionnés dans une boucle ou une hélice faisant partie du toit du site actif, à savoir les structures secondaires extrêmement variables des P450s : l'hélice F pour

le mutant L199M, la boucle B–C pour les mutants F107V et I114V, et enfin le long coude en β hairpin formé de part et d'autre du feuillet C-terminal (β_4) pour les mutations de la série 477. Ces informations viennent donc compléter les observations et remarques formulées dans les chapitres antérieurs concernant ces fameuses régions où il était difficile aussi bien de déterminer que de positionner les blocs structuraux lors de la reconstruction d'un modèle de CYP. Ainsi, même si les mécanismes mis en jeu dans la reconnaissance des substrats ne sont pas encore élucidés, ces résultats apportent des renseignements supplémentaires sur les régions impliquées dans la reconnaissance du substrat.

Par ailleurs, le façonnage du site actif du CYP 2B6 a abouti à de meilleurs résultats par voie stérique, en respectant les polarités d'origine du site pour ne pas disperser les positions stables du CPA dans le site actif : la stratégie « polaire » (mutation par des résidus polaires) s'est en effet conclue par des échecs, probablement en raison d'une dissipation (ou à l'inverse, multiplicité) des positions d'interactions établies avec le CPA, entrainant une diminution de la statistique de positionnement favorable au métabolisme.

Enfin, le GLU 301 semble jouer un rôle important vis-à-vis du CPA. Ainsi, par remplacement de ce résidu par une Arginine, une Glutamine ou par un groupement hydrophobe, pourrait supprimer le basculement du CPA et orienter le groupe P=0 vers une position unique à l'opposé de l'hélice I. Le cas d'un triple mutant a donc été soulevé avec nos collaborateurs, mais aucune expérience n'a été entamée encore.

6.6 Article

Les pages qui suivent correspondent à l'article intitulé « Improvement of Cyclophosphamide Activation by CYP2B6 Mutants : from *in Silico* to *ex Vivo* » (en soumission au moment de la rédaction du manuscrit).

Quatrième Partie

Conclusion et perspectives

CHAPITRE 7

Bilan et nouvelles idées

« Savoir ce que tout le monde sait, ce n'est rien savoir.
Le savoir commence là où commence ce que le monde ignore.»
Rémy de Gourmont (1858– 1915 ap JC)

L'objectif de ce travail a été initialement de comprendre les mécanismes moléculaires impliqués dans la reconnaissance de substrats par les cytochromes P450, en se servant de l'exemple du CYP3A4. Il est évident qu'en raison de la complexité du sujet et des moyens mis en œuvre pour étudier le sujet, il n'a pas été possible ni question de venir à bout de cette problématique au bout de ces trois années de thèse, d'autant plus que de par le monde, je n'étais pas le seul à travailler sur ce sujet : d'autres équipes étrangères, avec parfois le soutien de compagnies pharmaceutiques, sont sur l'étude de cette superfamille depuis déjà plusieurs années.

Je pense néanmoins avoir développé et mis en place les bases d'une méthode suffisamment précise qui peut apporter quelques éléments de réponse à cette problématique : cette méthode allie à la fois la recherche et la sélection de séquences de P450s, à leur reconstruction par homologie, même à bas taux d'identité. Dans les deux cas, la méthode se base sur des informations structurales, contenues dans des éléments particuliers et innovants : les CSBs. Afin d'exploiter l'information contenue dans ces CSBs, l'ensemble de la méthode s'articule autour d'un programme que j'ai développé au cours de mes trois années de thèse : le programme *Caliseq*. C'est ce dernier qui va permettre le positionnement optimal des CSBs sur une séquence cible, soit vers un but de production d'alignement de séquences (pour une construction de modèle par exemple), soit pour une identification de P450s à l'aide de ces CSBs utilisés comme signature structurale des P450s.

Pour le moment, c'est principalement le travail d'identification de P450s, à savoir la recherche de séquences de P450s dans un banque de données (ou dans un génome nouvellement séquencé) qui a donné les résultats les plus prometteurs. Les alignements construits suite au positionnement des blocs n'ont pour leur part pas pu conduire à l'obtention des meilleurs modèles lors de comparaison avec d'autres méthodes si l'on se base sur le calcul du RMSD entre la structure connue avec le modèle. Sachant que les jeux de *templates* utilisées et logiciel de reconstruction (Modeller) sollicité sont les mêmes pour toutes les méthodes utilisées, la principale différence réside donc exclusivement au niveau de l'alignement produit. Cet alignement multiple de séquences dépend principalement du positionnement des blocs sur la séquence cible : si le positionnement des CSBs sur la séquence cible est mal réalisée, toutes les étapes qui suivent s'en trouvent affectées. Pour le moment, l'information structurale (3D) contenue dans les CSBs doit être transformée en information séquentielle (1D) pour permettre l'alignement des blocs sur la séquence cible. Pour ne pas perdre toute l'information lors de ce « passage » 3D vers 1D, des profils sont générés. Dans la version finale de *Caliseq*, ces profils sont en outre construits à partir d'une matrice de similarité spécifique aux séquences contenues dans les

blocs, à savoir, ceux des *templates* de P450s : cela avait pour but d'aider l'alignement de ces profils sur la séquence cible avec une fonction de score spécifique aux P450s.

De nombreuses méthodes ont été proposées pour améliorer l'alignement inter-blocs, mais aucune en revanche n'a été suggérée pour favoriser le positionnement des blocs sur la séquence cible, en dehors d'un calcul d'une matrice de similarité spécifique aux P450s. Il est donc intéressant en perspective de faire un tour rapide des informations à « mi-chemin » entre la 3D et la 1D qui n'ont pas été encore exploitées pour améliorer le positionnement des blocs sur la séquence cible par *Caliseq*, et pouvoir trancher sur les incertitudes rencontrées au cours de la thèse pour les blocs « flous ».

La première idée qui vient à l'esprit est l'utilisation des éléments de structures secondaires dans les alignements. Ceux-ci ne seraient pas exploités à la manière de Matras, comme base pour l'alignement structural, mais plutôt comme une information supplémentaire qui seraient incluse ou non dans le profil. Il conviendrait alors d'appliquer le logiciel DSSP sur les *templates*, pour avoir les informations de structures secondaires de ces *templates*. Concernant la séquence cible, il n'est pas possible d'attribuer les SSEs dans la mesure où la structure est censée être inconnue, mais il est en revanche possible de les prédire grâce à des logiciels comme PSIPRED (Jones, 1999) par exemple. Lors d'essais de PSIPRED sur les séquences de P450s (dont les structures sont connues) j'ai obtenus des résultats d'une fiabilité de 50 à 60% en moyenne, où des décalages d'hélices sont observés par exemple, entre les SSEs de la structure réelle et ceux prédits par PSIPRED. Au vu de ces résultats, il paraît donc dangereux d'enrichir l'information des blocs par cette information de structure secondaire prédite sur la structure cible par PSIPRED et fournie des *templates* à partir de DSSP. Il serait néanmoins intéressant d'essayer en attendant une amélioration de PSIPRED ou d'un logiciel équivalent.

L'équipe de JP. Mornon (CNRS UMR 7590) a beaucoup travaillé sur la présence de résidus hydrophobes cruciaux, à certaines positions spécifiques dans des alignements structuraux, formant des « amas hydrophobes » lors du repliement. La conservation de repliement des protéines serait liée à la présence et à la position dans la séquence de ces résidus hydrophobes qui constitueraient le cœur de la protéine. Ces amas hydrophobes se formeraient dans les tout premiers stades de repliement de la protéine et guideraient la formation ultérieure des structures secondaires (Papandreou et *al*., 2004). La méthode HCA, ou Hydrophobic Cluster of Analysis (Gaboriaud et *al*., 1987), repose sur la détection et l'analyse de ces amas hydrophobes et permet entre autres, la comparaison de séquences, même très divergentes. Comme il s'agit ici de comparer des séquences à basse identité de séquence (< 30%), on

pourrait envisager d'utiliser cette information d'amas hydrophobe dans le positionnement des CSBs sur la séquence cible. En effet, ces CSBs sont censés contenir l'information de repliements conservés spécifiques des P450s : l'utilisation des amas hydrophobes renforcerait peut être l'information de conservation de repliement contenue dans les CSBs.

Ces deux approches (prise en compte des SSEs et des amas hydrophobes) pour améliorer le positionnement des blocs sur la séquence cible seraient déjà intéressantes à étudier. Le positionnement correct des blocs étant la clef vers la reconstruction de modèles plus précis, il est crucial de trouver des approches pour l'améliorer.

Lorsque les deux approches seront réellement au point, il sera alors temps de les combiner pour fournir des éléments de réponse à la problématique posée. En effet, la force de la méthode ne se situe pas dans les deux approches prises indépendamment –d'autant plus que depuis le début du mois d'Août, une banque spécifique des P450s, appelée CYPED est disponible (Fischer et *al*., 2007)– mais dans leur utilisation complémentaire : à terme, l'objectif consiste à construire une véritable banque de sites actifs de P450s, à partir des séquences identifiées et reconstruites à partir des CSBs. Cette banque qui sera mise à disposition de la communauté scientifique (idéalement au travers d'un serveur) peut servir de base de cibles structurales pour le criblage virtuel des divers composés chimiques ou médicamenteux, avec dans l'idéal, la possibilité de faire un tri grossier entre P450 affins et non affins pour une molécule donnée. Une telle banque peut permettre des comparaisons pour tester des méthodes rapides de docking en soumettant des molécules connues dont les cibles sont identifiées. Elle peut aussi fournir des éléments de réponse sur les mécanismes de reconnaissance des substrats par les P450s. Enfin des expériences de mutations *in silico* comme ceux opérés dans le modèle du CYP 2B6 pour des expériences de thérapie génique, pourront être effectuées à partir des modèles de cette banque.

Pour le moment, je n'en suis pas encore là, dans la mesure où il faudrait préalablement doter la méthode d'une automatisation qui réaliserait toutes les étapes entre le positionnement des blocs sur la séquence cible et la reconstruction et la validation du modèle obtenu à partir de l'alignement.

Par ailleurs, la banque de sites actifs présenterait une valeur ajoutée si elle était conçue de manière à ce que l'utilisateur puisse soumettre en requête un composé chimique et récupérer en résultat les modèles de P450s qui présenteraient la meilleure affinité pour ce ligand. Pour cela, il faudrait donc doter la banque d'un moyen « d'apprentissage » qui en fonction du ligand soumis, décidera quels

P450s retourner. L'idée que je propose pour répondre à ce problème est l'utilisation combinée des méthodes d'apprentissage (comme les SVM pour Support Vector Machine) aux méthodes de Screening Virtuels à Haut débit. Pour chaque modèle de P450 de cette banque, il faudrait rassembler dans un premier temps, l'ensemble des ligands du P450 étudié dont les données biochimiques sont disponibles, puis les séparer en deux groupes (généralement 2/3-1/3), les premiers pour l'apprentissage et les seconds pour le test. Ces ligands seraient utilisés dans des essais de criblage et le score attribué à chaque arrimage serait appris par le SVM. Cette étape est importante, dans la mesure où il faudra indiquer au SVM s'il a bien appris avec les donnés du premier groupe (distinguer les ligands reconnus par le P450 des ligands non reconnus en fonction du score). Une fois l'apprentissage effectué, les ligands du jeu de test seraient alors « dockés » dans chaque P450 et selon le score d'arrimage (et ce qu'il a appris), le SVM serait en mesure de déterminer si le ligand testé devrait être reconnu ou non par le P450 concerné. Par cette méthode, c'est le score d'arrimage qui servira à déterminer les CYPs qui reconnaissent le ligand soumis.

Ne disposant pas encore de cette banque, j'ai tout de même fait appel à M. Montes (INSERM U648), pour des essais de criblage virtuel à haut débit sur certaines structures de P450s disposant d'un ligand co-cristallisé. L'idée était de vérifier la faisabilité des expériences de docking automatisé sur ce genre de protéine. Les premiers résultats de docking par divers logiciels de docking (FRED, GOLD, FLEX, AutoDock) obtenus avec le CYP 2C9[21] et son ligand (8-bromo-adenosine-5'-monophosphate) co-cristallisé (pdb 1ro9) n'ont pas abouti aux résultats espérés : un positionnement similaire du ligand dans le site actif, à celui du cristal X a été observé, mais inversé au niveau de la polarité (l'atome à métaboliser se retrouvant à l'opposé du fer de l'hème). Nous n'avons pas eu le temps d'explorer davantage ces méthodes pour le moment. Dans tous les cas, il faudra nécessairement réussir à reproduire un positionnement de ligand observé dans les cristaux à partir de ces logiciels de docking dans un premier temps, avant d'envisager un criblage à plus grande échelle. Une fois les premiers tests avec des données structurales concluants, il sera alors le moment d'expérimenter des données biochimiques dont on ne dispose d'aucune information structurale. Pour cela, différents positionnements des ligands seront retenus, et ce n'est plus un score unique d'arrimage qui est appris pour un complexe (P450-substrat) donné, mais une répartition ou plutôt, un intervalle de confiance. Ce n'est qu'une fois ces expériences validées, que la méthode pourra être applicable sur une banque de sites actifs de P450s.

[21] Nous avons commencé par le CYP 2C9 dans la mesure où c'était le P450 qui disposait le plus de données expérimentales (aussi bien structurales que biochimiques).

Il reste donc beaucoup à faire, et l'issue de la méthode proposée (élaboration de la banque de sites actifs, couplée avec un système d'apprentissage) ne fournira qu'un outil de travail pour la problématique posée : les mécanismes de reconnaissances moléculaire par les P450s sont loin d'être élucidés. Finalement, mon travail de thèse a été réellement insignifiant par rapport à la masse de travail qui reste encore à fournir pour comprendre ces enzymes. Ma pierre posée à l'édifice de la connaissance des P450s me parait si fragile et anodine. Je suppose que c'est l'état d'esprit de la plupart des étudiants qui arrivent au terme de leur thèse, frustrés de n'avoir pu en faire plus. Je me console tant bien que mal en me disant que je ne suis pas seul à travailler sur ce sujet, et que cela fait déjà plus d'un demi-siècle que les chercheurs se concentrent sur cette superfamille.

Bibliographies

A

AHO, A. et CORASICK, H. (1975). "Efficient string matching: an aid to bibliographic search." *Comm. ACM* **18**(6): 333-340.

ALBERTS, B., BRAY, D., LEWIS, J., RAFF, M., TOBERTS, K. et WATSON, J. (1994). *Molecular Biology of the Cell*. New York, London, Arland Publishing, Inc.

ALTSCHUL, S. F., GISH, W., MILLER, W., MYERS, E. W. et LIPMAN, D. J. (1990). "Basic local alignment search tool." *J Mol Biol* **215**(3): 403-10.

ALTSCHUL, S. F., MADDEN, T. L., SCHAFFER, A. A., ZHANG, J., ZHANG, Z., MILLER, W. et LIPMAN, D. J. (1997). "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." *Nucleic Acids Res* **25**(17): 3389-402.

ANFINSEN, C. B. (1973). "Principles that govern the folding of protein chains." *Science* **181**(96): 223-30.

APWEILER, R., BAIROCH, A., WU, C. H., BARKER, W. C., BOECKMANN, B., FERRO, S., GASTEIGER, E., HUANG, H., LOPEZ, R., MAGRANE, M., et al. (2004). "UniProt: the Universal Protein knowledgebase." *Nucleic Acids Res* **32**(Database issue): D115-9.

APWEILER, R., GATEAU, A., CONTRINO, S., MARTIN, M. J., JUNKER, V., O'DONOVAN, C., LANG, F., MITARITONNA, N., KAPPUS, S. et BAIROCH, A. (1997). "Protein sequence annotation in the genome era: the annotation concept of SWISS-PROT+TREMBL." *Proc Int Conf Intell Syst Mol Biol* **5**: 33-43.

ARTYMIUK, P. J., RICE, D. W., MITCHELL, E. M. et WILLETT, P. (1990). "Structural resemblance between the families of bacterial signal-transduction proteins and of G proteins revealed by graph theoretical techniques." *Protein Eng* **4**(1): 39-43.

B

BACHMANN, K. A. (1996). "The Cytochrome P450 Enzymes of Hepatic Drug Metabolism: How are their Activities Assessed In Vivo, and what is their Clinical Relevance?" *Am J Ther* **3**(2): 150-171.

BAINS, W. (1986). "MULTAN: a program to align multiple DNA sequences." *Nucleic Acids Res* **14**(1): 159-77.

BAIROCH, A. (1993). "The PROSITE dictionary of sites and patterns in proteins, its current status." *Nucleic Acids Res* **21**(13): 3097-103.

BAIROCH, A. et BOECKMANN, B. (1993). "The SWISS-PROT protein sequence data bank, recent developments." *Nucleic Acids Res* **21**(13): 3093-6.

BATEMAN, A., COIN, L., DURBIN, R., FINN, R. D., HOLLICH, V., GRIFFITHS-JONES, S., KHANNA, A., MARSHALL, M., MOXON, S., SONNHAMMER, E. L., et al. (2004). "The Pfam protein families database." *Nucleic Acids Res* **32**(Database issue): D138-41.

BAUM, L., PETRIE, T., SOULES, G. et WEISS, N. (1970). "A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains." *Ann. Math. Stat.* **41**: 164-171.

BERNHARDT, R. (1996). "Cytochrome P450: structure, function, and generation of reactive oxygen species." *Rev Physiol Biochem Pharmacol* **127**: 137-221.

BOWIE, J. U., LUTHY, R. et EISENBERG, D. (1991). "A method to identify protein sequences that fold into a known three-dimensional structure." *Science* **253**(5016): 164-70.

BRANDON, C. et TOOZE, J. (1999). *Introduction to Protein Structure*. New York, 2nd edition, Garland Plubishing Inc.

BROOKS, B. R., BRUCCOLERI, R. E., OLAFSON, B. D., STATES, D. J., SWAMINATHAN, S. et KARPLUS, M. (1983). "CHARMM: A program for macromolecular energy, minimization, and dynamics calculations." *J. Comp. Chem.* **4**: 187–217.

BRUNGER, T. A. (1992). "X-PLOR 3.1: A system for X-ray crystallography and NMR." *Yale University Press, New Haven, CT.*

BRYANT, S. H. et AMZEL, L. M. (1987). "Correctly folded proteins make twice as many hydrophobic contacts." *Int J Pept Protein Res* **29**(1): 46-52.

BRYANT, S. H. et LAWRENCE, C. E. (1991). "The frequency of ion-pair substructures in proteins is quantitatively related to electrostatic potential: a statistical model for nonbonded interactions." *Proteins* **9**(2): 108-19.

C

CAMPROUX, A. C., GAUTIER, R. et TUFFERY, P. (2004). "A hidden markov model derived structural alphabet for proteins." *J Mol Biol* **339**(3): 591-605.

CANUTESCU, A. A., SHELENKOV, A. A. et DUNBRACK, R. L., JR. (2003). "A graph-theory algorithm for rapid protein side-chain prediction." *Protein Sci* **12**(9): 2001-14.

CARPENTIER, M. (2005). Méthodes de détection des similarités structurales: caractérisation des motifs conservés dans les familles de structures pour l'annotation des génomes, Université Pierre et Marie Curie.

CARPENTIER, M., BROUILLET, S. et POTHIER, J. (2005). "YAKUSA: a fast structural database scanning method." *Proteins* **61**(1): 137-51.

CHAPPLE, C. (1998). "Molecular-Genetic Analysis Of Plant Cytochrome P450-Dependent Monooxygenases." *Annu Rev Plant Physiol Plant Mol Biol* **49**: 311-343.

CHICHE, L., GREGORET, L. M., COHEN, F. E. et KOLLMAN, P. A. (1990). "Protein model structure evaluation using the solvation free energy of folding." *Proc Natl Acad Sci U S A* **87**(8): 3240-3.

CHOTHIA, C. et LESK, A. M. (1986). "The relation between the divergence of sequence and structure in proteins." *Embo J* **5**(4): 823-6.

COJOCARU, V., WINN, P. J. et WADE, R. C. (2007). "The ins and outs of cytochrome P450s." *Biochim Biophys Acta* **1770**(3): 390-401.

COLOVOS, C. et YEATES, T. O. (1993). "Verification of protein structures: patterns of nonbonded atomic interactions." *Protein Sci* **2**(9): 1511-9.

COOPER, D. Y., LEVIN, S., NARASIMHULU, S. et ROSENTHAL, O. (1965). "Photochemical Action Spectrum Of The Terminal Oxidase Of Mixed Function Oxidase Systems." *Science* **147**: 400-2.

CORPET, F. (1988). "Multiple sequence alignment with hierarchical clustering." *Nucleic Acids Res* **16**(22): 10881-90.

COSME, J. et JOHNSON, E. F. (2000). "Engineering microsomal cytochrome P450 2C5 to be a soluble, monomeric enzyme. Mutations that alter aggregation, phospholipid dependence of catalysis, and membrane binding." *J Biol Chem* **275**(4): 2545-53.

CRESPI, C. L., PENMAN, B. W., GELBOIN, H. V. et GONZALEZ, F. J. (1991). "A tobacco smoke-derived nitrosamine, 4-(methylnitrosamino)-1-(3-pyridyl)-1-butanone, is activated by multiple human cytochrome P450s including the polymorphic human cytochrome P4502D6." *Carcinogenesis* **12**(7): 1197-201.

CROCHEMORE, M. et RYTTER, W. (1994). *Text Algorithm*, Oxford University Press.

CROFTS, F., COSMA, G. N., CURRIE, D., TAIOLI, E., TONIOLO, P. et GARTE, S. J. (1993). "A novel CYP1A1 gene polymorphism in African-Americans." *Carcinogenesis* **14**(9): 1729-31.

CUPP-VICKERY, J. R. et POULOS, T. L. (1995). "Structure of cytochrome P450eryF involved in erythromycin biosynthesis." *Nat Struct Biol* **2**(2): 144-53.

D

DAVYDOV, R., MAKRIS, T. M., KOFMAN, V., WERST, D. E., SLIGAR, S. G. et HOFFMAN, B. M. (2001). "Hydroxylation of camphor by reduced oxy-cytochrome P450cam: mechanistic implications of EPR and ENDOR studies of catalytic intermediates in native and mutant enzymes." *J Am Chem Soc* **123**(7): 1403-15.

DAYHOFF, M. O., ECK, R. V. et PARK, C. M. (1972). "A Model of Evolutionary Changes In Proteins." *Atlas of Protein Sequence and Structure* **5**: 89-99.

DAYHOFF, M. O., SCHWARTZ, R. M. et ORCUTT, B. C. (1978). "A Model of Evolutionary Changes In Proteins." *Atlas of Protein Sequence and Structure* **5**: 345-352.

DEANE, C. M. et BLUNDELL, T. L. (2001). "CODA: a combined algorithm for predicting the structurally variable regions of protein models." *Protein Sci* **10**(3): 599-612.

DEGTYARENKO, K. N. et ARCHAKOV, A. I. (1993). "Molecular evolution of P450 superfamily and P450-containing monooxygenase systems." *FEBS Lett* **332**(1-2): 1-8.

DEPREZ, E., GERBER, N. C., DI PRIMO, C., DOUZOU, P., SLIGAR, S. G. et HUI BON HOA, G. (1994). "Electrostatic control of the substrate access channel in cytochrome P-450cam." *Biochemistry* **33**(48): 14464-8.

DING, X. et KAMINSKY, L. S. (2003). "Human extrahepatic cytochromes P450: function in xenobiotic metabolism and tissue-selective chemical toxicity in the respiratory and gastrointestinal tracts." *Annu Rev Pharmacol Toxicol* **43**: 149-73.

DONATE, L. E., RUFINO, S. D., CANARD, L. H. et BLUNDELL, T. L. (1996). "Conformational analysis and clustering of short and medium size loops connecting regular secondary structures: a database for modeling and prediction." *Protein Sci* **5**(12): 2600-16.

DUAN, Y., WU, C., CHOWDHURY, S., LEE, M. C., XIONG, G., ZHANG, W., YANG, R., CIEPLAK, P., LUO, R., LEE, T., et al. (2003). "A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations." *J Comput Chem* **24**(16): 1999-2012.

DUNBRACK, R. L., JR. et KARPLUS, M. (1993). "Backbone-dependent rotamer library for proteins. Application to side-chain prediction." *J Mol Biol* **230**(2): 543-74.

DUNN, A. R., HAYS, A. M., GOODIN, D. B., STOUT, C. D., CHIU, R., WINKLER, J. R. et GRAY, H. B. (2002). "Fluorescent probes for cytochrome p450 structural characterization and inhibitor screening." *J Am Chem Soc* **124**(35): 10254-5.

E

ESPOSITO, L., DE SIMONE, A., ZAGARI, A. et VITAGLIANO, L. (2005). "Correlation between omega and psi dihedral angles in protein structures." *J Mol Biol* **347**(3): 483-7.

ESTABROOK, R. W. (1996). "The remarkable P450s: a historical overview of these versatile hemeprotein catalysts." *Faseb J* **10**(2): 202-4.

ESTABROOK, R. W., COOPER, D. Y. et ROSENTHAL, O. (1963). "The Light Reversible Carbon Monoxide Inhibition Of The Steroid C21-Hydroxylase System Of The Adrenal Cortex." *Biochem Z* **338**: 741-55.

ESTABROOK, R. W., MASON, J. I., SIMPSON, E. R., PETERSON, J. A. et WATERMAN, M. R. (1991). "The heterologous expression of the cytochromes P450: a new approach for the study of enzyme activities and regulation." *Adv Enzyme Regul* **31**: 365-83.

EVANS, W. E. et RELLING, M. V. (1999). "Pharmacogenomics: translating functional genomics into rational therapeutics." *Science* **286**(5439): 487-91.

F

FALICOV, A. et COHEN, F. E. (1996). "A surface of minimum area metric for the structural comparison of proteins." *J Mol Biol* **258**(5): 871-92.

FINCH, S. A. E. et STIER, A. (1991). "Rotational diffusion of homo- and hetero-oligomers of cytochrome P450, the functional significance of cooperativity and the membrane structure." *Frontiers in Biotransformation* **5**: 34-70.

FISCHER, M., KNOLL, M., SIRIM, D., WAGNER, F., FUNKE, S. et PLEISS, J. (2007). "The Cytochrome P450 Engineering Database: a navigation and prediction tool for the cytochrome P450 protein family." *Bioinformatics* **23**(15): 2015-7.

FISER, A., DO, R. K. et SALI, A. (2000). "Modeling of loops in protein structures." *Protein Sci* **9**(9): 1753-73.

FLEISCHMANN, R. D., ADAMS, M. D., WHITE, O., CLAYTON, R. A., KIRKNESS, E. F., KERLAVAGE, A. R., BULT, C. J., TOMB, J. F., DOUGHERTY, B. A., MERRICK, J. M., et al. (1995). "Whole-genome random sequencing and assembly of Haemophilus influenzae Rd." *Science* **269**(5223): 496-512.

FLORY, P. J. (1969). *Statistical Mechanics of Chain Molecules*. New York, Wiley.

FRAUSTO DA SILVA, J. J. R. et WILLIAMS, R. J. P. (1991). *The Biological Chemistry of the Elements*. Oxford, Clarendo Press.

G

GABORIAUD, C., BISSERY, V., BENCHETRIT, T. et MORNON, J. P. (1987). "Hydrophobic cluster analysis: an efficient new way to compare and analyse amino acid sequences." *FEBS Lett* **224**(1): 149-55.

GALAKTIONOV, S., NIKIFOROVICH, G. V. et MARSHALL, G. R. (2001). "Ab initio modeling of small, medium, and large loops in proteins." *Biopolymers* **60**(2): 153-68.
GARFINKEL, D. (1958). "Studies on pig liver microsomes. I. Enzymic and pigment composition of different microsomal fractions." *Arch Biochem Biophys* **77**(2): 493-509.
GEORGE, D. G., BARKER, W. C. et HUNT, L. T. (1986). "The protein identification resource (PIR)." *Nucleic Acids Res* **14**(1): 11-5.
GEORGE, D. G., BARKER, W. C. et HUNT, L. T. (1990). "Mutation data matrix and its uses." *Methods Enzymol* **183**: 333-51.
GERBER, N. C. et SLIGAR, S. G. (1994). "A role for Asp-251 in cytochrome P-450cam oxygen activation." *J Biol Chem* **269**(6): 4260-6.
GERSTEIN, M. et ALTMAN, R. B. (1995). "Using a measure of structural variation to define a core for the globins." *Comput Appl Biosci* **11**(6): 633-44.
GERSTEIN, M. et LEVITT, M. (1996). "Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures." *Proc Int Conf Intell Syst Mol Biol* **4**: 59-67.
GERSTEIN, M. et LEVITT, M. (1998). "Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins." *Protein Sci* **7**(2): 445-56.
GIBRAT, J. F., MADEJ, T. et BRYANT, S. H. (1996). "Surprising similarities in structure comparison." *Curr Opin Struct Biol* **6**(3): 377-85.
GIBSON, G. G. et SKETT, P. (1994). *Introduction to Drug Metabolism*. London, Champman & Hall.
GOLDSTEIN, L. et WATERMAN, M. S. (1994). "Approximations to profile score distributions." *J Comput Biol* **1**(2): 93-104.
GOMEZ, D. Y., WACHER, V. J., TOMLANOVICH, S. J., HEBERT, M. F. et BENET, L. Z. (1995). "The effects of ketoconazole on the intestinal metabolism and bioavailability of cyclosporine." *Clin Pharmacol Ther* **58**(1): 15-9.
GONZALEZ, F. J. (1988). "The molecular biology of cytochrome P450s." *Pharmacol Rev* **40**(4): 243-88.
GONZALEZ, F. J. et KORZEKWA, K. R. (1995). "Cytochromes P450 expression systems." *Annu Rev Pharmacol Toxicol* **35**: 369-90.
GOODSELL, D. S. et OLSON, A. J. (1990). "Automated docking of substrates to proteins by simulated annealing." *Proteins* **8**(3): 195-202.
GORMAN, N., WALTON, H. S., SINCLAIR, J. F. et SINCLAIR, P. R. (1998). "CYP1A-catalyzed uroporphyrinogen oxidation in hepatic microsomes from non-mammalian vertebrates (chick and duck embryos, scup and alligator)." *Comp Biochem Physiol C Pharmacol Toxicol Endocrinol* **121**(1-3): 405-12.
GOTOH, O. (1992). "Substrate recognition sites in cytochrome P450 family 2 (CYP2) proteins inferred from comparative analyses of amino acid and coding nucleotide sequences." *J Biol Chem* **267**(1): 83-90.
GOTOH, O. et FUJII-KURIYAMA, Y. (1989). "Evolution, structure and gene regulation of cytochrome P450." *Frontiers in Biotransformation* **1**: 195-243.
GREGORET, L. M. et COHEN, F. E. (1991). "Protein folding. Effect of packing density on chain conformation." *J Mol Biol* **219**(1): 109-22.
GRIBSKOV, M., MCLACHLAN, A. D. et EISENBERG, D. (1987). "Profile analysis: detection of distantly related proteins." *Proc Natl Acad Sci U S A* **84**(13): 4355-8.
GRINDLEY, H. M., ARTYMIUK, P. J., RICE, D. W. et WILLETT, P. (1993). "Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm." *J Mol Biol* **229**(3): 707-21.
GUENGERICH, F. P. (1989a). "Characterization of human microsomal cytochrome P-450 enzymes." *Annu Rev Pharmacol Toxicol* **29**: 241-64.
GUENGERICH, F. P. (1989b). "Structure and function of cytochrome P450." *Frontiers in Biotransformation* **1**: 101-150.
GUENGERICH, F. P. (1991a). "Molecular advances for the cytochrome P-450 superfamily." *Trends Pharmacol Sci* **12**(8): 281-3.
GUENGERICH, F. P. (1991b). "Reactions and significance of cytochrome P-450 enzymes." *J Biol Chem* **266**(16): 10019-22.
GUENGERICH, F. P. (1992a). "Metabolic activation of carcinogens." *Pharmacol Ther* **54**(1): 17-61.
GUENGERICH, F. P. (1992b). "Cytochrome P450: advances and prospects." *Faseb J* **6**(2): 667-8.
GUENGERICH, F. P. (1992c). "Characterization of human cytochrome P450 enzymes." *Faseb J* **6**(2): 745-8.
GUENGERICH, F. P. (1992d). "Human cytochrome P-450 enzymes." *Life Sci* **50**(20): 1471-8.

GUENGERICH, F. P. (1997). "Comparisons of catalytic selectivity of cytochrome P450 subfamily enzymes from different species." *Chem Biol Interact* **106**(3): 161-82.
GUENGERICH, F. P. (2001). "Common and uncommon cytochrome P450 reactions related to metabolism and chemical toxicity." *Chem Res Toxicol* **14**(6): 611-50.
GUENGERICH, F. P. (2005). Human Cytochrome P450 Enzymes. *Cytochrome P450: Structure, Mechanism and Biochemistry*. P. R. E. r. E. (Ortiz de Montellano, Plenum. New York**:** 377-530.
GUENGERICH, F. P., HOSEA, N. A., PARIKH, A., BELL-PARIKH, L. C., JOHNSON, W. W., GILLAM, E. M. et SHIMADA, T. (1998). "Twenty years of biochemistry of human P450s: purification, expression, mechanism, and relevance to drugs." *Drug Metab Dispos* **26**(12): 1175-8.
GUENGERICH, F. P. et JOHNSON, W. W. (1997). "Kinetics of ferric cytochrome P450 reduction by NADPH-cytochrome P450 reductase: rapid reduction in the absence of substrate and variations among cytochrome P450 systems." *Biochemistry* **36**(48): 14741-50.
GUENGERICH, F. P., SHIMADA, T., RANEY, K. D., YUN, C. H., MEYER, D. J., KETTERER, B., HARRIS, T. M., GROOPMAN, J. D. et KADLUBAR, F. F. (1992). "Elucidation of catalytic specificities of human cytochrome P450 and glutathione S-transferase enzymes and relevance to molecular epidemiology." *Environ Health Perspect* **98**: 75-80.
GUNTERT, P., MUMENTHALER, C. et WUTHRICH, K. (1997). "Torsion angle dynamics for NMR structure calculation with the new program DYANA." *J Mol Biol* **273**(1): 283-98.
GUYON, F., CAMPROUX, A. C., HOCHEZ, J. et TUFFERY, P. (2004). "SA-Search: a web tool for protein structure mining based on a Structural Alphabet." *Nucleic Acids Res* **32**(Web Server issue): W545-8.

H

HAKKOLA, J., PASANEN, M., HUKKANEN, J., PELKONEN, O., MAENPAA, J., EDWARDS, R. J., BOOBIS, A. R. et RAUNIO, H. (1996). "Expression of xenobiotic-metabolizing cytochrome P450 forms in human full-term placenta." *Biochem Pharmacol* **51**(4): 403-11.
HAKKOLA, J., PELKONEN, O., PASANEN, M. et RAUNIO, H. (1998). "Xenobiotic-metabolizing cytochrome P450 enzymes in the human feto-placental unit: role in intrauterine toxicity." *Crit Rev Toxicol* **28**(1): 35-72.
HALLAHAN, D. L., CHERITON, A. K., HYDE, R., CLARK, I. et FORDE, B. G. (1993). "Plant cytochrome P-450 and agricultural biotechnology." *Biochem Soc Trans* **21**(4): 1068-73.
HALLAHAN, D. L. et WEST, J. M. (1995). "Cytochrome P-450 in plant/insect interactions: geraniol 10-hydroxylase and the biosynthesis of iridoid monoterpenoids." *Drug Metabol Drug Interact* **12**(3-4): 369-82.
HANNEMANN, F., BICHET, A., EWEN, K. M. et BERNHARDT, R. (2007). "Cytochrome P450 systems--biological variations of electron transport chains." *Biochim Biophys Acta* **1770**(3): 330-44.
HARRISON, A., PEARL, F., SILLITOE, I., SLIDEL, T., MOTT, R., THORNTON, J. et ORENGO, C. (2003). "Recognizing the fold of a protein structure." *Bioinformatics* **19**(14): 1748-59.
HASEMANN, C. A., KURUMBAIL, R. G., BODDUPALLI, S. S., PETERSON, J. A. et DEISENHOFER, J. (1995). "Structure and function of cytochromes P450: a comparative analysis of three crystal structures." *Structure* **3**(1): 41-62.
HELLMOLD, H., RYLANDER, T., MAGNUSSON, M., REIHNER, E., WARNER, M. et GUSTAFSSON, J. A. (1998). "Characterization of cytochrome P450 enzymes in human breast tissue from reduction mammaplasties." *J Clin Endocrinol Metab* **83**(3): 886-95.
HENIKOFF, S. et HENIKOFF, J. G. (1991). "Automated assembly of protein blocks for database searching." *Nucleic Acids Res* **19**(23): 6565-72.
HENIKOFF, S., HENIKOFF, J. G. et PIETROKOVSKI, S. (1999). "Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations." *Bioinformatics* **15**(6): 471-9.
HIGGINS, D. G., BLEASBY, A. J. et FUCHS, R. (1992). "CLUSTAL V: improved software for multiple sequence alignment." *Comput Appl Biosci* **8**(2): 189-91.
HIGGINS, D. G. et SHARP, P. M. (1988). "CLUSTAL: a package for performing multiple sequence alignment on a microcomputer." *Gene* **73**(1): 237-44.
HIROYA, K., MURAKAMI, Y., SHIMIZU, T., HATANO, M. et ORTIZ DE MONTELLANO, P. R. (1994). "Differential roles of Glu318 and Thr319 in cytochrome P450 1A2 catalysis supported by NADPH-cytochrome P450 reductase and tert-butyl hydroperoxide." *Arch Biochem Biophys* **310**(2): 397-401.

HOLM, L., OUZOUNIS, C., SANDER, C., TUPAREV, G. et VRIEND, G. (1992). "A database of protein structure families with common folding motifs." *Protein Sci* **1**(12): 1691-8.
HOLM, L. et SANDER, C. (1992). "Evaluation of protein models by atomic solvation preference." *J Mol Biol* **225**(1): 93-105.
HOLM, L. et SANDER, C. (1996). "Mapping the protein universe." *Science* **273**(5275): 595-603.
HOLM, L. et SANDER, C. (1997). "Dali/FSSP classification of three-dimensional protein folds." *Nucleic Acids Res* **25**(1): 231-4.
HOOFT, R. W. W., SANDER, C. et VRIEND, G. (1996). "Verification of protein structures: side-chain planarity." *J. Appl. Crystallog* **29**: 714-716.
HULO, N., BAIROCH, A., BULLIARD, V., CERUTTI, L., DE CASTRO, E., LANGENDIJK-GENEVAUX, P. S., PAGNI, M. et SIGRIST, C. J. (2006). "The PROSITE database." *Nucleic Acids Res* **34**(Database issue): D227-30.

I

INGELMAN-SUNDBERG, M. (2002). "Polymorphism of cytochrome P450 and xenobiotic toxicity." *Toxicology* **181-182**: 447-52.
IOANNIDES, C. (1999). "Effect of diet and nutrition on the expression of cytochromes P450." *Xenobiotica* **29**(2): 109-54.

J

JEAN, P., POTHIER, J., DANSETTE, P. M., MANSUY, D. et VIARI, A. (1997). "Automated multiple analysis of protein structures: application to homology modeling of cytochromes P450." *Proteins* **28**(3): 388-404.
JONES, D. T., TAYLOR, W. R. et THORNTON, J. M. (1992). "The rapid generation of mutation data matrices from protein sequences." *Comput Appl Biosci* **8**(3): 275-82.
JONES, G., WILLETT, P. et GLEN, R. C. (1995). "Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation." *J Mol Biol* **245**(1): 43-53.
JUDSON, R. S., JEAGER, E. P. et TREASURYWALA, A. M. (1994). "A Genetic Algorithm-Based Method for Docking Flexible Molecules." *Journal of Molecular Strcture: Theochem* **114**: 191-206.
JUNG, J. et LEE, B. (2000). "Protein structure alignment using environmental profiles." *Protein Eng* **13**(8): 535-43.

K

KABSCH, W. (1976). "A solution for the best rotation to relate two sets of vectors." *Acta Crystallographica Section A* **32**: 922-923.
KABSCH, W. (1978). "A discussion of the solution for the best rotation to relate two sets of vectors." *Acta Crystallographica Section A* **34**(5): 827-828.
KABSCH, W. et SANDER, C. (1983). "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features." *Biopolymers* **22**(12): 2577-637.
KAMATAKI, T., NUNOYA, K., SAKAI, Y., KUSHIDA, H. et FUJITA, K. (1999). "Genetic polymorphism of CYP2A6 in relation to cancer." *Mutat Res* **428**(1-2): 125-30.
KARP, R., MILLER, R. E. et ROSENBERG, A. (1972). "Rapid Identifaction of repeated patterns in strings, tree and arrays." *Fourth ACM Symposium on Theory of Computing*: 125-136.
KAWABATA, T. (2003). "MATRAS: A program for protein 3D structure comparison." *Nucleic Acids Res* **31**(13): 3367-9.
KAWABATA, T. et NISHIKAWA, K. (2000). "Protein structure comparison using the markov transition model of evolution." *Proteins* **41**(1): 108-22.
KEARSLEY, S. (1989). "On the orthongonal transformation used for strctural comparisons." *Acta Crystallographica Section A* **45**(2): 208-210.
KEARSLEY, S. K. (1990). "An algorithm for the simultatneaous superposition of a structural series." *Journal of Computational Chemestry* **11**(10): 1187-1192.

KENDREW, J. C., BODO, G., DINTZIS, H. M., PARRISH, R. G., WYCKOFF, H. et PHILLIPS, D. C. (1958). "A three-dimensional model of the myoglobin molecule obtained by x-ray analysis." *Nature* **181**(4610): 662-6.
KINCH, L. N., WRABL, J. O., KRISHNA, S. S., MAJUMDAR, I., SADREYEV, R. I., QI, Y., PEI, J., CHENG, H. et GRISHIN, N. V. (2003). "CASP5 assessment of fold recognition target predictions." *Proteins* **53 Suppl 6**: 395-409.
KLEYWEGT, G. J., ZHOU, J.-Y., KJELDGAARD, M. et JONES, T. A. (2006). *International Tables for Crystallography*.
KLINGENBERG, M. (1958). "Pigments of rat liver microsomes." *Arch Biochem Biophys* **75**(2): 376-86.
KOLARS, J. C., AWNI, W. M., MERION, R. M. et WATKINS, P. B. (1991). "First-pass metabolism of cyclosporin by the gut." *Lancet* **338**(8781): 1488-90.
KOSIOL, C. et GOLDMAN, N. (2005). "Different versions of the Dayhoff rate matrix." *Mol Biol Evol* **22**(2): 193-9.
KOYMANS, L., DONNE-OP DEN KELDER, G. M., KOPPELE TE, J. M. et VERMEULEN, N. P. (1993). "Cytochromes P450: their active-site structure and mechanism of oxidation." *Drug Metab Rev* **25**(3): 325-87.
KRISSINEL, E. et HENRICK, K. (2004). "Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions." *Acta Crystallogr D Biol Crystallogr* **60**(Pt 12 Pt 1): 2256-68.
KRUSKAL, J. B. et SANKOFF, D. (1983). "An anthology of algorithms and concepts for sequence comparison. in Time warps, string edits, and macromolecules: the theory and practice of sequence comparison." *Addison-Wesley, Reading, Mass.*: 265-310.
KUNTZ, I. D., BLANEY, J. M., OATLEY, S. J., LANGRIDGE, R. et FERRIN, T. E. (1982). "A geometric approach to macromolecule-ligand interactions." *J Mol Biol* **161**(2): 269-88.
KUPFERSCHMIDT, H. H., FATTINGER, K. E., HA, H. R., FOLLATH, F. et KRAHENBUHL, S. (1998). "Grapefruit juice enhances the bioavailability of the HIV protease inhibitor saquinavir in man." *Br J Clin Pharmacol* **45**(4): 355-9.

L

LAFITE, P. (2007). Etude du Cytochrome P450 2J2 Humain: Recherche de substrats et d'inhibiteurs sélectifs; Détermination de la topologie de son site actif, Université Paris Descartes.
LANDER, E. S., LINTON, L. M., BIRREN, B., NUSBAUM, C., ZODY, M. C., BALDWIN, J., DEVON, K., DEWAR, K., DOYLE, M., FITZHUGH, W., et al. (2001). "Initial sequencing and analysis of the human genome." *Nature* **409**(6822): 860-921.
LASKOWSKI, R. A., MACARTHUR, M. W., MOSS, D. S. et THORNTON, J. M. (1993). "PROCHECK: A program to check the stereochemical quality of protein structures." *J. Appl. Cryst.* **26**: 283-291.
LASKOWSKI, R. A., RULLMANNN, J. A., MACARTHUR, M. W., KAPTEIN, R. et THORNTON, J. M. (1996). "AQUA and PROCHECK-NMR: programs for checking the quality of protein structures solved by NMR." *J Biomol NMR* **8**(4): 477-86.
LAWRENCE, C. E., ALTSCHUL, S. F., BOGUSKI, M. S., LIU, J. S., NEUWALD, A. F. et WOOTTON, J. C. (1993). "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment." *Science* **262**(5131): 208-14.
LEVITT, M. (1976). "A simplified representation of protein conformations for rapid simulation of protein folding." *J Mol Biol* **104**(1): 59-107.
LEVITT, M. et WARSHEL, A. (1975). "Computer simulation of protein folding." *Nature* **253**(5494): 694-8.
LEWIS, D. F. (2001). *Guide to Cytochromes P450: Structure and Function*. New York, Taylor & Francis.
LEWIS, D. F., IOANNIDES, C. et PARKE, D. V. (1998). "Cytochromes P450 and species differences in xenobiotic metabolism and activation of carcinogen." *Environ Health Perspect* **106**(10): 633-41.
LEWIS, D. F. et PRATT, J. M. (1998). "The P450 catalytic cycle and oxygenation mechanism." *Drug Metab Rev* **30**(4): 739-86.
LOISEAU, N. (2002). Conception d'analogues structuraux d'un cyclopeptide modèle: Etude du mode de reconnaissance moléculaire par trois systèmes enzymatiques membranaires, Université Paris XI.
LUDEMANN, S. K., LOUNNAS, V. et WADE, R. C. (2000). "How do substrates enter and products exit the buried active site of cytochrome P450cam? 2. Steered molecular dynamics and adiabatic mapping of substrate pathways." *J Mol Biol* **303**(5): 813-30.
LUTHY, R., BOWIE, J. U. et EISENBERG, D. (1992). "Assessment of protein models with three-dimensional profiles." *Nature* **356**(6364): 83-5.

M

MACARTHUR, M. W. et THORNTON, J. M. (1996). "Deviations from planarity of the peptide bond in peptides and proteins." *J Mol Biol* **264**(5): 1180-95.

MACKERELL, A. D. J., BROOKS, B., BROOKS, C. L., III, NILSSON, L., ROUX, B., WON, Y. et KARPLUS, M. (1998). CHARMM: The Energy Function and Its Parameterization with an Overview of the Program. *The Encyclopedia of Computational Chemistry*. P. v. R. e. a. C. J. W. S. Ed. Schleyer. **1:** 271-277.

MADEJ, T., GIBRAT, J. F. et BRYANT, S. H. (1995). "Threading a database of protein cores." *Proteins* **23**(3): 356-69.

MAKRIS, T. M., DENISOV, I. G., SCHLICHTING, I. et SLIGRAR, S. G. (2005). "Activation of molecular oxygen by cytochrome P450." *Cytochrome P450: Structure, Mechanism and Biochemistry*: 149-182.

MANSUY, D. (1998). "The great diversity of reactions catalyzed by cytochromes P450." *Comp Biochem Physiol C Pharmacol Toxicol Endocrinol* **121**(1-3): 5-14.

MANSUY, D. et BATTIONI, P. (2000). "Diversity of reactions catalyzed by heme-thiolate proteins." *Biochemistry and binding: activation of small molecules* **4**: 1-15.

MANSUY, D., BATTIONI, P. et BATTIONI, J. P. (1989). "Chemical model systems for drug-metabolizing cytochrome-P-450-dependent monooxygenases." *Eur J Biochem* **184**(2): 267-85.

MANSUY, D. et RENAUD, J.-P. (1995). *Heme-thiolate proteins different from the cytichromes P450 catalysing monooxygenations*. New York, Plenum.

MARTI-RENOM, M. A., STUART, A. C., FISER, A., SANCHEZ, R., MELO, F. et SALI, A. (2000). "Comparative protein structure modeling of genes and genomes." *Annu Rev Biophys Biomol Struct* **29**: 291-325.

MCLACHLAN, A. D. (1979). "Gene duplication in the evolution of the yeast hexokinase active site." *Eur J Biochem* **100**(1): 181-7.

MCLACHLAN, A. D. (1982). "Rapid comparison of protein structures." *Acta Crystallographica Section A* **38**(6): 871-873.

MELO, F. et FEYTMANS, E. (1998). "Assessing protein structures with a non-local atomic interaction energy." *J Mol Biol* **277**(5): 1141-52.

MEYER, V. (2007). Detection d'homologies lointaines à faibles identités de séquences: Application aux protéines de la signalisation des dommages de l'ADN, Université Paris VII Denis Diderot.

MICHALSKY, E., GOEDE, A. et PREISSNER, R. (2003). "Loops In Proteins (LIP)--a comprehensive loop database for homology modelling." *Protein Eng* **16**(12): 979-85.

MITCHELL, E. M., ARTYMIUK, P. J., RICE, D. W. et WILLETT, P. (1990). "Use of techniques derived from graph theory to compare secondary structure motifs in proteins." *J Mol Biol* **212**(1): 151-66.

MIZUGUCHI, K., DEANE, C. M., BLUNDELL, T. L. et OVERINGTON, J. P. (1998). "HOMSTRAD: a database of protein structure alignments for homologous families." *Protein Sci* **7**(11): 2469-71.

MONTOLIU, C., SANCHO-TELLO, M., AZORIN, I., BURGAL, M., VALLES, S., RENAU-PIQUERAS, J. et GUERRI, C. (1995). "Ethanol increases cytochrome P4502E1 and induces oxidative stress in astrocytes." *J Neurochem* **65**(6): 2561-70.

MORAWIECKA, I. (2000). "Cisapride (Prepulsid): interactions with grapefruit and drugs." *Cmaj* **162**(1): 105-6, 109-10.

MOULT, J. (2005). "A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction." *Curr Opin Struct Biol* **15**(3): 285-9.

MOULT, J., FIDELIS, K., ROST, B., HUBBARD, T. et TRAMONTANO, A. (2005). "Critical assessment of methods of protein structure prediction (CASP)--round 6." *Proteins* **61 Suppl 7**: 3-7.

MUELLER, E., LOIDA, P. et SLIGAR, S. G. (1995). "Twentyfive years of P450cam Research: Mechanistic Insights into Oxygenase Catalysis." *Cytochrome P450*: 83-124.

MURATA, M., RICHARDSON, J. S. et SUSSMAN, J. L. (1985). "Simultaneous comparison of three protein sequences." *Proc Natl Acad Sci U S A* **82**(10): 3073-7.

MURZIN, A. G., BRENNER, S. E., HUBBARD, T. et CHOTHIA, C. (1995). "SCOP: a structural classification of proteins database for the investigation of sequences and structures." *J Mol Biol* **247**(4): 536-40.

N

NEBERT, D. W. et GONZALEZ, F. J. (1987). "P450 genes: structure, evolution, and regulation." *Annu Rev Biochem* **56**: 945-93.

NEBERT, D. W., MCKINNON, R. A. et PUGA, A. (1996). "Human drug-metabolizing enzyme polymorphisms: effects on risk of toxicity and cancer." *DNA Cell Biol* **15**(4): 273-80.

NEBERT, D. W., NELSON, D. R., ADESNIK, M., COON, M. J., ESTABROOK, R. W., GONZALEZ, F. J., GUENGERICH, F. P., GUNSALUS, I. C., JOHNSON, E. F., KEMPER, B., et al. (1989a). "The P450 superfamily: updated listing of all genes and recommended nomenclature for the chromosomal loci." *Dna* **8**(1): 1-13.

NEBERT, D. W., NELSON, D. R., COON, M. J., ESTABROOK, R. W., FEYEREISEN, R., FUJII-KURIYAMA, Y., GONZALEZ, F. J., GUENGERICH, F. P., GUNSALUS, I. C., JOHNSON, E. F., et al. (1991). "The P450 superfamily: update on new sequences, gene mapping, and recommended nomenclature." *DNA Cell Biol* **10**(1): 1-14.

NEBERT, D. W., NELSON, D. R. et FEYEREISEN, R. (1989b). "Evolution of the cytochrome P450 genes." *Xenobiotica* **19**(10): 1149-60.

NEDELCHEVA, V. et GUT, I. (1994). "P450 in the rat and man: methods of investigation, substrate specificities and relevance to cancer." *Xenobiotica* **24**(12): 1151-75.

NEEDLEMAN, S. B. et WUNSCH, C. D. (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins." *J Mol Biol* **48**(3): 443-53.

NELSON, D. R. (1998). "Metazoan cytochrome P450 evolution." *Comp Biochem Physiol C Pharmacol Toxicol Endocrinol* **121**(1-3): 15-22.

NELSON, D. R. (1999). "Cytochrome P450 and the individuality of species." *Arch Biochem Biophys* **369**(1): 1-10.

NELSON, D. R., KAMATAKI, T., WAXMAN, D. J., GUENGERICH, F. P., ESTABROOK, R. W., FEYEREISEN, R., GONZALEZ, F. J., COON, M. J., GUNSALUS, I. C., GOTOH, O., et al. (1993). "The P450 superfamily: update on new sequences, gene mapping, accession numbers, early trivial names of enzymes, and nomenclature." *DNA Cell Biol* **12**(1): 1-51.

NELSON, D. R., KOYMANS, L., KAMATAKI, T., STEGEMAN, J. J., FEYEREISEN, R., WAXMAN, D. J., WATERMAN, M. R., GOTOH, O., COON, M. J., ESTABROOK, R. W., et al. (1996). "P450 superfamily: update on new sequences, gene mapping, accession numbers and nomenclature." *Pharmacogenetics* **6**(1): 1-42.

NELSON, D. R. et STROBEL, H. W. (1987). "Evolution of cytochrome P-450 proteins." *Mol Biol Evol* **4**(6): 572-93.

NOVOTNY, J., BRUCCOLERI, R. et KARPLUS, M. (1984). "An analysis of incorrectly folded protein models. Implications for structure predictions." *J Mol Biol* **177**(4): 787-818.

NOVOTNY, J., RASHIN, A. A. et BRUCCOLERI, R. E. (1988). "Criteria that discriminate between native proteins and incorrectly folded models." *Proteins* **4**(1): 19-30.

O

ODA, A., YAMAOTSU, N. et HIRONO, S. (2005). "New AMBER force field parameters of heme iron for cytochrome P450s determined by quantum chemical calculations of simplified models." *J Comput Chem* **26**(8): 818-26.

OESTERHELD, J. R. (1998). "A review of developmental aspects of cytochrome P450." *J Child Adolesc Psychopharmacol* **8**(3): 161-74.

OKITA, R. T. et MASTERS, B. S. S. (1992). Biotransformations: the cytochromes P450. *Textbook of Biochemistry with Clinical Correlations*. T. M. D. Wiley-Liss. New York**:** 981-999.

OLDFIELD, T. J. (1992). "SQUID: a program for the analysis and display of data from crystallography and molecular dynamics." *J Mol Graph* **10**(4): 247-52.

OLDFIELD, T. J. et HUBBARD, R. E. (1994). "Analysis of C alpha geometry in protein structures." *Proteins* **18**(4): 324-37.

OLIVA, B., BATES, P. A., QUEROL, E., AVILES, F. X. et STERNBERG, M. J. (1997). "An automated classification of the structure of protein loops." *J Mol Biol* **266**(4): 814-30.

OMURA, T., ISHIMURA, Y. et FUJII-KURIYAMA, Y. (1993). *Cytochrome P450*. Tokyo, Kodensha.

OMURA, T. et SATO, R. (1962). "A new cytochrome in liver microsomes." *J Biol Chem* **237**: 1375-6.

ORENGO, C. A., MICHIE, A. D., JONES, S., JONES, D. T., SWINDELLS, M. B. et THORNTON, J. M. (1997). "CATH--a hierarchic classification of protein domain structures." *Structure* **5**(8): 1093-108.
ORTIZ DE MONTELLANO, P. R. (1986). *Cytochrome P450*. New York, Plenum.
ORTIZ DE MONTELLANO, P. R. (1995). *Cytochrome P450*. New York, 2nd edition, Plenum.
ORTIZ DE MONTELLANO, P. R. (2005). *Cytochrome P450: Structure, Mechanism and Biochemistry*. New York, 3rd edition, Plenum.
OSHIRO, C. M., KUNTZ, I. D. et DIXON, J. S. (1995). "Flexible ligand docking using a genetic algorithm." *J Comput Aided Mol Des* **9**(2): 113-30.
OTYEPKA, M., SKOPALIK, J., ANZENBACHEROVA, E. et ANZENBACHER, P. (2007). "What common structural features and variations of mammalian P450s are known to date?" *Biochim Biophys Acta* **1770**(3): 376-89.
OVERINGTON, J., JOHNSON, M. S., SALI, A. et BLUNDELL, T. L. (1990). "Tertiary structural constraints on protein evolutionary diversity: templates, key residues and structure prediction." *Proc Biol Sci* **241**(1301): 132-45.

P

PAINE, M. F., SCRUTTON, N., MUNRO, A., GUTIERREZ, A., ROBERTS, G. et WOLF, C. R. (2005). "Electron transfert partners of cytochromes P450." *Cytochrome P450: Structure, Mechanism and Biochemistry*: 115-148.
PAINE, M. F., SHEN, D. D., KUNZE, K. L., PERKINS, J. D., MARSH, C. L., MCVICAR, J. P., BARR, D. M., GILLIES, B. S. et THUMMEL, K. E. (1996). "First-pass metabolism of midazolam by the human intestine." *Clin Pharmacol Ther* **60**(1): 14-24.
PAPANDREOU, N., BEREZOVSKY, I. N., LOPES, A., ELIOPOULOS, E. et CHOMILIER, J. (2004). "Universal positions in globular proteins." *Eur J Biochem* **271**(23-24): 4762-8.
PARK, B. H., HUANG, E. S. et LEVITT, M. (1997). "Factors affecting the ability of energy functions to discriminate correct from incorrect folds." *J Mol Biol* **266**(4): 831-46.
PARKE, D. V. et IOANNIDES, C. (1994). "The effects of nutrition on chemical toxicity." *Drug Metab Rev* **26**(4): 739-65.
PATARD, L., STOVEN, V., GHARIB, B., BONTEMS, F., LALLEMAND, J. Y. et DE REGGI, M. (1996). "What function for human lithostathine? structural investigations by three-dimensional structure modeling and high-resolution NMR spectroscopy." *Protein Eng* **9**(11): 949-57.
PEARSON, W. R. (1991). "Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms." *Genomics* **11**(3): 635-50.
PEARSON, W. R. et LIPMAN, D. J. (1988). "Improved tools for biological sequence comparison." *Proc Natl Acad Sci U S A* **85**(8): 2444-8.
PERNECKY, S. J., LARSON, J. R., PHILPOT, R. M. et COON, M. J. (1993). "Expression of truncated forms of liver microsomal P450 cytochromes 2B4 and 2E1 in Escherichia coli: influence of NH2-terminal region on localization in cytosol and membranes." *Proc Natl Acad Sci U S A* **90**(7): 2651-5.
PESSAYRE, D. (1995). "[Mechanism of drug-induced hepatitis. A consequence of superposition of two xenobiotic systems]." *Gastroenterol Clin Biol* **19**(5 Pt 2): B47-56.
PETREK, M., OTYEPKA, M., BANAS, P., KOSINOVA, P., KOCA, J. et DAMBORSKY, J. (2006). "CAVER: a new tool to explore routes from protein clefts, pockets and cavities." *BMC Bioinformatics* **7**: 316.
PINOT, F., BENVENISTE, I., SALAUN, J. P., LOREAU, O., NOEL, J. P., SCHREIBER, L. et DURST, F. (1999). "Production in vitro by the cytochrome P450 CYP94A1 of major C18 cutin monomers and potential messengers in plant-pathogen interactions: enantioselectivity studies." *Biochem J* **342 (Pt 1)**: 27-32.
PONTIUS, J., RICHELLE, J. et WODAK, S. J. (1996). "Deviations from standard atomic volumes as a quality measure for protein crystal structures." *J Mol Biol* **264**(1): 121-36.
PORTER, T. D. et COON, M. J. (1991). "Cytochrome P-450. Multiplicity of isoforms, substrates, and catalytic and regulatory mechanisms." *J Biol Chem* **266**(21): 13469-72.
POULOS, T. L., FINZEL, B. C., GUNSALUS, I. C., WAGNER, G. C. et KRAUT, J. (1985). "The 2.6-A crystal structure of Pseudomonas putida cytochrome P-450." *J Biol Chem* **260**(30): 16122-30.
POULOS, T. L., FINZEL, B. C. et HOWARD, A. J. (1987). "High-resolution crystal structure of cytochrome P450cam." *J Mol Biol* **195**(3): 687-700.
POULOS, T. L. et JOHNSON, E. F. (2005). Structures of cytochromes P450 enzymes. *Cytochrome P450: Structure, Mechanism and Biochemistry*. O. d. Montellano. New York, Plenum**:** 87-114.

PRICE-EVANS, D. A. (1993). *Genetic Factors in Drug Therapy*. Cambridge, Cambridge University Press.
PROJEAN, D., MORIN, P. E., TU, T. M. et DUCHARME, J. (2003). "Identification of CYP3A4 and CYP2C8 as the major cytochrome P450 s responsible for morphine N-demethylation in human liver microsomes." *Xenobiotica* **33**(8): 841-54.
PUGA, A., NEBERT, D. W., MCKINNON, R. A. et MENON, A. G. (1997). "Genetic polymorphisms in human drug-metabolizing enzymes: potential uses of reverse genetics to identify genes of toxicological relevance." *Crit Rev Toxicol* **27**(2): 199-222.

R

RAMACHANDRAN, G. N., RAMAKRISHNAN, C. et SASISEKHARAN, V. (1963). "Stereochemistry of polypeptide chain configurations." *J Mol Biol* **7**: 95-9.
RAO, S. T. et ROSSMANN, M. G. (1973). "Comparison of super-secondary structures in proteins." *J Mol Biol* **76**(2): 241-56.
RENDIC, S. (2002). "Summary of information on human CYP enzymes: human P450 metabolism data." *Drug Metab Rev* **34**(1-2): 83-448.
RENDIC, S. et DI CARLO, F. J. (1997). "Human cytochrome P450 enzymes: a status report summarizing their reactions, substrates, inducers, and inhibitors." *Drug Metab Rev* **29**(1-2): 413-580.
RICHARDSON, J. S. (1981). "The anatomy and taxonomy of protein structure." *Adv Protein Chem* **34**: 167-339.
RISLER, J. L., DELORME, M. O., DELACROIX, H. et HENAUT, A. (1988). "Amino acid substitutions in structurally related proteins. A pattern recognition approach. Determination of a new and efficient scoring matrix." *J Mol Biol* **204**(4): 1019-29.
ROSE, G. D. (1979). "Hierarchic organization of domains in globular proteins." *J Mol Biol* **134**(3): 447-70.
ROWLAND, P., BLANEY, F. E., SMYTH, M. G., JONES, J. J., LEYDON, V. R., OXBROW, A. K., LEWIS, C. J., TENNANT, M. G., MODI, S., EGGLESTON, D. S., et al. (2006). "Crystal structure of human cytochrome P450 2D6." *J Biol Chem* **281**(11): 7614-22.
RUCKPAUL, K. et REIN, H. (1984). *Cytochrome P450*. Berlin, Akademie-Verlag.
RUFINO, S. D., DONATE, L. E., CANARD, L. H. et BLUNDELL, T. L. (1997). "Predicting the conformational class of short and medium size loops connecting regular secondary structures: application to comparative modelling." *J Mol Biol* **267**(2): 352-67.
RUSSELL, R. B. et BARTON, G. J. (1992). "Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels." *Proteins* **14**(2): 309-23.

S

SALI, A. et BLUNDELL, T. L. (1990). "Definition of general topological equivalence in protein structures. A procedure involving comparison of properties and relationships through simulated annealing and dynamic programming." *J Mol Biol* **212**(2): 403-28.
SALI, A. et BLUNDELL, T. L. (1993). "Comparative protein modelling by satisfaction of spatial restraints." *J Mol Biol* **234**(3): 779-815.
SANGER, F., COULSON, A. R., HONG, G. F., HILL, D. F. et PETERSEN, G. B. (1982). "Nucleotide sequence of bacteriophage lambda DNA." *J Mol Biol* **162**(4): 729-73.
SANKOFF, D., CEDERGREN, R. J. et LAPALME, G. (1976). "Frequency of insertion-deletion, transversion, and transition in the evolution of 5S ribosomal RNA." *J Mol Evol* **7**(2): 133-49.
SCHAFFER, A. A., ARAVIND, L., MADDEN, T. L., SHAVIRIN, S., SPOUGE, J. L., WOLF, Y. I., KOONIN, E. V. et ALTSCHUL, S. F. (2001). "Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements." *Nucleic Acids Res* **29**(14): 2994-3005.
SCHENKMAN, J. B., FREY, I., REMMER, H. et ESTABROOK, R. W. (1967). "Sex differences in drug metabolism by rat liver microsomes." *Mol Pharmacol* **3**(6): 516-25.
SCHENKMANN, J. B. et GRIEM, H. (1993). *Cytochrome P450*. Berlin, Springer-Verlag.
SCHLEINKOFER, K., SUDARKO, WINN, P. J., LUDEMANN, S. K. et WADE, R. C. (2005). "Do mammalian cytochrome P450s show multiple ligand access pathways and ligand channelling?" *EMBO Rep* **6**(6): 584-9.

SCHLICHTING, I., BERENDZEN, J., CHU, K., STOCK, A. M., MAVES, S. A., BENSON, D. E., SWEET, R. M., RINGE, D., PETSKO, G. A. et SLIGAR, S. G. (2000). "The catalytic pathway of cytochrome p450cam at atomic resolution." *Science* **287**(5458): 1615-22.

SCHUETTELKOPF, A. W. et VAN AALTEN, D. M. F. (2004). "PRODRG - a tool for high-throughput crystallography of protein-ligand complexes." *Acta Crystallographica Section A* **D60**: 355-1363.

SCHWARTZ, R. M. et DAYHOFF, M. O. (1979). "Matrices for detecting distant relationships." *Atlas of Protein Structure* **5**(3): 353-358.

SCHWARZ, D. (1991). "Rotational motion and membrane topology of the microsomal cytochrome P450 system as analyzed by saturation transfer EPR." *Frontiers in Biotransformation* **5**: 94-137.

SCHWEDE, T., KOPP, J., GUEX, N. et PEITSCH, M. C. (2003). "SWISS-MODEL: An automated protein homology-modeling server." *Nucleic Acids Res* **31**(13): 3381-5.

SCOTT, E. E., HE, Y. A., WESTER, M. R., WHITE, M. A., CHIN, C. C., HALPERT, J. R., JOHNSON, E. F. et STOUT, C. D. (2003). "An open conformation of mammalian cytochrome P450 2B4 at 1.6-A resolution." *Proc Natl Acad Sci U S A* **100**(23): 13196-201.

SCOTT, E. E., WHITE, M. A., HE, Y. A., JOHNSON, E. F., STOUT, C. D. et HALPERT, J. R. (2004). "Structure of mammalian cytochrome P450 2B4 complexed with 4-(4-chlorophenyl)imidazole at 1.9-A resolution: insight into the range of P450 conformations and the coordination of redox partner binding." *J Biol Chem* **279**(26): 27294-301.

SCOTT, J. G., LIU, N. et WEN, Z. (1998). "Insect cytochromes P450: diversity, insecticide resistance and tolerance to plant toxins." *Comp Biochem Physiol C Pharmacol Toxicol Endocrinol* **121**(1-3): 147-55.

SCOTT, W. R. P., HÜNENBERGER, P. H., TIRONI, I. G., MARK, A. E., BILLETER, S. R., FENNEN, J., TORDA, A. E., HUBER, T., KRÜGER, P. et VAN GUNSTEREN, W. F. (1999). "The GROMOS biomolecular simulation program package." *J. Phys. Chem.* **103**: 3596–3607.

SHAPIRO, A., BOTHA, J. D., PASTORE, A. et LESK, A. M. (1992). "A method for multiple superposition of structures." *Acta Crystallogr A* **48 (Pt 1)**: 11-4.

SHAPIRO, J. et BRUTLAG, D. (2004). "FoldMiner and LOCK 2: protein structure comparison and motif discovery on the web." *Nucleic Acids Res* **32**(Web Server issue): W536-41.

SHATSKY, M., NUSSINOV, R. et WOLFSON, H. J. (2002). "Flexible protein alignment and hinge detection." *Proteins* **48**(2): 242-56.

SHATSKY, M., NUSSINOV, R. et WOLFSON, H. J. (2004). "FlexProt: alignment of flexible protein structures without a predefinition of hinge regions." *J Comput Biol* **11**(1): 83-106.

SHINDYALOV, I. N. et BOURNE, P. E. (1998). "Protein structure alignment by incremental combinatorial extension (CE) of the optimal path." *Protein Eng* **11**(9): 739-47.

SINGH, A. P. et BRUTLAG, D. L. (1997). "Hierarchical protein structure superposition using both secondary structure and atomic representations." *Proc Int Conf Intell Syst Mol Biol* **5**: 284-93.

SIPPL, M. J. (1990). "Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins." *J Mol Biol* **213**(4): 859-83.

SIPPL, M. J. (1993). "Recognition of errors in three-dimensional structures of proteins." *Proteins* **17**(4): 355-62.

SIPPL, M. J., LACKNER, P., DOMINGUES, F. S., PRLIC, A., MALIK, R., ANDREEVA, A. et WIEDERSTEIN, M. (2001). "Assessment of the CASP4 fold recognition category." *Proteins* **Suppl 5**: 55-67.

SIPPL, M. J. et STEGBUCHNER, H. (1991). "Superposition of three-dimensioal objects: A fast and numerically stable algorithm for the calculation fo the matrix of optimal rotation." *Computers Chemestry* **15**(1): 73-78.

SMITH, D. A. (1991). "Species differences in metabolism and pharmacokinetics: are we close to an understanding?" *Drug Metab Rev* **23**(3-4): 355-73.

SMITH, T. F. et WATERMAN, M. S. (1981). "Identification of common molecular subsequences." *J Mol Biol* **147**(1): 195-7.

SODING, J., BIEGERT, A. et LUPAS, A. N. (2005). "The HHpred interactive server for protein homology detection and structure prediction." *Nucleic Acids Res* **33**(Web Server issue): W244-8.

SOUCEK, P. et GUT, I. (1992). "Cytochromes P-450 in rats: structures, functions, properties and relevant human forms." *Xenobiotica* **22**(1): 83-103.

STEGEMAN, J. J. et LIVINGSTONE, D. R. (1998). "Forms and functions of cytochrome P450." *Comp Biochem Physiol C Pharmacol Toxicol Endocrinol* **121**(1-3): 1-3.

STIER, A. (1976). "Lipid structure and drug metabolizing enzymes." *Biochem Pharmacol* **25**(2): 109-13.

STRYER, L. (1994). *Biochemestry*. New York, W.H. Freeman and Company.

STURROCK, S. S. et COLLINS, J. F. (1994). "MPsrch version 1.4. Edinburgh: University of Edinburgh Biocomputing Research Unit."

SUTCLIFFE, M. J., HANEEF, I., CARNEY, D. et BLUNDELL, T. L. (1987). "Knowledge based modelling of homologous proteins, Part I: Three-dimensional frameworks derived from the simultaneous superposition of multiple structures." *Protein Eng* **1**(5): 377-84.

T

TAKAYA, N., SUZUKI, S., KUWAZAKI, S., SHOUN, H., MARUO, F., YAMAGUCHI, M. et TAKEO, K. (1999). "Cytochrome p450nor, a novel class of mitochondrial cytochrome P450 involved in nitrate respiration in the fungus Fusarium oxysporum." *Arch Biochem Biophys* **372**(2): 340-6.

TAYLOR, W. R. (1987). "Multiple sequence alignment by a pairwise algorithm." *Comput Appl Biosci* **3**(2): 81-7.

TAYLOR, W. R. (1999). "Protein structure comparison using iterated double dynamic programming." *Protein Sci* **8**(3): 654-65.

TAYLOR, W. R. et ORENGO, C. A. (1989a). "A holistic approach to protein structure alignment." *Protein Eng* **2**(7): 505-19.

TAYLOR, W. R. et ORENGO, C. A. (1989b). "Protein structure alignment." *J Mol Biol* **208**(1): 1-22.

THOMPSON, J. D., HIGGINS, D. G. et GIBSON, T. J. (1994). "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice." *Nucleic Acids Res* **22**(22): 4673-80.

THOMPSON, J. D., PLEWNIAK, F. et POCH, O. (1999). "BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs." *Bioinformatics* **15**(1): 87-8.

THORNTON, J. M., JONES, D. T., MACARTHUR, M. W., ORENGO, C. M. et SWINDELLS, M. B. (1995). "Protein folds: towards understanding folding from inspection of native structures." *Philos Trans R Soc Lond B Biol Sci* **348**(1323): 71-9.

TOPHAM, C. M., SRINIVASAN, N., THORPE, C. J., OVERINGTON, J. P. et KALSHEKER, N. A. (1994). "Comparative modelling of major house dust mite allergen Der p I: structure validation using an extended environmental amino acid propensity table." *Protein Eng* **7**(7): 869-94.

TSUTSUMI, M., MATSUDA, Y. et TAKADA, A. (1993). "Role of ethanol-inducible cytochrome P-450 2E1 in the development of hepatocellular carcinoma by the chemical carcinogen, N-nitrosodimethylamine." *Hepatology* **18**(6): 1483-9.

V

VAN AALTEN, D. M., BYWATER, R., FINDLAY, J. B., HENDLICH, M., HOOFT, R. W. et VRIEND, G. (1996). "PRODRG, a program for generating molecular topologies and unique molecular descriptors from coordinates of small molecules." *J Comput Aided Mol Des* **10**(3): 255-62.

VENCLOVAS, C. (2003). "Comparative modeling in CASP5: progress is evident, but alignment errors remain a significant hindrance." *Proteins* **53 Suppl 6**: 380-8.

VENTER, J. C., ADAMS, M. D., MYERS, E. W., LI, P. W., MURAL, R. J., SUTTON, G. G., SMITH, H. O., YANDELL, M., EVANS, C. A., HOLT, R. A., et al. (2001). "The sequence of the human genome." *Science* **291**(5507): 1304-51.

VON RICHTER, O., GREINER, B., FROMM, M. F., FRASER, R., OMARI, T., BARCLAY, M. L., DENT, J., SOMOGYI, A. A. et EICHELBAUM, M. (2001). "Determination of in vivo absorption, metabolism, and transport of drugs by the human intestinal wall and liver with a novel perfusion technique." *Clin Pharmacol Ther* **70**(3): 217-27.

VON WACHENFELDT, C., RICHARDSON, T. H., COSME, J. et JOHNSON, E. F. (1997). "Microsomal P450 2C3 is expressed as a soluble dimer in Escherichia coli following modification of its N-terminus." *Arch Biochem Biophys* **339**(1): 107-14.

VRIEND, G. (1990). "WHAT IF: a molecular modeling and drug design program." *J Mol Graph* **8**(1): 52-6, 29.

VRIEND, G. et SANDER, C. (1991). "Detection of common three-dimensional substructures in proteins." *Proteins* **11**(1): 52-8.

W

WADE, R. C., WINN, P. J., SCHLICHTING, I. et SUDARKO (2004). "A survey of active site access channels in cytochromes P450." *J Inorg Biochem* **98**(7): 1175-82.

WALKER, C. H. (1998). "Avian forms of cytochrome P450." *Comp Biochem Physiol C Pharmacol Toxicol Endocrinol* **121**(1-3): 65-72.

WANG, M., ROBERTS, D. L., PASCHKE, R., SHEA, T. M., MASTERS, B. S. et KIM, J. J. (1997). "Three-dimensional structure of NADPH-cytochrome P450 reductase: prototype for FMN- and FAD-containing enzymes." *Proc Natl Acad Sci U S A* **94**(16): 8411-6.

WANG, Y. J., YU, C. F., CHEN, L. C., CHEN, C. H., LIN, J. K., LIANG, Y. C., LIN, C. H., LIN, S. Y., CHEN, C. F. et HO, Y. S. (2002). "Ketoconazole potentiates terfenadine-induced apoptosis in human Hep G2 cells through inhibition of cytochrome p450 3A4 activity." *J Cell Biochem* **87**(2): 147-59.

WATERMAN, M. R. (1992). "Cytochrome P450: cellular distribution and structural considerations." *Current Opinion in Structural Biology* **2**: 384-387.

WELTMAN, M. D., FARRELL, G. C. et LIDDLE, C. (1996). "Increased hepatocyte CYP2E1 expression in a rat nutritional model of hepatic steatosis with inflammation." *Gastroenterology* **111**(6): 1645-53.

WESTER, M. R., JOHNSON, E. F., MARQUES-SOARES, C., DANSETTE, P. M., MANSUY, D. et STOUT, C. D. (2003a). "Structure of a substrate complex of mammalian cytochrome P450 2C5 at 2.3 A resolution: evidence for multiple substrate binding modes." *Biochemistry* **42**(21): 6370-9.

WESTER, M. R., JOHNSON, E. F., MARQUES-SOARES, C., DIJOLS, S., DANSETTE, P. M., MANSUY, D. et STOUT, C. D. (2003b). "Structure of mammalian cytochrome P450 2C5 complexed with diclofenac at 2.1 A resolution: evidence for an induced fit model of substrate binding." *Biochemistry* **42**(31): 9335-45.

WICKRAMASINGHE, R. H. et VILLE, C. A. (1975). "Early role during chemical evolution for cytochrome P450 in oxygen detoxification." *Nature* **256**: 509-511.

WILLIAMS, J. A., MARTIN, F. L., MUIR, G. H., HEWER, A., GROVER, P. L. et PHILLIPS, D. H. (2000a). "Metabolic activation of carcinogens and expression of various cytochromes P450 in human prostate tissue." *Carcinogenesis* **21**(9): 1683-9.

WILLIAMS, P. A., COSME, J., SRIDHAR, V., JOHNSON, E. F. et MCREE, D. E. (2000b). "Microsomal cytochrome P450 2C5: comparison to microbial P450s and unique features." *J Inorg Biochem* **81**(3): 183-90.

WILLIAMS, P. A., COSME, J., SRIDHAR, V., JOHNSON, E. F. et MCREE, D. E. (2000c). "Mammalian microsomal cytochrome P450 monooxygenase: structural adaptations for membrane binding and functional diversity." *Mol Cell* **5**(1): 121-31.

WILLIAMS, P. A., COSME, J., WARD, A., ANGOVE, H. C., MATAK VINKOVIC, D. et JHOTI, H. (2003). "Crystal structure of human cytochrome P450 2C9 with bound warfarin." *Nature* **424**(6947): 464-8.

WINN, P. J., LUDEMANN, S. K., GAUGES, R., LOUNNAS, V. et WADE, R. C. (2002). "Comparison of the dynamics of substrate access channels in three cytochrome P450s reveals different opening mechanisms and a novel functional role for a buried arginine." *Proc Natl Acad Sci U S A* **99**(8): 5361-6.

WOJCIK, J., MORNON, J. P. et CHOMILIER, J. (1999). "New efficient statistical sequence-dependent structure prediction of short to medium-sized protein loops based on an exhaustive loop classification." *J Mol Biol* **289**(5): 1469-90.

WOLFF, T. et STRECKER, M. (1992). "Endogenous and exogenous factors modifying the activity of human liver cytochrome P-450 enzymes." *Exp Toxicol Pathol* **44**(5): 263-71.

Y

YU, C. et GUNSALUS, I. C. (1970). "Crystalline cytochrome P-450cam." *Biochem Biophys Res Commun* **40**(6): 1431-6.

Z

ZHANG, Y. et SKOLNICK, J. (2005). "TM-align: a protein structure alignment algorithm based on the TM-score." *Nucleic Acids Res* **33**(7): 2302-9.

ZHAO, Q., MODI, S., SMITH, G., PAINE, M., MCDONAGH, P. D., WOLF, C. R., TEW, D., LIAN, L. Y., ROBERTS, G. C. et DRIESSEN, H. P. (1999). "Crystal structure of the FMN-binding domain of human cytochrome P450 reductase at 1.93 A resolution." *Protein Sci* **8**(2): 298-306.

ZHAO, Y., WHITE, M. A., MURALIDHARA, B. K., SUN, L., HALPERT, J. R. et STOUT, C. D. (2006). "Structure of microsomal cytochrome P450 2B4 complexed with the antifungal drug bifonazole: insight into P450 conformational plasticity and membrane interaction." *J Biol Chem* **281**(9): 5973-81.

ZIMMERMANN, K. (1991). "ORAL: All Purpose Molecular Mechanics Simulator and Energy Minimizer." *J. Comput. Chem.* **12**: 310-319.

Annexes

ANNEXE 1

Le monde merveilleux des protéines

« Pour réaliser quelque chose de vraiment extraordinaire,
commencez par la rêver »
Walt Disney (1901 – 1966 ap JC)

1.1 Généralités

Les protéines sont parmi les principaux types de molécules du vivant (les autres étant les acides nucléiques, les lipides et les glucides). Dans la cellule elles assurent la grande majorité des fonctions enzymatiques, et une bonne part des fonctions de maintien de la structure et de transport. Ces protéines disposent d'une organisation à quatre niveaux :

- **Structure primaire** : séquence d'acides aminés ;
- **Structure secondaire** : répartition de certaines structures locales régulières nommées structures secondaires (hélice α et feuillet β) ;
- **Structure tertiaire** : organisation dans l'espace comprenant l'interaction entre eux des éléments de structure secondaire ;
- **Structure quaternaire** : organisation spatiale de monomères (protéines multimériques).

1.2 Les composants de la protéine : les acides aminés

1.2.1 Structure générale d'un acide aminé

Au nombre de 20 (si on ne compte que les acides aminés naturels terrestres), les acides aminés sont les constituants de la protéine (ou du peptide). Ils sont composés d'un atome de carbone central (C_α) chiral le plus souvent (exception de la glycine), lié à un groupe aminé (NH_2), à un groupe carboxylique (COOH), à un atome d'hydrogène (H_α) et à un des 20 groupements chimiques différents appelés chaînes latérales (cf. Figure A 1-1 et Figure A 1-2).

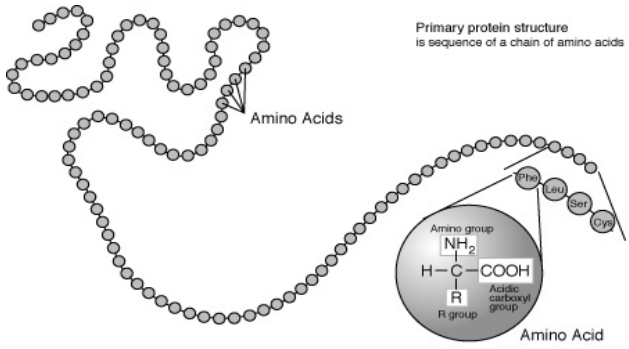


Figure A 1-1 Schéma simplifié d'une séquence primaire d'une protéine. R correspond à n'importe quelle chaîne latérale parmi les 20 existantes. Les acides aminés sont reliés entre eux par des liaisons peptidiques. (source : **http://www2.lifl.fr/~touzet/BI/TP2**)

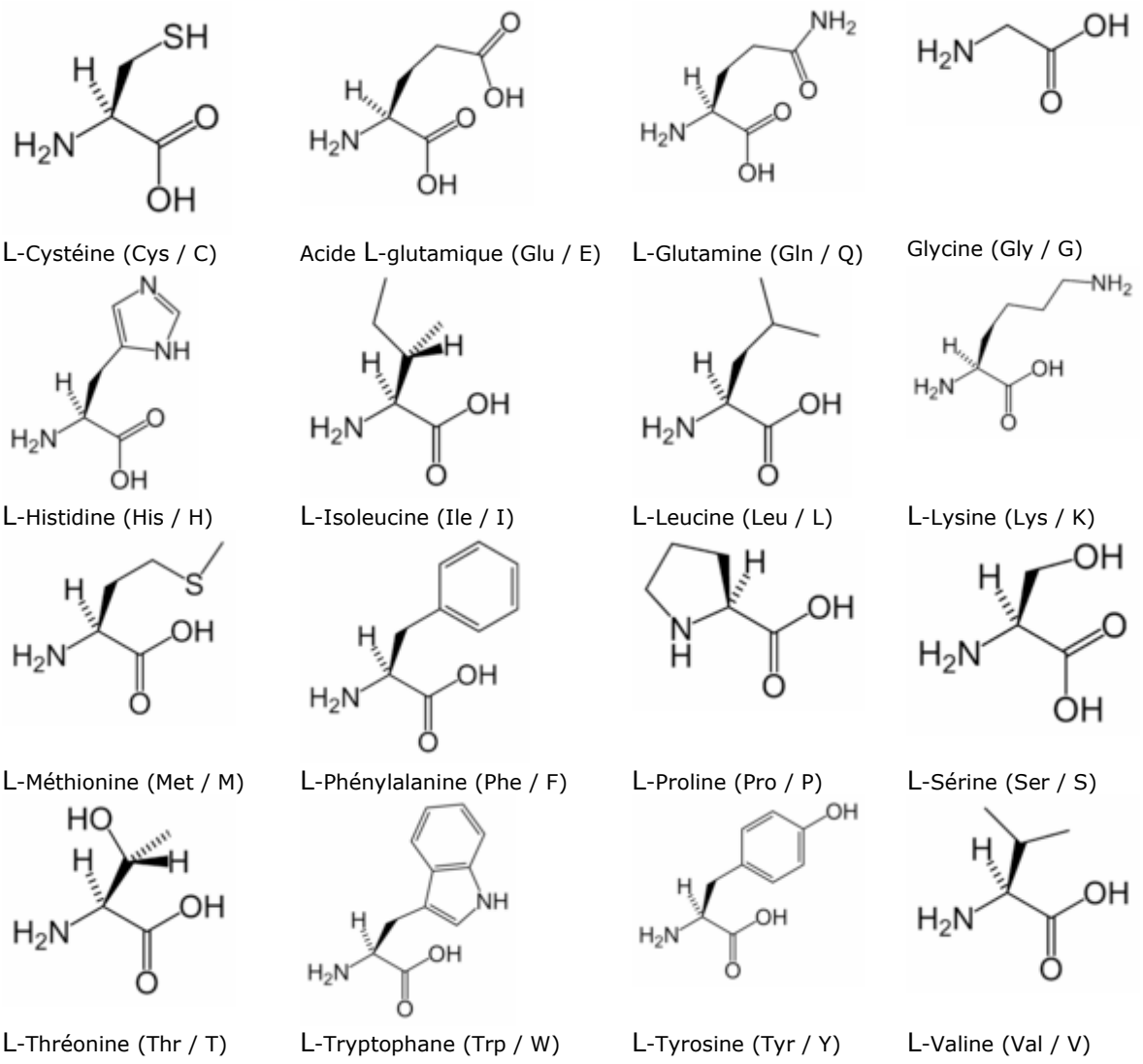


Figure A 1-2 Formules chimiques des 20 acides aminés avec leur code à 3 lettres et à une lettre.

Les résidus et leurs fonctions chimiques différentes confèrent aux protéines leur diversité fonctionnelle. Les protéines sont des chaînes polypeptidiques où le groupe α-carboxylique d'un acide aminé est relié au groupe α-aminé de l'amino-acide suivant par une liaison peptidique (*–CO–NH–*). Les protéines naturelles sont généralement constituées de 50 à 2000 résidus d'acide aminé (Stryer, 1994). La chaîne non ramifiée des résidus est polarisée de l'extrémité aminée (extrémité NH_2 terminale) à l'extrémité carboxy-terminale (*COOH*). La chaîne d'atome répétant régulièrement les liaisons peptidiques se nomme le squelette peptidique (ou carboné si seuls les C_α sont pris en compte). La liaison peptidique est rigide et plane en raison du caractère partiel de double liaison *–CO–NH–*, mais des rotations sont possibles autour des autres liaisons (C_α–*CO* et *NH*– C_α).

1.2.2 Propriétés générales des acides aminés

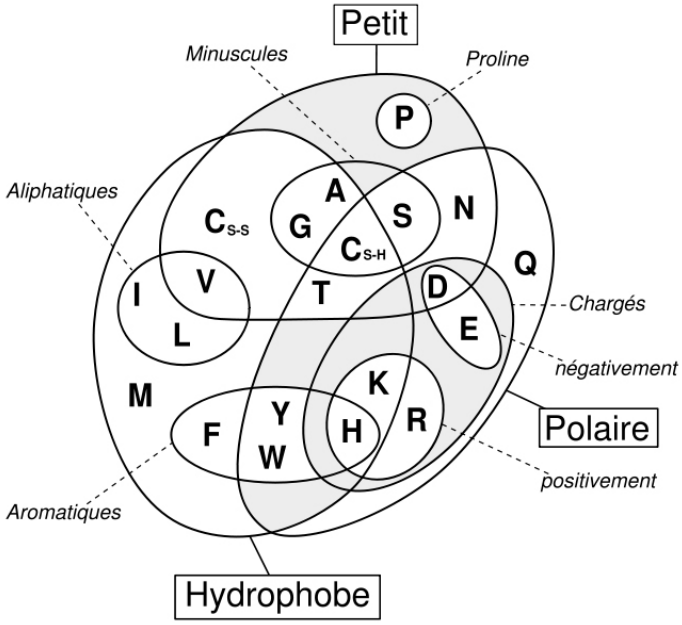


Figure A 1-3 Diagramme de Venn des propriétés des acides aminés.

1.2.2.1 Solubilité

La plupart des acides aminés subissent facilement la solvatation par les solvants polaires tels que l'eau, ou l'alcool (particulièrement proline et hydroxyproline) dans lesquels ils sont solubles. D'autre part, les acides α-aminés sont solubles, mais à moindre degré dans les solvants non polaires. Il est important de retenir que cette solubilité est largement dépendante des propriétés de la chaîne latérale: la solubilité diminue avec le nombre d'atomes de carbone du radical, mais inversement augmente si ce radical R est porteur de fonctions polaires (NH2, COOH) ou hydrophiles (OH).

1.2.2.2 Propriétés ioniques

Les acides aminés contiennent un groupement carboxyle -COOH acide et un groupement amino $-NH_2$ basique. En solution, ces groupements existent sous deux formes, l'une chargée, l'autre neutre :

$$R\text{-}COOH \; ; \; R\text{-}COO^{-} + H^{+}$$

$$R\text{-}NH3^{+} \; ; \; R\text{-}NH2 + H^{+}$$

Les acides aminés sont appelés pour cette structure diionique amphotères. L'ionisation varie avec le pH : les acides aminés existent, en solution aqueuse, sous 3 formes possibles :

a) en milieu acide : La fonction amine s'ionise en captant un proton et la dissociation du carboxyle est inhibée. L'acide aminé se trouve sous forme de cation.

b) en milieu basique La fonction acide s'ionise en libérant un proton, la base du milieu bloque l'ionisation du groupement amino. L'acide aminé se trouve sous forme d'anion.

c) Le pH pour lequel les 2 dissociations s'effectuent est appelé point isoélectrique : ou pH_i. À ce pH, on a un ion dipolaire ou zwitterion de charge nette nulle, donc ne migrant pas dans un champ électrique.

De part et d'autre du pH_i, on définit des pH qui correspondent à une demi-dissociation de COOH et de NH_3^+, ce sont les pK_s. Il existe donc 2 pK :

- le pK de COOH (environ 2 – 3)
- le pK de NH_3^+ (environ 10)

Le point (ou pH) isoélectrique ou isoionique est égal à la demi somme des pKs. Le radical R, lorsqu'il renferme un groupe ionisable, participe à la valeur du point isoélectrique. Un pK supplémentaire apparaît alors. Par exemple pour l'histidine :

- pK_1 acide
- pK_2 demi dissociation du groupe imidazole
- pK_3 amine

1.2.2.3 Propriétés des chaînes latérales

Les interactions entre atomes de chaînes latérales et celles entre atomes du squelette polypeptidique sont responsables du repliement des protéines et de sa stabilité. On observe trois types de liaisons non covalentes dans les protéines : (a) les liaisons hydrogènes, (b) les interactions de Van der Waals et (c) les interactions électrostatiques. La **liaison hydrogène** est formée lorsqu'un atome d'hydrogène est partagé entre un donneur d'hydrogène (l'atome lié covalemment à l'hydrogène) et un accepteur d'hydrogène. Les **liaisons électrostatiques** interviennent entre atomes chargés positivement et négativement. Enfin, les **liaisons de Van der Waals** sont des liaisons de type dipôle induit – dipôle induit : une asymétrie passagère de charge autour d'un atome (possible parce que la distribution électronique de charge autour d'un atome évolue avec le temps) induit une asymétrie opposée dans un atome adjacent : ils s'attirent alors mutuellement. Les liaisons de Van der Waals sont plus faibles que les liaisons hydrogène ou électrostatiques, mais elles sont efficaces en grand nombre. Ces liaisons sont les principales responsables du repliement des protéines.

Enfin, l'encombrement stérique joue également un rôle dans la structure des protéines car lorsque la chaîne protéique s'enroule, une chaîne latérale encombrante peut empêcher une courbure forte. La répulsion des atomes est provoquée par le recouvrement des nuages électroniques lorsque les atomes se rapprochent. Les liaisons covalentes inter-résidus – sous forme de points disulfures entre les résidus cystéines – permettent une meilleure stabilité pour les protéines qui sortent de l'environnement cellulaire (Brandon et Tooze, 1999).

1.3 Propriétés structurales des protéines

1.3.1 Détermination

1.3.1.1 La Radiocristallographie aux rayons X

La cristallographie aux RX est la méthode de choix pour résoudre la structure atomique des protéines. La première structure déterminée par cristallographie fut la myoglobine : elle a été déterminée par M. Perutz et Sir J.C Kendrew en 1958 (Kendrew et *al.*, 1958, Prix Nobel de Chimie en 1959). Cette méthode s'appuie sur la diffraction des RX par les électrons dans un cristal. Le principe repose sur un concept simple : les RX diffractés par les différents « plans » réticulaire du cristal seront en phase à différents angles d'incidence selon la longueur d'onde du rayonnement (obtention de la distance entre ces plans par interférence). Le nuage électronique des atomes lourds (la « densité électronique ») est déterminé par cette méthode. La seconde étape consiste à placer les atomes correctement dans cette densité électronique par des techniques de modélisation moléculaire. La limitation la plus importante de la cristallographie est l'obtention d'un cristal de protéine, opération difficile voir impossible, notamment pour les protéines membranaires pour les protéines flexibles.

1.3.1.2 La Résonance Magnétique Nucléaire

La RMN permet d'obtenir la structure des protéines en solution. Plusieurs paramètres géométriques peuvent être calculés sur la structure protéique, comme les angles de torsion par la mesure du couplage scalaire, ou bien les distances entre atomes (hydrogène généralement) *via* la mesure du couplage dipolaire (Nuclear Overhauser Effect ou NOE). Ces dernières mesures étant sensibles à la dynamique de la protéine, elles peuvent donner une indication sur celle-ci. Les contraintes de distances entre atomes obtenues par RMN sont utilisées lors de calculs de dynamique moléculaire afin d'approcher la structure de la protéine (et sa dynamique). Une des limitations de la RMN porte sur la taille de la protéine à analyser.

1.3.2 Arrangement structural

1.3.2.1 Le repliement

En théorie, une protéine pourrait adopter plusieurs conformations mais concrètement, la plupart se replie spontanément dans une forme stable particulière et unique. Cette forme particulière provient du fait que les groupes polaires du squelette peptidique et les chaînes latérales interagissent entre eux et aussi avec l'eau. Ainsi, certaines conformations ont plus d'interactions stabilisantes que d'autres et sont donc favorisées (Alberts et *al.*, 1994). Le paradigme du rapport entre la séquence protéique et sa structure tridimensionnelle provient des études de C. Anfinsen sur la ribonucléase (Anfinsen, 1973). Il a déterminé qu'une protéine repliée aléatoirement est inactive et que les protéines isolées en solution peuvent retrouver leur conformation active originale après dénaturation. La conclusion était donc que toute l'information nécessaire au repliement d'une protéine devait être inhérente à l'ordre des acides aminés (Alberts et *al.*, 1994). D'autres études ont également tiré les mêmes conclusions, menant à la théorie générale que la séquence des acides aminés d'une protéine spécifie sa conformation (Stryer, 1994). Il est intéressant de noter que sur l'ensemble des séquences possibles, la population des différents repliements est limitée, 9 repliements représentent 46% des protéines d'une base de données non redondante (Thornton et *al.*, 1995). Pour résumer, en environnement aqueux, le repliement des protéines est conduit par la tendance des résidus hydrophobes à être exclus de l'eau. Les résidus et les groupes polaires du squelette peuvent interagir entre eux dans le cœur hydrophobe et avec l'eau à l'extérieur. Le repliement des protéines destinées aux environnements non aqueux, par exemple les protéines membranaires, diffère car les résidus non polaires ne doivent plus forcément se trouver dans un cœur hydrophobe.

1.3.2.2 L'organisation de la structure des protéines

Outre les quatre niveaux d'organisation décrits précédemment, d'autres niveaux sont souvent ajoutés à cette hiérarchie. Par exemple, on définit les structures super-secondaires (*supersecondary structures)* ou les domaines.

- Les structures secondaires sont des conformations locales et périodiques présentes dans les protéines. Les hélices alpha et les brins bêta – ces dernier étant associés par paires ou plus de manière parallèle ou anti-parallèle – en sont les plus fréquences. Ces structures locales sont stabilisées par des liaisons hydrogènes. Elles sont aussi nommées structures périodiques car elles sont caractérisées par une répétition de valeur d'angles (ϕ,ψ) ou (α,τ) proches. L'hélice α la plus fréquente est une hélice droite comportant 3,6 résidus par tour. Elle est stabilisée par des liaisons hydrogène, entre le groupe *CO* du résidu *i* et le groupe *NH* du résidu *i* + 4 (cf.

Figure A 1-4 A et B). D'autres hélices droites existent telles les hélices 3-10 et les hélices *pi*. Les hélices gauches sont très peu fréquentes. Dans le feuillet β, les liaisons hydrogènes établies mettent en jeu des régions distantes. Les brins β sont donc généralement organisés en feuillets β parallèles ou antiparallèles (cf. Figure A 1-4 C), mais des coudes « bêta » peuvent aussi se former. Les résidus qui n'appartiennent pas à une structure locale de ces types sont dits organisés en boucle. Tous ces termes (hélices α, brins/feuillets β) sont par la suite regroupés sous le terme de *Secondary Structure Elements* (SSE).

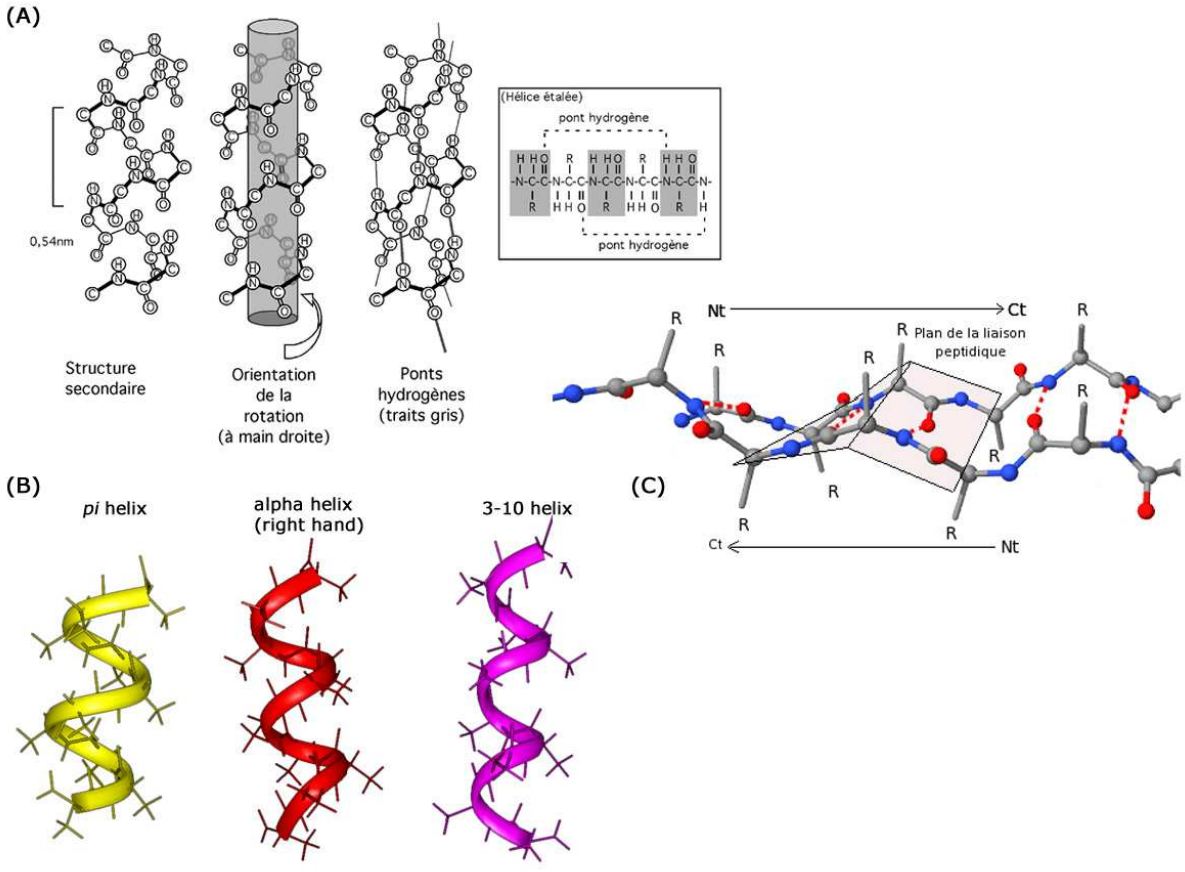


Figure A 1-4 Structures secondaires – Pour les trois figures, les traits pointillés indiquent les liaisons hydrogènes. (A) hélice α (source : http://pages.usherbrooke.ca/bcm-514-bl/2b.html); (B) trois types d'hélice α (source : http://www.imb-jena.de/IMAGE.html); (C) Brins et feuillets β. Les chaînes latérales (R) se répartissent de part et d'autre du feuillet. (source : http://sti-bio.scola.ac-paris.fr/pedago/proteines/html/structure_prot.html)

- Les structures super-secondaires (Rao et Rossmann, 1973 ; Richardson, 1981) sont des organisations de structures secondaires récurrentes dans les protéines. Elles se situent donc entre les niveaux de structure secondaire et tertiaire. Certaines des plus connues sont les

motifs βαβ où deux brins β parallèles sont reliés par une hélice α et les motifs en épingle à cheveux (β *hairpin* et α *hairpin*).

- Entre ce denier niveau et les structures tertiaires se trouvent également les domaines. La définition du terme domaine est délicate et n'est pas consensuelle. Un domaine structural est défini comme étant une unité compacte qui pourrait être stable indépendamment même si cela n'a pas été prouvé expérimentalement (Rose, 1979). Néanmoins, les résidus d'un domaine structural partagent plus de contact entre eux qu'avec d'eux d'un autre domaine. Le terme de domaine fonctionnel est lié à la notion de site fonctionnel (par exemple site de fixation d'une molécule). Les domaines fonctionnels sont plus restreints en terme d'étendue et de nombre de résidus et peuvent être localisés à l'interface des domaines structuraux.

ANNEXE 2

Fichier, Format et Exemples

« Sans exemple, on ne peut renseigner correctement. »

Columelle (1e siècle ap JC)

2.1 Format standard des fichiers de séquences et de structures

A l'image des banques, il existe une multitude de format utilisé pour présenter les données sources. Certains sont bien entendu plus utilisés que d'autres suivant le besoin : ils ne sont pas égaux en termes d'informations. Durant ma thèse, j'ai principalement utilisé trois types de format : deux pour les séquences et la dernière pour les structures.

2.1.1 Les formats FASTA et CLUSTAL pour les séquences

2.1.1.1 Le format FASTA

Le format fasta est un forma basé sur du texte pour représenter soit une séquence nucléique soit une séquence protéique, dans lesquelles chaque base (ou résidu) est représenté par un code à simple lettre. Ce format permet également de renseigner la séquence sur son nom ou toute sorte de commentaire dans un champ spécial précédent la séquence. C'est un format simple qui le rend très facilement manipulable par des scripts de « parsing ». Une séquence au format fasta commence toujours par une ligne unique de description (appelé *header*), suivi par des lignes de données de séquences. La ligne de description se distingue des lignes de données de séquences par le symbole « > » (chevron) à la première colonne. Tout ce qui suit ce symbole sur cette ligne est considéré comme description. On a coutume de placer juste après le chevron l'identifiant de la séquence. En principe, il ne devrait pas y avoir d'espace entre le chevron et l'identifiant. Il n'y a pas de taille limite pour cette première ligne, mais on recommande généralement de mettre un texte inférieur à 80 caractères. La séquence est considérée entière lorsqu'on atteint la fin du fichier ou qu'une autre ligne débute par le chevron, indiquant la présence d'une nouvelle séquence.

```
>gi|117205|sp|P20813|CP2B6_HUMAN Cytochrome P450 2B6 (CYPIIB6) (P450 IIB1)
MELSVLLFLALLTGLLLLLVQRHPNTHDRLPPGPRPLPLLGNLLQMDRRGLLKSFLRFREKYGDVFTVHL
GPRPVVMLCGVEAIREALVDKAEAFSGRGKIAMVDPFFRGYGVIFANGNRWKVLRRFSVTTMRDFGMGKR
SVEERIQEEAQCLIEELRKSKGALMDPTFLFQSITANIICSIVFGKRFHYQDQEFLKMLNLFYQTFSLIS
SVFGQLFELFSGFLKYFPGAHRQVYKNLQEINAYIGHSVEKHRETLDPSAPKDLIDTYLLHMEKEKSNAH
SEFSHQNLNLNTLSLFFAGTETTSTTLRYGFLLMLKYPHVAERVYREIEQVIGPHRPPELHDRAKMPYTE
AVIYEIQRFSDLLPMGVPHIVTQHTSFRGYIIPKDTEVFLILSTALHDPHYFEKPDAFNPDHFLDANGAL
KKTEAFIPFSLGKRICLGEGIARAELFLFFTTILQNFSMASPVAPEDIDLTPQECGVGKIPPTYQIRFLP
R

>gi|85544275|pdb|2BDM|A Chain A, Structure Of Cytochrome P450 2b4 With Bound Bifonazole
MAKKTSSKGKLPPGPSPLPVLGNLLQMDRKGLLRSFLRLREKYGDVFTVYLGSRPVVVLCGTDAIREALV
DQAEAFSGRGKIAVVDPIFQGYGVIFANGERWRALRRFSLATMRDFGMGKRSVEERIQEEARCLVEELRK
SKGALLDNTLLFHSITSNIICSIVFGKRFDYKDPVFLRLLDLFFQSFSLISSFSSQVFELFSGFLKYFPG
THRQIYRNLQEINTFIGQSVEKHRATLDPSNPRDFIDVYLLRMEKDKSDPSSEFHHQNLILTVLSLFFAG
TETTSTTLRYGFLLMLKYPHVTERVQKEIEQVIGSHRPPALDDRAKMPYTDAVIHEIQRLGDLIPFGVPH
TVTKDTQFRGYVIPKNTEVFPVLSSALHDPRYFETPNTFNPGHFLDANGALKRNEGFMPFSLGKRICLGE
GIARTELFLFFTTILQNFSIASPVPPEDIDLTPRESGVGNVPPSYQIRFLARHHHH
```

Figure A 2-1 Exemple de format fasta

Après la ligne de description, un ou plusieurs commentaires peuvent apparaître sous forme de colonne. Comme la plupart des BDD ne reconnaisse pas ces commentaires, leur utilisation s'est vu disparaître, alors qu'elles font partie de la nomenclature officielle de ce format. Concernant l'extension de fichier, les fichiers au format fasta arborent généralement les extensions .fasta, .fsa, .fna, .fa ou encore .mfa. Une dernière remarque concerne la partie « identifiant » de la séquence au niveau de la première ligne. Cette partie est maintenant codifiée comme il apparaît en Figure A 2-2.

GenBank	gi|*gi-number*|gb|*accession*|*locus*
EMBL Data Library	gi|*gi-number*|emb|*accession*|*locus*
DDBJ, DNA Database of Japan	gi|*gi-number*|dbj|*accession*|*locus*
NBRF PIR	pir||*entry*
Protein Research Foundation	prf||*name*
SWISS-PROT	sp|*accession*|*name*
Brookhaven Protein Data Bank (1)	pdb|*entry*|*chain*
Brookhaven Protein Data Bank (2)	*entry*:*chain*|PDBID|CHAIN|SEQUENCE
Patents	pat|country|number
GenInfo Backbone Id	bbs|number
General database identifier	gnl|database|identifier
NCBI Reference Sequence	ref|accession|locus
Local Sequence identifier	lcl|identifier

Figure A 2-2 Nomenclature de la partie « identifiant »

2.1.1.2 Le format d'alignement Clustal

Il s'agit d'un format standard d'alignement de séquences nucléotidiques ou protéiques. C'est le format de sortie de l'outil d'alignement multiple Clustalw. Le mot CLUSTAL figure sur la première ligne du fichier. L'alignement est divisé en bloc de longueur fixe où chaque ligne de ce bloc correspond à une séquence. Chaque ligne de chaque bloc commence avec le nom de la séquence (sur un nombre de caractères limité, maximum 10), suivi d'au moins un espace. La séquence est ensuite représentée: en majuscules ou minuscules, les gaps sont indiqués par des tirets hauts "–". Le nombre de résidus est parfois ajouté à la fin de la première ligne de chaque bloc. Un exemple de format Clustal est présenté en Figure A 2-3.

```
CLUSTAL W (1.8) multiple sequence alignment
                1                                                     60
TRGJ1_01        ------------GAATTATTATAAGAAACTCTTTGGCAGTGGAACAACACTGGTTGTCAC
TRGJ2_01        ------------GAATTATTATAAGAAACTCTTTGGCAGTGGAACAACTCTTGTTGTCAC
TRGJP_01        TGGGCAAGAGTTGGGCAAAAAAATCAAGGTATTTGGTCCCGGAACAAAGCTTATCATTAC
TRGJP1_01       --------ATACCACTGGTTGGTTCAAGATATTTGCTGAAGGGACTAAGCTCATAGTAAC
TRGJP2_01       --------ATAGTAGTGATTGGATCAAGACGTTTGCAAAAGGGACTAGGCTCATAGTAAC
                                       **    ****     ** ** *  **  *  * **

                61    68
TRGJ1_01        AG------
TRGJ2_01        AG------
TRGJP_01        AG------
TRGJP1_01       TTCACCTG
TRGJP2_01       TTCGCCTG
```

Figure A 2-3 Exemple de format Clustal

2.1.2 Le format PDB pour les structures

Le format le plus couramment observé pour les structures est le format pdb, format original de la banque. Le guide de ce format a été révisé à plusieurs reprises ; la version actuelle est la version 2.2, qui existe depuis 1996. Les fichiers au format pdb contiennent différentes informations telles que les coordonnées cartésiennes des atomes, la bibliographie, les informations structurales, les facteurs de la structure cristallographique et les données expérimentales de la RMN. A l'origine, le format pdb a été dicté par l'utilisation et la largeur de cartes perforées pour ordinateur. En conséquence, chaque ligne contient exactement 80 caractères. Un fichier au format pdb est un fichier texte où chaque colonne possède sa signification : chaque paramètre est positionné de façon immuable. Ainsi, les 6 premières colonnes, c'est-à-dire les 6 premiers caractères pour une ligne donnée, déterminent le champ du fichier. On retrouve par exemple les champs « TITLE_ » (c'est-à-dire le titre de la macromolécule étudiée), « KEYWDS » (les mots-clé de l'entrée), « EXPDTA » qui donne des informations sur la méthode expérimentale employée, « SEQRES » (la séquence de la protéine étudiée), « ATOM__ » ou « HETATM », champs comprenant toutes les informations liées à un atome particulier. Dernier exemple, dans ces derniers champs, le nom de l'atome est décrit par les colonnes 13 à 16 (soit du treizième au seizième caractère de la ligne). Les lignes « ATOM__ » concernent les acides aminés ou les acides nucléiques, et les lignes « HETATM » sont dédiées aux autres molécules (solvant, substrat, ion, détergent…). Il y a autant de lignes « ATOM__ » et « HETATM » que d'atomes observés par l'expérimentateur, pour une macromolécule ou un complexe donné.

La longue histoire du format pdb a abouti sur des données non uniformes. Ce format laisse également la place à de nombreuses erreurs, qui ne sont pas systématiquement éliminées lors des contrôles accompagnant le dépôt des structures. Il peut s'agir de désaccords entre la séquence et les résidus représentés, ou de problèmes liés à la nomenclature des atomes des acides aminés ou des ligands. Par ailleurs, ce format présente des limites liées à sa conception même : le fait même qu'il ne puisse contenir que 80 colonnes fait de lui un format relativement restrictif. Le nombre maximum d'atomes d'un fichier pdb est de 99999, vu qu'il n'y a que 5 colonnes allouées pour les numéros des atomes. De même le nombre de résidus par chaîne est au maximum de 9999 : il n'y a que 4 colonnes autorisées pour ce chiffre. Le nombre de chaînes, lui, est limité à 62 : une seule colonne est disponible, et les valeurs possibles sont une des lettres des 26 lettres de l'alphabet, en minuscule ou en majuscule, ou un des chiffres de 0 à 9. Quant ce format a été défini, ces limitations ne semblaient pas restrictives, mais elles ont plusieurs fois été franchies lors du dépôt de structures extrêmement grandes, comme des virus, des ribosomes ou des complexes multienzymatiques.

```
HEADER    OXIDOREDUCTASE                          17-JUN-04   1TQN
TITLE     CRYSTAL STRUCTURE OF HUMAN MICROSOMAL P450 3A4
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: CYTOCHROME P450 3A4;
COMPND   3 CHAIN: A;
COMPND   4 EC: 1.14.14.1;
COMPND   5 ENGINEERED: YES
SOURCE    MOL_ID: 1;
SOURCE   2 ORGANISM_SCIENTIFIC: HOMO SAPIENS;
SOURCE   3 ORGANISM_COMMON: HUMAN;
SOURCE   4 GENE: CYP3A4;
SOURCE   5 EXPRESSION_SYSTEM: ESCHERICHIA COLI;
SOURCE   6 EXPRESSION_SYSTEM_COMMON: BACTERIA;
SOURCE   7 EXPRESSION_SYSTEM_STRAIN: DH5 ALPHA;
SOURCE   8 EXPRESSION_SYSTEM_VECTOR_TYPE: PLASMID;
SOURCE   9 EXPRESSION_SYSTEM_PLASMID: PSE3A4DHHIS
KEYWDS    P450, CYP3A4, MONOOXYGENASE, DRUG METABOLIZING ENZYME,
KEYWDS   2 OXIDOREDUCTASE, HEME
EXPDTA    X-RAY DIFFRACTION
AUTHOR    J.K.YANO,M.R.WESTER,G.A.SCHOCH,K.J.GRIFFIN,C.D.STOUT,
AUTHOR   2 E.F.JOHNSON
REVDAT   2   14-SEP-04 1TQN    1       JRNL
REVDAT   1   27-JUL-04 1TQN    0
JRNL        AUTH   J.K.YANO,M.R.WESTER,G.A.SCHOCH,K.J.GRIFFIN,
JRNL        AUTH 2 C.D.STOUT,E.F.JOHNSON
JRNL        TITL   THE STRUCTURE OF HUMAN MICROSOMAL CYTOCHROME P450
JRNL        TITL 2 3A4 DETERMINED BY X-RAY CRYSTALLOGRAPHY TO 2.05-A
JRNL        TITL 3 RESOLUTION
JRNL        REF    J.BIOL.CHEM.                 V. 279 38091 2004
JRNL        REFN   ASTM JBCHA3  US ISSN 0021-9258
REMARK   1
[...]
SEQRES   1 A  486  MET ALA LEU TYR GLY THR HIS SER HIS GLY LEU PHE LYS
SEQRES   2 A  486  LYS LEU GLY ILE PRO GLY PRO THR PRO LEU PRO PHE LEU
SEQRES   3 A  486  GLY ASN ILE LEU SER TYR HIS LYS GLY PHE CYS MET PHE
SEQRES   4 A  486  ASP MET GLU CYS HIS LYS LYS TYR GLY LYS VAL TRP GLY
SEQRES   5 A  486  PHE TYR ASP GLY GLN GLN PRO VAL LEU ALA ILE THR ASP
SEQRES   6 A  486  PRO ASP MET ILE LYS THR VAL LEU VAL LYS GLU CYS TYR
SEQRES   7 A  486  SER VAL PHE THR ASN ARG ARG PRO PHE GLY PRO VAL GLY
SEQRES   8 A  486  PHE MET LYS SER ALA ILE SER ILE ALA GLU ASP GLU GLU
SEQRES   9 A  486  TRP LYS ARG LEU ARG SER LEU LEU SER PRO THR PHE THR
[...]
ATOM      1  N   HIS A  28     -30.070   8.178 -13.891  1.00 62.96           N
ATOM      2  CA  HIS A  28     -29.618   8.226 -15.315  1.00 62.61           C
ATOM      3  C   HIS A  28     -28.098   8.416 -15.437  1.00 60.27           C
ATOM      4  O   HIS A  28     -27.574   9.544 -15.459  1.00 60.60           O
ATOM      5  CB  HIS A  28     -30.349   9.350 -16.065  1.00 65.70           C
ATOM      6  CG  HIS A  28     -31.223   8.866 -17.185  1.00 68.78           C
ATOM      7  ND1 HIS A  28     -32.203   7.912 -17.008  1.00 70.34           N
ATOM      8  CD2 HIS A  28     -31.276   9.222 -18.492  1.00 70.01           C
ATOM      9  CE1 HIS A  28     -32.823   7.701 -18.158  1.00 69.94           C
ATOM     10  NE2 HIS A  28     -32.281   8.484 -19.074  1.00 70.77           N
ATOM     11  N   SER A  29     -27.405   7.284 -15.471  1.00 56.14           N
ATOM     12  CA  SER A  29     -25.961   7.227 -15.632  1.00 50.52           C
[...]
HETATM 3801  C1D HEM A 500     -15.561 -22.334  -8.362  1.00 36.10           C
HETATM 3802  C2D HEM A 500     -15.431 -21.280  -7.478  1.00 36.32           C
HETATM 3803  C3D HEM A 500     -15.279 -20.141  -8.213  1.00 36.60           C
HETATM 3804  C4D HEM A 500     -15.329 -20.563  -9.563  1.00 35.03           C
HETATM 3805  CMD HEM A 500     -15.447 -21.345  -5.937  1.00 35.22           C
HETATM 3806  CAD HEM A 500     -15.092 -18.715  -7.649  1.00 33.19           C
HETATM 3807  CBD HEM A 500     -13.578 -18.419  -7.435  1.00 36.93           C
HETATM 3808  CGD HEM A 500     -13.146 -17.057  -6.844  1.00 38.80           C
HETATM 3809  O1D HEM A 500     -12.106 -16.955  -6.219  1.00 40.03           O
HETATM 3810  O2D HEM A 500     -13.867 -16.061  -7.005  1.00 39.63           O
HETATM 3811  O   HOH     1     -17.896 -22.802 -11.166  1.00 39.05           O
HETATM 3812  O   HOH     2     -30.465 -23.810 -12.844  1.00 47.23           O
HETATM 3813  O   HOH     3     -19.375 -19.692  -9.011  1.00 54.56           O
HETATM 3814  O   HOH     4     -19.657 -17.459 -10.200  1.00 59.14           O
HETATM 3815  O   HOH     5     -20.277 -10.689 -11.676  1.00 37.63           O
HETATM 3816  O   HOH     6     -25.143 -21.181 -18.039  1.00 28.45           O
HETATM 3817  O   HOH     7     -25.255 -18.716 -10.661  1.00 37.59           O
HETATM 3818  O   HOH     8     -16.940 -10.863  -5.739  1.00 45.76           O
[...]
CONECT 3806 3803 3807
CONECT 3807 3806 3808
CONECT 3808 3807 3809 3810
CONECT 3809 3808
CONECT 3810 3808
MASTER      320    0    1   24    9    0    0    6 3999    1   45   38
END
```

Figure A 2-4 Exemple de fichier PDB (Source 1TQN). Le fichier a été sectionné en raison de son volume.

À noter que certains logiciels qui ont été utilisés dans ce travail de thèse, comme SYBYL, admettent également un autre type de format pour les structures : le format MOL2. Il s'agit d'un format de coordonnées beaucoup plus complète et précise que celle de PDB. Toutefois, comme les structures de la PDB sont données au format PDB, les formats MOL2 n'ont pas été utilisés.

2.2 Format spécifique à certains logiciels

Les formats de fichier présentés précédemment sont les plus utilisés en bioinformatique. La plupart des logiciels sont en mesure de lire et d'interpréter les séquences ou les structures écrites sous ces formats. Toutefois, certains programmes nécessitent des informations supplémentaires qui doivent être renseignées dans un même fichier que les séquences ou les structures. Pour cela, de nouveaux formats spécifiques à certains programmes sont dérivés des formats standards. C'est le cas par exemple pour le logicielle Modeller qui utilise un format particulier de séquence, dérivé du format FASTA.

2.2.1 Format d'alignement de séquence de Modeller

Le fichier d'alignement pour Modeller s'apparente beaucoup à un fichier de séquences au format FASTA. En entête, le chevron est suivit du symbole « P1 ; » puis du code PDB de la structure. En seconde ligne, juste après l'entête, un champ spécial est utilisé par Modeller pour déterminer la « nature » de la séquence qui suit : s'agit-il d'une séquence de *template* ou de la séquence cible à reconstruire. Dans le premier cas, le mot clef « *structureX* » doit apparaitre pour désigner les structures de références, suivi de la correspondance entre les résidus de la séquence et ceux de la structure : il faut donc indiquer le numéro de résidu de début et de fin (ainsi que la chaîne d'appartenance) correspondant dans le fichier PDB. Chacune de ces informations sont séparés par un délimiteur de champs : « : ». Modeller est très pointilleux sur cette délimitation des résidus : la séquence de la structure utilisée dans le fichier d'alignement doit comporter le même nombre de résidus (et surtout les mêmes) que ceux présent dans le fichier PDB correspondant. Dans le cas de la séquence à reconstruire, la deuxième ligne de renseignement est simplifiée : elle débute par le délimiteur « *sequence* » puis suivi de champ vide entre les délimiteurs « : ». Une fois les deux premières lignes renseignées, vient ensuite la séquence. La séquence comportant des gaps (« – ») est placé ensuite sous la deuxième ligne. Il n'y a pas de règle particulière pour la longueur de chaque ligne pour la séquence, mais chaque séquence doit être marqué par un délimiteur de fin : l'astérisque « * ». À ce délimiteur de fin, on peut ajouter un autre délimiteur « /h » lorsqu'on désire utiliser les hétéroatomes d'un fichier PDB. Il est à noter que dans ce cas, il faut penser à modifier le numéro correspondant au résidu final de la seconde

ligne. Par exemple 1suo comporte les résidus 28 (une glycine) à 492 (une histidine). Le fait d'avoir ajouté « /h » informe à Modeller de prendre en compte les hétéroatomes présents dans le fichier 1suo.pdb. Comme je n'ai laissé qu'un seul hétéroatome, l'hème, il a fallu inscrire 493 au lieu de 492 (cf. Figure A 2-5). Pour la séquence, il y a moins d'information à donner, sauf le fait qu'il reçoit des hétéroatomes, c'est pourquoi la balise « /h » est placé en fin de séquence.

Figure A 2-5 Exemple de fichier d'alignement pour Modeller. Trois *templates* sont utilisés ici pour construire le modèle du CYP 1A2. Ces trois *templates* sont le 1suo (CYP 2B4), 2p85 (CYP 2A13) et 2f9q (CYP 2D6). Les *templates* sont identifiés par Modeller par l'identifiant présent sur la seconde ligne (en rouge) : « structureX » suivit de plusieurs champs séparés par des « : ». Parmis ces champs, on retrouve leur code PDB les numéros de résidus de début et de fin de séquence dans le fichier PDB correspondant, la chaîne etc… Un astérisque (*) est placée en fin de chaque séquence pour informer au logiciel de la fin de chaque séquence. Des marqueurs supplémentaires tels que « /h » peuvent être ajoutés lorsqu'on désire prendre en compte les hétéroatomes présents dans le fichier PDB. La séquence cible est reconnue avec un même type de délimiteur (en vert) avec « sequence » à la place.

2.2.2 Format des fichiers de CSBs

Dans le programme *Caliseq*, un ensemble de blocs doit être fourni au programme. Ces derniers contiennent des sous-alignements des séquences des structures utilisées en référence. Par souci de simplicité, j'ai opté le format FASTA pour représenter chaque sous-alignement correspondant chacun à un bloc. Afin de séparer chaque bloc et conserver ainsi leur séquentialité, chaque sous-alignement sont séparé par le symbole « # » qui sert de **délimiteur**. Par ailleurs, pour des raisons qui seront expliquées dans la partie détaillant le programme, il faut penser à avoir le même nombre et le même ordre des séquences dans chaque sous-alignement. Le contenu des trois fichiers des ensembles de blocs utilisés sont présentés dans les figures suivantes.

```
>1OXA
------------
>2HPD
PKTFGELKNLPL
>1PQ2
PTPLPIIGNMLQ
>1OG5
PTPLPVIGNILQ
>1NR6
PTPFPIIGNILQ
>1GWI
------------
#
>1OXA
VDWYSTYAELRE
>2HPD
DKPVQALMKIAD
>1PQ2
KDICKSFTNFSK
>1OG5
KDISKSLTNLSK
>1NR6
KDISKSLTKFSE
>1GWI
TDLDGESARLRA
#
>1OXA
PVTPVRF
>2HPD
EIFKFEA
>1PQ2
PVFTVYF
>1OG5
PVFTLYF
>1NR6
PVFTVYL
>1GWI
PLAAVEL
#
>1OXA
LGQDAWLVTGYDEAKAAL
>2HPD
PGRVTRYLSSQRLIKEAC
>1PQ2
GMNPIVVFHGYEAVKEAL
>1OG5
GLKPIVVLHGYEAVKEAL
>1NR6
GMKPTVVLHGYEAVKEAL
>1GWI
GGVPVWAVTHHAEAKALL
#
>1OXA
FPEDVRNY
>2HPD
LSQALKFV
>1PQ2
NSPISQRI
>1OG5
IFPLAERA
>1NR6
SVPILEKV
>1GWI
DWPLIGLA
#
```

```
>1OXA
NMGTS
>2HPD
GLFTS
>1PQ2
GIISS
>1OG5
GIVFS
>1NR6
GIAFS
>1GWI
SMLTV
#
>1OXA
PTHTRLRKLVSQEFTVRR
>2HPD
KNWKKAHNILLPSFSQQA
>1PQ2
KRWKEIRRFSLTTLRNFG
>1OG5
KKWKEIRRFSLMTLRNFG
>1NR6
KTWKEMRRFSLMTLRNFG
>1GWI
AEHRRLRTLVAQALTVRR
#
>1OXA
VEAMRPRVEQITAELLDEV
>2HPD
YHAMMVDIAVQLVQKWERL
>1PQ2
IEDRVQEEAHCLVEELRKT
>1OG5
IEDRVQEEARCLVEELRKT
>1NR6
IEDRIQEEARCLVEELRKT
>1GWI
VEHMRGRITELTDRLLDEL
#
>1OXA
VVDIVDRFAHPLPIKVICELL
>2HPD
HIEVPEDMTRLTLDTIGLCGF
>1PQ2
PCDPTFILGCAPCNVICSVVF
>1OG5
PCDPTFILGCAPCNVICSIIF
>1NR6
PCDPTFILGCAPCNVICSVIF
>1GWI
VVDLKAAFAYPLPMYVVADLM
#
>1OXA
DEAARGAFGRWSSEIL
>2HPD
HPFITSMVRALDEAMN
>1PQ2
DQNFLTLMKRFNENFR
>1OG5
DQQFLNLMEKLNENIE
>1NR6
DEEFLKLMESLHENVE
>1GWI
EEARLPRLKVLFEKFF
#
```

```
>1OXA
AEQRGQAAREVVNFILDLVERRRT
>2HPD
KRQFQEDIKVMNDLVDKIIADRKA
>1PQ2
HNKVLKNVALTRSYIREKVKEHQA
>1OG5
HNKLLKNVAFMKSYILEKVKEHQE
>1NR6
HKTLLKNADYIKNFIMEKVKEHQK
>1GWI
PEEVVATLTELASIMTDTVAAKRA
#
>1OXA
DLLSALI
>2HPD
DLLTHML
>1PQ2
DFIDCFL
>1OG5
DFIDCFL
>1NR6
DFIDCFL
>1GWI
DLTSALI
#
>1OXA
SADELTSIALVLLLAGFEASVSLIGIGTYLLLT
>2HPD
DDENIRYQIITFLIAGHETTSGLLSFALYFLVK
>1PQ2
NIENLVGTVADLFVAGTETTSTTLRYGLLLLLK
>1OG5
TIESLENTAVDLFGAGTETTSTTLRYALLLLLK
>1NR6
TLESLVIAVSDLFGAGTETTSTTLRYSLLLLLK
>1GWI
TDAEIVSTLQLMVAAGHETTISLIVNAVVNLST
#
>1OXA
HPDQLALVR
>2HPD
NPHVLQKAA
>1PQ2
HPEVTAKVQ
>1OG5
HPEVTAKVQ
>1NR6
HPEVAARVQ
>1GWI
HPEQRALVL
#
>1OXA
------
>2HPD
SYKQVK
>1PQ2
CMQDRS
>1OG5
CMQDRS
>1NR6
CMQDRS
>1GWI
------
#
```

```
>1OXA
LPNAVEEILRYIAPPET
>2HPD
VGMVLNEALRLWPTAPA
>1PQ2
TDAVVHEIQRYSDLVPT
>1OG5
TDAVVHEVQRYIDLLPT
>1NR6
TDAVIHEIQRFIDLLPT
>1GWI
WSAVVEETLRFSTPTSH
#
>1OXA
TRFAAEEVEIGG
>2HPD
SLYAKEDTVLGG
>1PQ2
PHAVTTDTKFRN
>1OG5
PHAVTCDIKFRN
>1NR6
PHAVTRDVRFRN
>1GWI
IRFAAEDVPVGD
#
>1OXA
VAIPQYSTVLVANGAANRDPSQFP
>2HPD
YPLEKGDELMVLIPQLHRDKTIWG
>1PQ2
YLIPKGTTIMALLTSVLHDDKEFP
>1OG5
YLIPKGTTILISLTSVLHDNKEFP
>1NR6
YFIPKGTDIITSLTSVLHDEKAFP
>1GWI
RVIPAGDALIVSYGALGRDERAHG
#
>1OXA
DPHRFDVT
>2HPD
DVEEFRPE
>1PQ2
NPNIFDPG
>1OG5
NPEMFDPH
>1NR6
NPKVFDPG
>1GWI
TADRFDLT
#
>1OXA
RGHLSFGQ
>2HPD
HAFKPFGN
>1PQ2
DYFMPFSA
>1OG5
KYFMPFSA
>1NR6
DYFMPFSA
>1GWI
--------
#
```

```
>1OXA
GIHFCMGRPLAKLEGEVALRALFGRF
>2HPD
GQRACIGQQFALHEATLVLGMMLKHF
>1PQ2
GKRICAGEGLARMELFLFLTTILQNF
>1OG5
GKRICVGEALAGMELFLFLTSILQNF
>1NR6
GKRMCVGEGLARMELFLFLTSILQNF
>1GWI
GPHVCPGAALSRMEAGVALPALYARF
#
>1OXA
ALSLGID
>2HPD
DFEDHTN
>1PQ2
NLKSVDD
>1OG5
NLKSLVD
>1NR6
KLQSLVE
>1GWI
HLDLAVP
#
>1OXA
ADDVVWRRS
>2HPD
---------
>1PQ2
KNLNTTAVT
>1OG5
KNLDTTPVV
>1NR6
KDLDITAVV
>1GWI
AAELRNKPV
#
>1OXA
LLLRGIDHLPVR
>2HPD
TLTLKPEGFVVK
>1PQ2
GIVSLPPSYQIC
>1OG5
GFASVPPFYQLC
>1NR6
GFVSVPPSYQLC
>1GWI
VTQNDLFELPVR
```

Figure A 2-6 Fichier de l'ensemble de blocs de TA. NGUYEN. Il s'agit d'un unique fichier contentant 24 sous-alignements des six structures de référence : 1oxa, 2hpd, 1pq2, 1og5, 1nr6 et 1gwi. Trois sont bactériennes et trois proviennent des mammifères.

```
>1rom
AEFAKLRA
>1oxa
WYSTYAEL
>1cpt
IYPAFKWL
>3ccp
VQEAWAVL
>1e9x
PIGLMQRV
>2hpd
PVQALMKI
>1dt6
ISKSLTKF
#
>1rom
PVSQVKLF
>1oxa
PVTPVRFL
>1cpt
PLAMAHIE
>3ccp
DLVWTRCN
>1e9x
DVGTFQLA
>2hpd
EIFKFEAP
>1dt6
PVFTVYLG
#
>1rom
LAWLVTKHKDVCFV
>1oxa
DAWLVTGYDEAKAA
>1cpt
PMWIATKHADVMQI
>3ccp
GHWIATRGQLIREA
>1e9x
QVVLLSGSHANEFF
>2hpd
VTRYLSSQRLIKEA
>1dt6
PTVVLHGYEAVKEA
#
>1rom
SASGKQ
>1oxa
PEDVRN
>1cpt
DQNNEA
>3ccp
PREAGE
>1e9x
AKAYPF
>2hpd
SQALKF
>1dt6
SVPILE
#
>1rom
TFVDMDP
>1oxa
NMGTSDP
>1cpt
SLTSMDP
>3ccp
IPTSMDP
>1e9x
GVVFDAS
>2hpd
GLFTSWT
>1dt6
GIAFSNA
#
```

```
>1rom
PEHMHQRSMVEPTF
>1oxa
PTHTRLRKLVSQEF
>1cpt
PTHTAYRGLTLNWF
>3ccp
PEQRQFRALANQVV
>1e9x
PERRKEMLHNAALR
>2hpd
KNWKKAHNILLPSF
>1dt6
KTWKEMRRFSLMTL
#
>1rom
LQPYIQRTVDDLLEQMKQK
>1oxa
VEAMRPRVEQITAELLDEV
>1cpt
LEENIRRIAQASVQRLLDF
>3ccp
LENRIQELACSLIESLRPQ
>1e9x
MKGHAATIEDQVRRMIADW
>2hpd
YHAMMVDIAVQLVQKWERL
>1dt6
IEDRIQEEARCLVEELRKT
#
>1rom
VDLVKEFALPVPSYIIYTLLGVP
>1oxa
VDIVDRFAHPLPIKVICELLGVD
>1cpt
CDFMTDCALYYPLHVVMTALGVP
>3ccp
CNFTEDYAEPFPIRIFMLLAGLP
>1e9x
IDLLDFFEALTIYTSSACLIGKK
>2hpd
IEVPEDMTRLTLDTIGLCGFNYR
>1dt6
CDPTFILGCAPCNVICSVIFHNR
#
>1rom
FNDLEYLTQQNAIRT
>1oxa
EAARGAFGRWSSEIL
>1cpt
EDDEPLMLKLTQDFF
>3ccp
EEDIPHLKYLTDQMT
>1e9x
GRFAKLYHELERGTD
>2hpd
PFITSMVRALDEAMN
>1dt6
EEFLKLMESLHENVE
#
>1rom
ASAANQELLDYLAILVEQRLV
>1oxa
RGQAAREVVNFILDLVERRRT
>1cpt
FHETIATFYDYFNGFTVDRRS
>3ccp
FAEAKEALYDYLIPIIEQRRQ
>1e9x
RDEARNGLVALVADIMNGRIA
>2hpd
FQEDIKVMNDLVDKIIADRKA
>1dt6
LLKNADYIKNFIMEKVKEHQK
#
```

```
>1rom
DIISKLCT
>1oxa
DLLSALIS
>1cpt
DVMSLLAN
>3ccp
DAISIVAN
>1e9x
DMLDVLIA
>2hpd
DLLTHMLN
>1dt6
DFIDCFLI
#
>1rom
DKSDAVQIAFLLLVAGNATMVNMIALGVATLAQ
>1oxa
SADELTSIALVLLLAGFEASVSLIGIGTYLLLT
>1cpt
DDKYINAYYVAIATAGHDTTSSSSGGAIIGLSR
>3ccp
TSDEAKRMCGLLLVGGLDTVVNFLSFSMEFLAK
>1e9x
SADEITGMFISMMFAGHHTSSGTASWTLIELMR
>2hpd
DDENIRYQIITFLIAGHETTSGLLSFALYFLVK
>1dt6
TLESLVIAVSDLFGAGTETTSTTLRYSLLLLLK
#
>1rom
HPDQLAQLKA
>1oxa
HPDQLALVRA
>1cpt
NPEQLALAKS
>3ccp
SPEHRQELIE
>1e9x
HRDAYAAVID
>2hpd
NPHVLQKAAE
>1dt6
HPEVAARVQE
#
>1rom
SLAPQFVEELCRYHTAS
>1oxa
SALPNAVEEILRYIAPP
>1cpt
ALIPRLVDEAVRWTAPV
>3ccp
ERIPAACEELLRRFSLV
>1e9x
PQLENVLKETLRLHPPL
>2hpd
KYVGMVLNEALRLWPTA
>1dt6
PYTDAVIHEIQRFIDLL
#
>1rom
KRTAKEDVMIGD
>1oxa
TRFAAEEVEIGG
>1cpt
MRTALADTEVRG
>3ccp
GRILTSDYEFHG
>1e9x
MRVAKGEFEVQG
>2hpd
SLYAKEDTVLGG
>1dt6
PHAVTRDVRFRN
#
```

```
>1rom
KLVRANEGIIASNQSANRDEEVF
>1oxa
VAIPQYSTVLVANGAANRDPSQF
>1cpt
QNIKRGDRIMLSYPSANRDEEVF
>3ccp
VQLKKGDQILLPQMLSGLDEREN
>1e9x
HRIHEGDLVAASPAISNRIPEDF
>2hpd
YPLEKGDELMVLIPQLHRDKTIW
>1dt6
YFIPKGTDIITSLTSVLHDEKAL
#
>1rom
ENPDEFNM
>1oxa
PDPHRFDV
>1cpt
SNPDEFDI
>3ccp
ACPMHVDF
>1e9x
PDPHDFVP
>2hpd
DDVEEFRP
>1dt6
PNPKVFDP
#
>1rom
PLGFGFGDHRC
>1oxa
HLSFGQGIHFC
>1cpt
HLGFGWGAHMC
>3ccp
HTTFGHGSHLC
>1e9x
WIPFGAGRHRC
>2hpd
FKPFGNGQRAC
>1dt6
FMPFSAGKRMC
#
>1rom
IAEHLAKAELTTVFSTLYQKF
>1oxa
MGRPLAKLEGEVALRALFGRF
>1cpt
LGQHLAKLEMKIFFEELLPKL
>3ccp
LGQHLARREIIVTLKEWLTRI
>1e9x
VGAAFAIMQIKAIFSVLLREY
>2hpd
IGQQFALHEATLVLGMMLKHF
>1dt6
VGEGLARMELFLFLTSIlQNF
#
>1rom
PDLKV
>1oxa
ALSLG
>1cpt
SVELS
>3ccp
FSIAP
>1e9x
EFEMA
>2hpd
DFEDH
>1dt6
KLQSL
#
```

```
>1rom
KINYTPLNR
>1oxa
DVVWRRSLL
>1cpt
PPRLVATNF
>3ccp
AQIQHKSGI
>1e9x
SYRNDHSKM
>2hpd
YELDIKETL
>1dt6
LDITAVVNG
#
>1rom
VGIVDLPV
>1oxa
RGIDHLPV
>1cpt
GGPKNVPI
>3ccp
SGVQALPL
>1e9x
VQLAQPAC
>2hpd
LKPEGFVV
>1dt6
VSVPPSYQ
```

Figure A 2-7 Fichier de l'ensemble de blocs de N. Loiseau. Celui-ci contient 22 sous-alignements de six structures de référence : 1rom, 1oxa, 1cpt, 3cpp, 1e9x, 2hpd, 1dt6. Dans ce jeu, une seule structure provient des mammifères : il s'agit du 1dt6, le CYP 2C5 de lapin.

```
>1oxa
VDWYSTYAELRETAP
>1e9x
TDPIGLMQRVRDECG
>2hpd
DKPVQALMKIADELG
>1dt6
KDISKSLTKFSECYG
#
>1oxa
VTPVRFL
>1e9x
VGTFQLA
>2hpd
IFKFEAP
>1dt6
VFTVYLG
#
>1oxa
GQDAWLVTGYDEAKAA
>1e9x
GKQVVLLSGSHANEFF
>2hpd
GRVTRYLSSQRLIKEA
>1dt6
MKPTVVLHGYEAVKEA
#
>1oxa
EDVRN
>1e9x
PFMTP
>2hpd
QALKF
>1dt6
VPILE
#
>1oxa
TSD
>1e9x
DAS
>2hpd
TSW
>1dt6
FSN
#
>1oxa
THTRLRKLVSQEFTVRR
>1e9x
PERRKEMLHNAALRGEQ
>2hpd
NWKKAHNILLPSFSQQA
>1dt6
TWKEMRRFSLMTLRNFG
#
>1oxa
VEAMRPRVEQITAELLDEV
>1e9x
MKGHAATIEDQVRRMIADW
>2hpd
YHAMMVDIAVQLVQKWERL
>1dt6
IEDRIQEEARCLVEELRKT
#
>1oxa
GDS
>1e9x
GEA
>2hpd
NAD
>1dt6
NAS
#
>1oxa
VVDIVDRFAHPLPIKVICELL
>1e9x
EIDLLDFFAELTIYTSSACLI
>2hpd
HIEVPEDMTRLTLDTIGLCGF
>1dt6
PCDPTFILGCAPCNVICSVIF
#
>1oxa
AFGRWSSEIL
>1e9x
RFAKLYHELE
>2hpd
PFITSMVRAL
>1dt6
KLMESMVRAL
#
>1oxa
AEQRGQAAREVVNFILDLVERRRT
>1e9x
FRRRDEARNGLVALVADIMNGRIA
>2hpd
KRQFQEDIKVMNDLVDKIIADRKA
>1dt6
HKTLLKNADYIKNFIMEKVKEHQK
#
>1oxa
DLLSALIS
>1e9x
DMLDVLIA
>2hpd
DLLTHMLN
>1dt6
DFIDCFLI
#
>1oxa
QDDDD
>1e9x
VKAET
>2hpd
KDPET
>1dt6
KMEQE
#
>1oxa
LSADELTSIALVLLLAGFEASVSLIGIGTYLLLT
>1e9x
FSADEITGMFISMMFAGHHTSSGTASWTLIELMR
>2hpd
LDDENIRYQIITFLIAGHETTSGLLSFALYFLVK
>1dt6
FTLESLVIAVSDLFGAGTETTSTTLRYSLLLLK
#
>1oxa
HPDQLALVRA
>1e9x
HRDAYAAVID
>2hpd
NPHVLQKAAE
>1dt6
HPEVAARVQE
#
>1oxa
------
>1e9x
ELDELY
>2hpd
EAARVL
>1dt6
EIERVI
#
>1oxa
----------
>1e9x
RSVSFHALRQI
>2hpd
PVPSYKQVKQL
>1dt6
RSPCMQDRSRM
#
>1oxa
SALPNAVEEILRYIAPPET
>1e9x
PQLENVLKETLRLHPPLII
>2hpd
KYVGMVLNEALRLWPTAPA
>1dt6
PYTDAVIHEIQRFIDLLPT
#
>1oxa
TRFAAEEVEIGG
>1e9x
MRVAKGEFEVQG
>2hpd
SLYAKEDTVLGG
>1dt6
PHAVTRDVRFRN
#
>1oxa
VAI
>1e9x
HRI
>2hpd
YPL
>1dt6
YFI
#
>1oxa
PQYSTVLVANGAANRDPSQF
>1e9x
HEGDLVAASPAISNRIPEDF
>2hpd
EKGDELMVLIPQLHRDKTIW
>1dt6
PKGTDIITSLTSVLHDEKAF
#
>1oxa
PDPHRFDVT
>1e9x
PDPHDFVPA
>2hpd
DDVEEFRPE
>1dt6
PNPKVFDPG
#
>1oxa
DTRGHLSFG
>1e9x
NRWTWIPFG
>2hpd
PQHAFKPFG
>1dt6
KSDYFMPFS
#
>1oxa
GIHFCMGRPLAKLEGEVALRALFGRF
>1e9x
GRHRCVGAAFAIMQIKAIFSVLLREY
>2hpd
GQRACIGQQFALHEATLVLGMMLKHF
>1dt6
GKRMCVGEGLARMELFLFLTSILQNF
#
>1oxa
ALSLGID
>1e9x
EFEMAQP
>2hpd
DFEDHTN
>1dt6
KLQSLVE
#
>1oxa
DVVWRRS
>1e9x
SYRNDHS
>2hpd
YELDIKE
>1dt6
LDITAVV
#
>1oxa
LLRGIDHLPVR
>1e9x
MVVQLAQPACV
>2hpd
LTLKPEGFVVK
>1dt6
FVSVPPSYQLC
```

Figure A 2-8 Fichier de l'ensemble de blocs de M. Cotevieille. Celui-ci contient 27 sous-alignements de quatre structures de référence : 1oxa, 1e9x, 2hpd, 1dt6. Dans ce jeu, une seule structure provient des mammifères : il s'agit du 1dt6, le CYP 2C5 de lapin.

2.2.3 Format des fichiers de GROMACS

Le programme *pdb2gmx* génère à partir d'un fichier de structure PDB, deux fichiers : i) un fichier de topologie contenant toutes les interactions présentes à l'intérieur de la protéine, et ii) un ficher de structure spécifique à GROMACS.

2.2.3.1 Fichier de Topologie (.top ou .itp)

Le fichier de topologie est un fichier ascii (un fichier texte) contenant les informations d'interactions d'une protéine ou un composé, généré à partir de *pdb2gmx* ou du serveur PRODRG.

```
;       Exemple de fichier ITP généré par le serveur
;       PRODRG (pour le CPA)
;
;

[ moleculetype ]
; Name nrexcl
CPA      3

[ atoms ]
;   nr       type  resnr resid  atom  cgnr    charge      mass
     1         CL     1  CPA    CLAK     1   -0.050  35.4530
     2        CH2     1  CPA     CAJ     1    0.025  14.0270
     3        CH2     1  CPA     CAI     1    0.025  14.0270
     4         NL     1  CPA     NAH     2   -0.281  14.0067
     5        CH2     1  CPA     CAL     2    0.015  14.0270
     6        CH2     1  CPA     CAM     2    0.015  14.0270
     7         CL     1  CPA    CLAN     2   -0.086  35.4530
     8          P     1  CPA      P8     2    1.337  30.9738
     9         OM     1  CPA      O9     3    0.000  15.9994
    10         OS     1  CPA     OAF     4   -0.425  15.9994
    11        CH2     1  CPA     CAG     4    0.007  14.0270
    12        CH2     1  CPA     CAA     4    0.007  14.0270
    13        CH2     1  CPA     CAB     4    0.007  14.0270
    14         NL     1  CPA     NAC     4   -0.596  14.0067
    15          H     1  CPA     HAA     5    0.000   1.0080

[ bonds ]
; ai  aj  fu    c0, c1, ...
   1   2   1    0.178    394315.8    0.178    394315.8 ;  CLAK  CAJ
   2   3   1    0.153    334720.0    0.153    334720.0 ;   CAJ  CAI
   3   4   1    0.147    376560.0    0.147    376560.0 ;   CAI  NAH
   4   5   1    0.147    376560.0    0.147    376560.0 ;   NAH  CAL
   4   8   1    0.162    304571.8    0.162    304571.8 ;   NAH   P8
   5   6   1    0.153    334720.0    0.153    334720.0 ;   CAL  CAM
   6   7   1    0.178    394315.8    0.178    394315.8 ;   CAM CLAN
   8   9   1    0.148    376560.0    0.148    376560.0 ;    P8   O9
   8  10   1    0.161    251040.0    0.161    251040.0 ;    P8  OAF
   8  14   1    0.162    304571.8    0.162    304571.8 ;    P8  NAC
  10  11   1    0.143    251040.0    0.143    251040.0 ;   OAF  CAG
  11  12   1    0.153    334720.0    0.153    334720.0 ;   CAG  CAA
  12  13   1    0.153    334720.0    0.153    334720.0 ;   CAA  CAB
  13  14   1    0.147    376560.0    0.147    376560.0 ;   CAB  NAC
  14  15   1    0.100    374468.0    0.100    374468.0 ;   NAC  HAA

[ pairs ]
; ai  aj  fu    c0, c1, ...
   1   4   1                                          ;  CLAK  NAH
   2   5   1                                          ;   CAJ  CAL
   2   8   1                                          ;   CAJ   P8
   3   6   1                                          ;   CAI  CAM
   3   9   1                                          ;   CAI   O9
   3  10   1                                          ;   CAI  OAF
   3  14   1                                          ;   CAI  NAC
   4   7   1                                          ;   NAH CLAN
   4  11   1                                          ;   NAH  CAG
   4  13   1                                          ;   NAH  CAB
   4  15   1                                          ;   NAH  HAA
   5   9   1                                          ;   CAL   O9
   5  10   1                                          ;   CAL  OAF
   5  14   1                                          ;   CAL  NAC
   6   8   1                                          ;   CAM   P8
   8  12   1                                          ;    P8  CAA
   9  11   1                                          ;    O9  CAG
   9  13   1                                          ;    O9  CAB
   9  15   1                                          ;    O9  HAA
  10  13   1                                          ;   OAF  CAB
  10  15   1                                          ;   OAF  HAA
  11  14   1                                          ;   CAG  NAC
  12  15   1                                          ;   CAA  HAA
```

```
[ angles ]
; ai  aj  ak  fu  c0, c1, ...
   1   2   3   1  111.0   460.2  111.0  460.2 ;  CLAK  CAJ  CAI
   2   3   4   1  109.5   460.2  109.5  460.2 ;   CAJ  CAI  NAH
   3   4   5   1  109.5   376.6  109.5  376.6 ;   CAI  NAH  CAL
   3   4   8   1  109.5   376.6  109.5  376.6 ;   CAI  NAH   P8
   5   4   8   1  109.5   376.6  109.5  376.6 ;   CAL  NAH   P8
   4   5   6   1  109.5   460.2  109.5  460.2 ;   NAH  CAL  CAM
   5   6   7   1  111.0   460.2  111.0  460.2 ;   CAL  CAM CLAN
   4   8   9   1  109.6   397.5  109.6  397.5 ;   NAH   P8   O9
   4   8  10   1  103.0   397.5  103.0  397.5 ;   NAH   P8  OAF
   4   8  14   1  103.0   397.5  103.0  397.5 ;   NAH   P8  NAC
   9   8  10   1  109.6   397.5  109.6  397.5 ;    O9   P8  OAF
   9   8  14   1  109.6   397.5  109.6  397.5 ;    O9   P8  NAC
  10   8  14   1  103.0   397.5  103.0  397.5 ;   OAF   P8  NAC
   8  10  11   1  120.0   397.5  120.0  397.5 ;    P8  OAF  CAG
  10  11  12   1  109.5   460.2  109.5  460.2 ;   OAF  CAG  CAA
  11  12  13   1  111.0   460.2  111.0  460.2 ;   CAG  CAA  CAB
  12  13  14   1  109.5   460.2  109.5  460.2 ;   CAA  CAB  NAC
   8  14  13   1  109.5   376.6  109.5  376.6 ;    P8  NAC  CAB
   8  14  15   1  109.5   376.6  109.5  376.6 ;    P8  NAC  HAA
  13  14  15   1  109.5   376.6  109.5  376.6 ;   CAB  NAC  HAA

[ dihedrals ]
; ai  aj  ak  al  fu  c0, c1, m, ...
   4   3   8   5  2 35.3  836.8    35.3  836.8   ; imp  NAH  CAI   P8  CAL
   8   4  10   9  2 35.3  836.8    35.3  836.8   ; imp   P8  NAH  OAF   O9
  14   8  13  15  2 35.3  836.8    35.3  836.8   ; imp  NAC   P8  CAB  HAA
   4   3   2   1  1  0.0    5.9 3   0.0    5.9 3 ; dih  NAH  CAI  CAJ CLAK
   2   3   4   8  1  0.0    3.8 3   0.0    3.8 3 ; dih  CAJ  CAI  NAH   P8
   6   5   4   3  1  0.0    3.8 3   0.0    3.8 3 ; dih  CAM  CAL  NAH  CAI
   3   4   8  14  1  0.0    1.0 3   0.0    1.0 3 ; dih  CAI  NAH   P8  NAC
   3   4   8  14  1  0.0    3.1 2   0.0    3.1 2 ; dih  CAI  NAH   P8  NAC
   7   6   5   4  1  0.0    5.9 3   0.0    5.9 3 ; dih CLAN  CAM  CAL  NAH
  11  10   8   4  1  0.0    1.0 3   0.0    1.0 3 ; dih  CAG  OAF   P8  NAH
  11  10   8   4  1  0.0    3.1 2   0.0    3.1 2 ; dih  CAG  OAF   P8  NAH
   4   8  14  15  1  0.0    1.0 3   0.0    1.0 3 ; dih  NAH   P8  NAC  HAA
   4   8  14  15  1  0.0    3.1 2   0.0    3.1 2 ; dih  NAH   P8  NAC  HAA
   8  10  11  12  1  0.0    3.8 3   0.0    3.8 3 ; dih   P8  OAF  CAG  CAA
  13  12  11  10  1  0.0    5.9 3   0.0    5.9 3 ; dih  CAB  CAA  CAG  OAF
  14  13  12  11  1  0.0    5.9 3   0.0    5.9 3 ; dih  NAC  CAB  CAA  CAG
  12  13  14  15  1  0.0    3.8 3   0.0    3.8 3 ; dih  CAA  CAB  NAC  HAA
```

Figure A 2-9 Exemple de fichier .itp généré par le serveur PRODRG pour le CPA. Le fichier .top est similaire à ce fichier.

2.2.3.2 Fichier de Coordonnées (.gro)

Lorsque le programme *pdb2gmx* est exécuté pour générer une topologie moléculaire, il traduit également le fichier de structure (au format .pdb) vers un fichier de structure Gromos87 (au format .gro). La différence majeur entre le fichier au format pdb et le fichier au format Gromos87 est d'une part le formatage du fichier, ainsi que le fait qu'un fichier .gro contient également les vitesses. Toutefois, il n'est pas toujours nécessaire d'utiliser les vitesses, c'est pourquoi le fichier PDB est plus commun pour tous les programmes. Pour créer une boîte d'eau autour de la molécule, le programme *genbox* est appelé couplé au programme *editconf* qui défini préalablement la taille de la boîte. Le fichier de sortie du programme *genbox* est également un fichier de type Gromos87. Les fichiers au format Gromos87 peuvent être utilisés comme des trajectoires, simplement en les concaténant. En effet, dans le titre de chaque fichier, le temps peut être signalé par 't =' suivi d'un temps en ps.

```
MD of 2 waters, t= 0.0
    6
    1WATER  OW1    1   0.126   1.624   1.679  0.1227 -0.0580  0.0434
    1WATER  HW2    2   0.190   1.661   1.747  0.8085  0.3191 -0.7791
    1WATER  HW3    3   0.177   1.568   1.613 -0.9045 -2.6469  1.3180
    2WATER  OW1    4   1.275   0.053   0.622  0.2519  0.3140 -0.1734
    2WATER  HW2    5   1.337   0.002   0.680 -1.0641 -1.1349  0.0257
    2WATER  HW3    6   1.326   0.120   0.568  1.9427 -0.8216 -0.0244
   1.82060   1.82060   1.82060
```

Figure A 2-10 Exemple de fichier .gro pour deux molécules d'eau.

Le formatage du fichier .gro est décomposé de la manière suivante (du haut vers le bas) :

- Une ligne de titre (dans un format libre, avec optionnellement en temps en ps après 't =')
- Le nombre d'atome (dans un format libre, mais de type entier)
- Une ligne pour chaque atome (dans un format fixe, voir plus bas)
- Dimension de la boîte (dans un format libre, de type entier séparé par des espaces), les valeurs étant v1(x) v2(y) v3(z) v1(y) v1(z) v2(x) v2(z) v3(x) v3(y). Les six dernières valeurs peuvent être omises (elles seront mises à zéro). GROMACS ne supporte que les boites avec v1(y)=v1(z)=v2(z)=0.

Pour le format fixe, toutes les colonnes sont à une position fixée. Ces colonnes portent les informations suivantes (de gauche à droite) :

- Numéro du résidu (5 positions, entier)
- Nom du résidu (5 caractères)
- Nom de l'atome (5 caractères)

- Position (en nm, x y z dans 3 colonnes, chacun 8 positions avec 3 places pour les décimales)
- Vitesse (en nm/ps (ou km/s), x y z dans trois colonnes, chacun 8 poisitons avec 4 places pour les décimales)

ANNEXE 3

Caliseq : dissection d'un programme

« Ne cherchons pas hors de nous le mal, il est chez nous, il est planté en nos entrailles.»
Michel de Montaigne (1533 – 1692 ap JC)

3.1 Introduction

Cette annexe a pour objet une description plus poussée de l'outil *Caliseq*. Il peut être considéré comme le manuel de référence du programme, décrivant aussi bien la procédure à suivre pour lancer le programme mais également comment transitent les informations de leurs saisies par l'utilisateur, à leur affichage en sortie de programme. La première partie de ce chapitre décrira donc l'utilisation de l'outil *Caliseq* : qu'attend le programme, quelles sont les options nécessaires à paramétrer, quelles sont celles optionnelles et pour quelles intérêts. Cette première description est en quelque sorte une présentation générale d'un point de vu externe au programme, comme une boîte noire qu'on se servirait sans prêter attention aux mécanismes intrinsèques. Le « contenant » du programme sera vu ensuite plus en détails, avec une présentation succincte des fichiers sources de *Caliseq*, où le code du programme est écrit, et comment compiler ce dernier. Enfin, dans une dernière partie, chaque fonction de l'outil sera expliquée, accompagné de leur algorithme dans un langage simple et non celui du programme lui-même, afin qu'un lecteur non initié puisse quand même comprendre son essence, son principe.

3.2 Les paramètres en entrée de *Caliseq*

Il existe deux manière de faire tourner *Caliseq* : l'une pour un alignement simple des blocs sur une séquence ou un pré-alignement de séquences, et l'autre, de faire confronter le jeu de bloc sur une banque de données de séquence en vu de déterminer des P450s. Pour la suite de cette annexe, l'alignement simple des blocs sur une séquence (ou un pré-alignement de séquences) sera décrit comme le « mode normale » de *Caliseq*, en opposition avec le « mode d'alignement sur banque » correspondant à l'alignement des blocs sur une banque de séquences. Dans les deux cas, deux fichiers doivent être fournis en entrée au programme : i) un fichier contenant le jeu de bloc (appelé *cons_file* dans le programme) dont le contenu a été présenté dans le chapitre précédent, et ii) un fichier contentant soit la séquence cible, ou son pré-alignement avec d'autres séquences à fort taux d'identité, soit une banque où chaque enregistrement figure les uns à la suite des autres (appelé *ali_file* dans le programme).

La différence pour distinguer les deux modes de fonctionnement se fera alors sur les autres paramètres insufflés en entrée. Le listing de l'help du logiciel *SmartConsAlign* montré à la Figure A 3-1, présente les diverses possibilités du logiciel. *Caliseq* étant une extension de *SmartConsAlign*, certaines options sont requises pour faire fonctionner *SmartConsAlign* en mode *Caliseq*.

```
[777]nguyen@kezia:~/> smartconsalign -h
Usage: smartconsalign [-h] [-v] [-B <nbseq>] [-N] [-b|-c [-k <nbpos>]| -s [-k <nbpos>]]
       [-m matrix_file] [-C <1|2|3>] [-g <gap_penalty>]
       [-e <gap_extension_penalty>] [-d <gap_penalty against gap>]
       [-o|-O] [-A|F|M] [-S|T|L|Z <score>] ali_file cons_file
 === options ====================================
 --- general options ----------------------------
   -h : this help
   -v : verbose mode
   -B <nbseq>: ali_file is a database and the program will process each entry
               keeping <nbseq> sequences (need the –N option to work)
   -N : cons_file is a multiple consensus (Caliseq program)
 --- alignment mode -----------------------------
   -b : bestfit mode (default mode)
   -s : smith-waterman mode
   -c : PSSM consensus mode (2nd file must be consensus file, print only
solutions)
 --- in multiple consensus mode (-N option) ------
   -a : sequences are ADN sequences
   -I : search also on complementary sequences
   -w <len> : ali_file is a database and the program will process each entry
with window length <len>
   -p <len> : step for moving window
   -t <threshold> : threshold for outputting results in window mode
   -G : scan mode (in window mode), output window positions and scores along thesequences
 --- alignment options --------------------------
   -m <matrix_file> :
        in bestfit/SW mode : matrix is used for alignment
        in consensus mode  : compute weight from matrix_file matrix
   -C <code for conversion of PAM/BLOSUM/proximity matrix into distance matrix>
        1: distance(i,j) is Sqrt[Sum(((i,k)-(k,j))^2) for all k]
        2: distance(i,j) is 1-((prox(i,j)-min)/(max-min))
        3: Chi2 method: (i,j) are turned into probabilities,
           and dist(i,j) is Sqrt[Sum((p(i,k)p(k,j))^2) for all k]
        default mode is mode 1
 --- gap options --------------------------------
    Gaps costs are input as ratio, if MAX_VAL is the maximum value in
    the substitution matrix (BLOSUM for example) and MIN_VAL is the
    minimum value, gap cost are computed as:
           gap cost = MAX_VAL - ratio * (MAXVAL-MINVAL)
   -g <gap_opening_cost> :
        in bestfit/SW mode : cost ratio for opening gap (default:  1.20)
   -e <gap_extension_cost> :
        in bestfit/SW mode : cost ratio for extending gap (default:  0.80)
   -d : cost ratio for insertions-deletions against gap (default:  0.50)
 --- output options -----------------------------
   -A : output alignment in CLUSTAL format (in bestfit, smith-waterman or multiple consensus
mode only)
   -F : output alignment in FASTA format (in bestfit, smith-waterman or multiple consensus mode
only)
   -M : output alignment in SmartMulti format (in bestfit or smith-waterman mode only)
   -o : output results as consensus with no gap (in bestfit mode only)
   -O : output results as consensus with gaps (in bestfit mode only)
   -k : Number of kept positions in PSSM consensus mode/default 10
          or number of optimal alignments to keep in smith-waterman mode/default 10
   -S <float value> : minimum score threshold in smith-waterman (default:  5.00)
   -T <float value> : ratio for the score threshold in consensus (default: 0.50)
         (A random sequence is used for mean score)
   -Z <float value> : Zscore threshold in consensus (default:  0.00)
   -L <integer value> : cut off between 1st bloc and last bloc for db scanning (default:    NO)
 === arguments ==================================
    ali_file, cons_file : files with single sequence, multiple
        alignment (Clustal, Fasta or MSF)  or consensus
        2 files are mandatory, and sequences in the second file
        (cons_file) must be shorter than those in the
        first file (ali_file))
```

Figure A 3-1 Listing de l'aide du logiciel *SmartConsAlign*. Pour « activer » l'outil *Caliseq*, il suffit d'ajouter en ligne de commande l'option « –N ». Le premier fichier « ali_file » correspond alors à la séquence cible ou le pré-alignement ou la banque de séquence, tandis que le second fichier correspond au fichier de blocs. L'option « –B » suivi d'un nombre (nombre de séquences maximales à récupérer) doit être ajouté au précédent, si on désire scanner les banques à la recherche de P450. D'autres options peuvent être informées tels que le z-score « –Z », le poids des gap « –g », la longueur maximale entre les blocs « –L », tous suivi d'un chiffre décimal (ou d'un entier pour le dernier).

En fait, l'appel à l'outil *Caliseq* se fait au moyen de l'option « –N » qui informe alors à *SmartConsAlign* que le premier fichier d'entrée est un alignement de séquence, que le second fichier contient le jeu de blocs dans une nomenclature particulière décrite dans le chapitre précédent, et enfin que ces deux fichiers doivent être traités par l'outil *Caliseq*. Pour différencier les deux modes de fonctionnement de *Caliseq*, une option supplémentaire est nécessaire : « –B » : dans ce cas, le premier fichier correspond à une banque de séquences, où chaque enregistrement est traité individuellement, les uns à la suite des autres. Par ailleurs, l'option « –B » requière un second paramètre correspondant au nombre maximale de séquences que l'utilisateur désire récupérer de la banque. Comme il sera vu plus tard, ces séquences récupérées sont alors affichées selon un classement établi sur le score d'alignement des blocs sur la séquence. En fait, l'option « –B » va principalement influencer sur la manière de lire et de traiter le fichier « ali_file ».

D'autres options additionnelles peuvent être ajoutées à ces options globales, certains communs aux deux modes, d'autres spécifiques à un des deux modes. Ainsi, l'option « –g » suivit d'un nombre réel est commun aux deux modes : il permet à l'utilisateur d'influer sur le poids attribué aux gaps. En réalité, la valeur indiquée par l'utilisateur ne correspond pas au poids donné aux gaps proprement dit, mais à un coefficient défini, rappelé comme suit : si MAX_VAL est la valeur maximale dans la matrice de substitution (BLOSUM par exemple) et MIN_VAL la valeur minimale, alors le poids de gap est égale à *MAX_VAL – coefficient * (MAX_VAL – MIN_VAL)*. Associer le poids de gap à un coefficient permet une plus grande souplesse et adaptabilité du programme, notamment lors d'utilisation de matrices de similarités différentes (PAM à la place de BLOSUM etc…) pour les alignements. L'option « –A » permet quant à lui d'obtenir les résultats d'alignement au format Clustal (par défaut, c'est le format fasta qui est affiché). Deux autres options peuvent être utilisés, mais cette fois-ci spécifiques au mode de recherche sur banque : les options « – Z » pour définir un seuil de z-score et « – L » pour déterminer la longueur maximale entre le premier et le dernier bloc, que l'utilisateur autorise au programme pour récupérer des séquences identifiées comme P450. Ces deux options, comme il le sera décrit plus tard au cours de l'annexe, permettent d'influer sur le nombre de séquences de P450 récupérées mais surtout de filtrer les faux positifs. Un résumé de toutes les options utilisées pour *Caliseq* est présenté dans la Figure A 3-2.

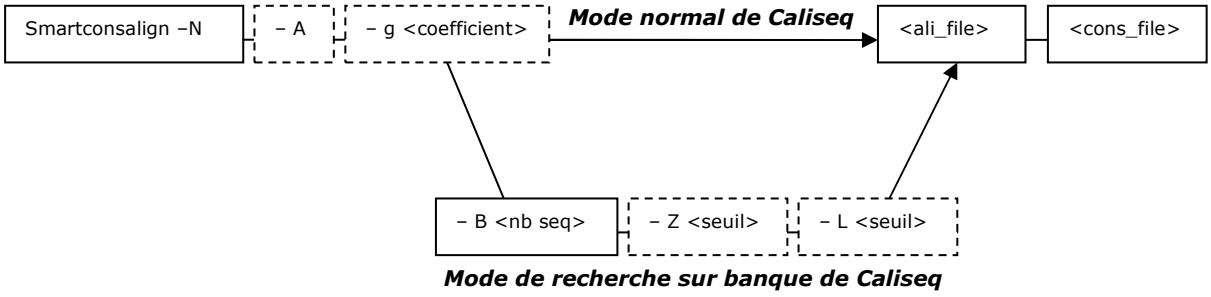


Figure A 3-2 Exemple de ligne de commande pour lancer *Caliseq*. Les options nécessaires au fonctionnement de *Caliseq* sont inscrites dans des rectangles à trait plein, tandis que les options facultatives sont inscrites dans des rectangles à bords en pointillés. L'option « –N » est obligatoire pour informer le programme *SmartConsAlign* de faire appel à l'outil *Caliseq*. Les options « –A » et « –g » commun aux deux modes de *Caliseq* sont facultatives et permettent l'affiche du résultat en format Clustal (le format fasta étant le format par défaut) ainsi que de modifier le poids attribué aux gaps respectivement. L'option « –B » informe à *Caliseq* de fonctionner en mode « recherche » sur banque. Les options supplémentaires tels que « –Z » et « –L » peuvent venir s'y greffer pour filtrer les résultats. Ali_file dans tous les cas est un fichier comprenant un ou plusieurs enregistrements au format fasta. Selon le mode, il sera lu et traité différemment par *Caliseq*. Cons_file est un fichier unique comprenant les jeux de blocs.

3.3 Fichiers sources de Caliseq

SmartConsAlign a été conçu au départ comme un ensemble d'outils, dont la particularité commune est de réaliser un alignement de courts alignements de séquences (aussi bien protéique que nucléique) écrits sous forme de matrice consensus, sur une autre séquence cible seule ou pré alignée avec d'autres séquences. Le programme comprend actuellement une vingtaine de fichiers sources (cf. le listing de la Figure A 3-3) qui regroupent chacun des fonctions spécifiques, destinées à une tâche particulière.

```
[778]nguyen@kezia:~/Src/SmartConsAlign>ls
total 1216
-rwxr-xr-x    1 thienan  users        1838 Jul 27  2006 ADN
-rw-r--r--    1 thienan  users        2071 Jul 27  2006 BLOSUM62
-rwxr-xr-x    1 thienan  users      185587 Jul 27  2006 cons_align
-rw-r--r--    1 thienan  users       79841 Jul 27  2006 cons_align.c
-rw-r--r--    1 thienan  users        4444 Jul 27  2006 cons_align.h
-rw-r--r--    1 thienan  users       34532 Nov 14  2006 cons_align.o
-rw-r--r--    1 thienan  users       35079 Jul 27  2006 cons_seqs_bestfit.c
-rw-r--r--    1 thienan  users        1810 Jul 27  2006 cons_seqs_bestfit.h
-rw-r--r--    1 thienan  users        6676 Nov 14  2006 cons_seqs_bestfit.o
-rw-r--r--    1 thienan  users       49217 Jul 27  2006 cons_seqs_sw.c
-rw-r--r--    1 thienan  users        1482 Jul 27  2006 cons_seqs_sw.h
-rw-r--r--    1 thienan  users        7916 Nov 14  2006 cons_seqs_sw.o
-rw-r--r--    1 thienan  users        9187 Jul 27  2006 fasta_io.c
-rw-r--r--    1 thienan  users        2698 Jul 27  2006 fasta_io.h
-rw-r--r--    1 thienan  users        3700 Nov 14  2006 fasta_io.o
-rw-r--r--    1 thienan  users         438 Jul 27  2006 macros.h
-rw-r--r--    1 thienan  users        2162 Jul 27  2006 Makefile
-rw-r--r--    1 thienan  users        7049 Nov 14  2006 minimier.c
-rw-r--r--    1 thienan  users         397 Jul 27  2006 minimier.h
-rw-r--r--    1 thienan  users        3216 Nov 14  2006 minimier.o
-rw-r--r--    1 thienan  users       55470 Nov 14  2006 mulcons.c
-rw-r--r--    1 thienan  users        1616 Jul 28  2006 mulcons.h
-rw-r--r--    1 thienan  users       16948 Nov 14  2006 mulcons.o
-rw-r--r--    1 thienan  users       37597 Jul 27  2006 Nmatrix.c
-rw-r--r--    1 thienan  users        3811 Jul 27  2006 Nmatrix.h
-rw-r--r--    1 thienan  users       14436 Nov 14  2006 Nmatrix.o
-rw-r--r--    1 thienan  users       15683 Jul 27  2006 README
-rw-r--r--    1 thienan  users       35959 Jul 27  2006 readmultali.c
-rw-r--r--    1 thienan  users        2007 Jul 27  2006 readmultali.h
-rw-r--r--    1 thienan  users       18116 Nov 14  2006 readmultali.o
-rwxr-xr-x    1 thienan  users      110362 Nov 14  2006 smartconsalign
-rw-r--r--    1 thienan  users       24490 Nov 14  2006 smartconsalign.c
-rw-r--r--    1 thienan  users        2497 Jul 28  2006 smartconsalign.h
-rw-r--r--    1 thienan  users       19816 Nov 14  2006 smartconsalign.o
-rw-r--r--    1 thienan  users        5083 Jul 27  2006 smutil.c
-rw-r--r--    1 thienan  users         668 Jul 27  2006 smutil.h
-rw-r--r--    1 thienan  users        2484 Nov 14  2006 smutil.o
-rw-r--r--    1 thienan  users        7933 Jul 27  2006 waterstat.c
-rw-r--r--    1 thienan  users        1474 Jul 27  2006 waterstat.h
-rw-r--r--    1 thienan  users        3312 Nov 14  2006 waterstat.o
```

Figure A 3-3 Listing du répertoire comprenant les fichiers sources de *SmartConsAlign*

Ainsi, c'est dans le fichier **mulcons.c** et sa librairie **mulcons.h** qu'on retrouve essentiellement toutes les fonctions et déclarations de variables nécessaire au fonctionnement de *Caliseq*. *Caliseq* a néanmoins besoin du fichier source principal, smarconsalign.c, les fichiers additionnels Nmatrix.c, readmultali.c, fasta_io.c, smulti.c, minimier.c ainsi que leurs librairies respectives, car ces derniers comportent quelques fonctions utiles et nécessaires à son utilisation. Par exemple, l'outil *Caliseq* est lancé à partir de la procédure principale (« `Main()` » en langage C) définie dans le fichier source smartconsalign.c. Il peut lire et écrire des fichiers de séquences au format fasta grâce des fonctions écrites dans readmultali.c et fasta_io.c. Les créations de matrices de fréquences et de similarités sont réalisées par des fonctions contenues dans Nmatrix.c tandis que smulti.c regroupe des outils pour l'indexation des séquences afin que ces dernières soient traitées plus rapidement etc. Pour faire simple, on peut dire que tous ces fichiers sources se comportent comme de vastes bibliothèques de fonctions, déjà écrites pour les différents outils présents dans *SmartConsAlign.* Ces fonctions ont été développées de manière suffisamment modulaires pour être réintégré dans *Caliseq* sans en toucher le contenu : il me suffisait d'adapter dans la majeure partie du temps mon code et utiliser correctement les variables

attendues par ces fonctions en arguments[22]. À d'autres moments, pourtant, il a fallu réécrire certaines de ces fonctions préexistantes ou du moins, les arranger pour mes besoins en veillant à ce que ces modifications ne perturbent pas les autres outils de *SmartConsAlign* utilisant ces fonctions. Ainsi, même si la majeur partie des fonctions que j'ai écrites pour *Caliseq* est retrouvée dans les fichiers source mulcons.c et mulcons.h, certaines fonctions vitales pour le programme se retrouve disséminées dans les autres fichiers sources. En conséquence, même si *Caliseq* se comporte comme une « entité » individuelle dans *SmartConsAlign*, il a toutefois besoin du programme *SmartConsAlign* dont il dépend et de toutes les fonctions relatives à ce dernier. L'ensemble des fichiers sources et de leurs librairies respectives est regroupé et compilé par le programme *Makefile*.

3.4 Quelques fonctions de *Caliseq* au peigne fin

Après avoir décrit les paramètres en entrée au programme (qui serviront donc d'arguments d'entrée à la fonction `Main()`, fonction principale du programme *SmartConsAlign*) ainsi que les fichiers sources, qui servent de « contenant » aux fonctions permettant le déroulement du programme, je propose d'examiner chaque fonction du programme l'un après l'autre, en essayant de rester simple et en expliquant leur grande ligne, sans trop entrer dans les détails. Pour comprendre l'enchevêtrement ainsi que la connectivité entre toutes les fonctions, deux diagrammes du programme sont proposés sur les Figure A 3-4 et Figure A 3-5 permettant ainsi de suivre le flux d'information. Le Tableau A 3-1 permet également de retrouver les caractéristiques de chaque fonction. Par souci de clarté, certaines fonctions seront regroupées ou éludées : le programme étant modulaire, des fonctions ont été développées pour éviter la répétition de code ou pour un besoin particulier au langage de programmation, mais ne présentent pas d'intérêt explicatif pour le fonctionnement global du programme lui-même.

[22] On parle de passage d'arguments aux fonctions, lorsqu'on envoie des variables dans ces fonctions pour être traitées. Le « type » de variable adressée à la fonction doit être similaire à celui attendu par la fonction. Ainsi, si la fonction attend une variable de type « entier », il faudra fournir en entrée à cette fonction une variable de type « entier ».

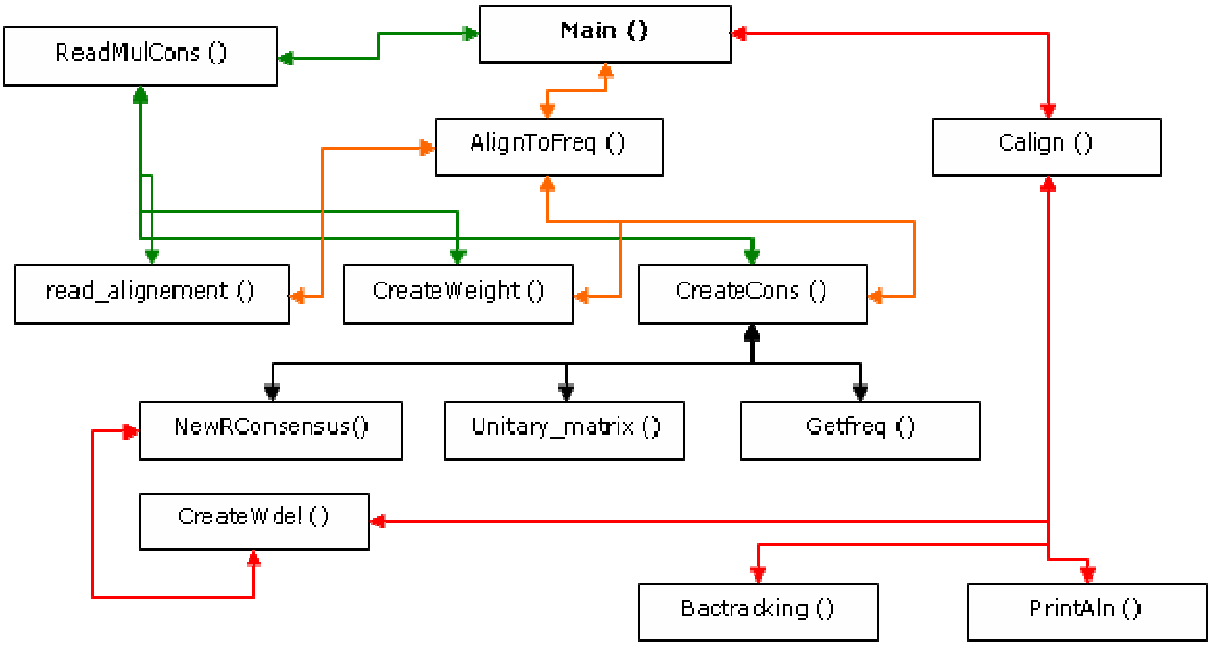


Figure A 3-4 Diagramme du programme Caliseq dans son premier mode. L'ordre d'appel des fonctions se fait de la gauche vers la droite et du haut vers le bas. L'ordre prioritaire est donc le suivant : haut > gauche > bas > droite. Les fonctions sont regroupées par des flèches de couleurs pour pouvoir suivre la chronologie

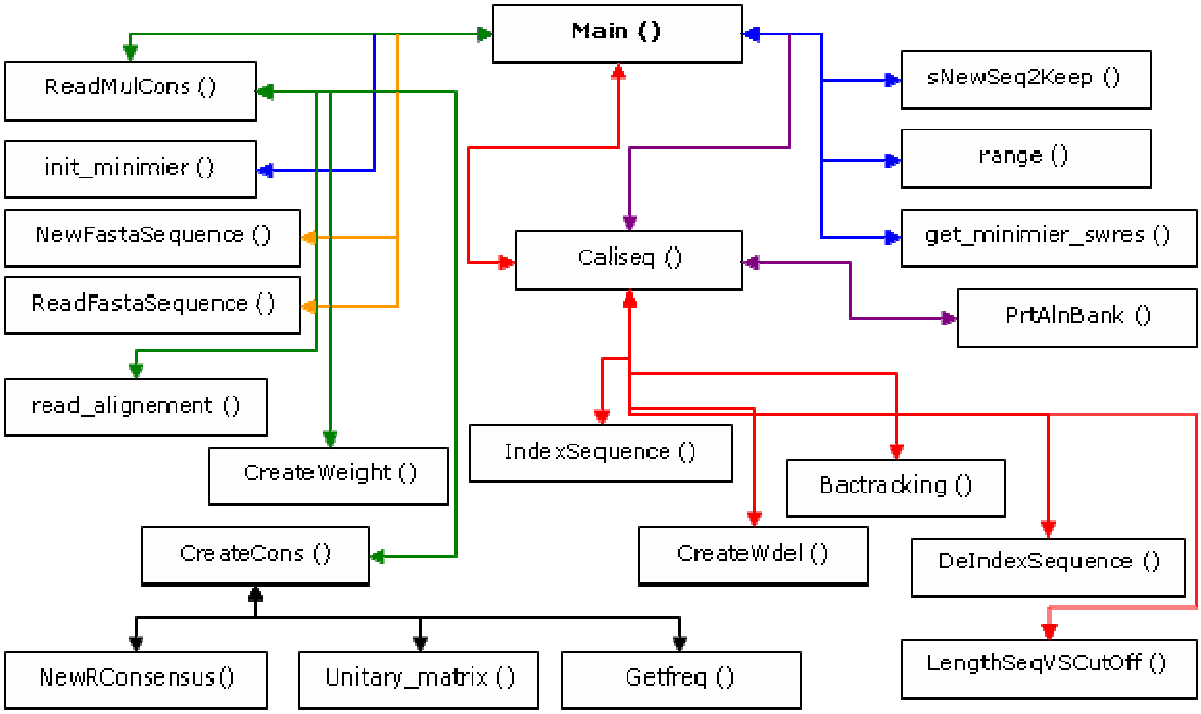


Figure A 3-5 Diagramme du programme Caliseq dans son mode de recherche sur banque avec les mêmes annotations que précédemment. Caliseq est appelé deux fois (flèches rouges puis violettes).

Tableau A 3-1 Tableaux récapitulatif des principales fonctions usitées par *Caliseq*

Nom	***Rôle***	***Appelé par***	***Localisation***
AlignToFreq	Uniquement dans le mode normal de *Caliseq*, cette procédure est l'équivalent de ReadMulCons mais pour l'alignement de la séquence cible. Appelle les mêmes fonctions que ReadMulCons	Main	mulcons.c
Backtracking	Permet de remonter le chemin optimal en fin de la programmation dynamique	Calign Caliseq	mulcons.c
Calign	C'est la procédure d'alignement de Caliseq en mode normal et contient l'algorithme de programmation dynamique. Il appelle Backtracking et PrintAln	Main	mulcons.c
Caliseq	C'est la procédure d'alignement de Caliseq en mode de recherche sur banque. Il est appelé deux fois : la première pour retenir les séquences présentant les meilleurs scores d'alignement et calculer les termes nécessaires au Z-score et la seconde fois pour afficher les séquences répondant aux critères de sélection. Appelle IndexSequence, DeIndexsequence, CreateWdel, Bactracking et PrtAlnBank	Main	mulcons.c
CreateCons	Construit des profils à partir des alignements de séquences. Apelle NewRConsensus, unitary_matrix et Getfreq	ReadMulCons AlignToFreq	mulcons.c
CreateWdel	Creation de la « matrice » de délétion à partir de la « matrice » de substitution (au sens artifice de programmation)	Calign Caliseq	mulcons.c
CreateWeight	Établi une pondération pour les séquences trop identiques dans l'alignement	ReadMulCons AlignToFreq	mulcons.c
DeIndexSequence	Désindexation de la séquence à son état d'origine	Caliseq	smulti.c
Get_minimier_swres	Renvoie de la séquence de l'arbre au rang voulu	Main	minimier.c
Getfreq	Renvoie la fréquence d'un acide aminé dans la banque Swiss-Prot	CreateCons	mulcons.c
IndexSequence	Indexation de la séquence cible au format fasta pour l'alignement par programmation dynamique	Caliseq	smulti.c
init_minimier	Permet l'initialisation et la construction de l'arbre qui retient les séquences identifiés comme P450s dans le mode de recherche sur banque	Main	minimier.c
LengthSeqVSCutOff	Vérifie si l'alignement finale n'est pas supérieur au cuttoff « -L »	Caliseq	mulcons.c
Main	Chef d'orchestre du programme, récupère les paramètres d'entrée et lance le programme Caliseq selon un des modes.	utilisateur	smartconsalign.c
NewFastaSequence	Allocation d'une variable pour recevoir une séquence fasta	Main	fasta_io.c
NewRConsensus	Alloue de la place pour sauvegarder les profils	CreateCons CreateWdel	cons_align.c
PrintAln	Procédure permettant de créer l'alignement finale à partir du chemin obtenu par le Backtracking. Appelle deux fonctions non représentées sur les diagrammes qui permettent d'afficher les alignements soit au format clustal, soit au format fasta (par défaut)	Calign	mulcons.c
PrtAlnBank	Procédure similaire à PrintAln, mais pour une seule séquence	Caliseq	mulcons.c
range	Permet de ranger dans l'arbre la séquence récupérée en fonction du score d'alignement obtenu	Main	minimier.c
read_alignement	Permet la lecture de fichiers d'alignement de séquences et la sauvegarde de ces séquences dans une structure	ReadMulCons AlignToFreq	readmultali.c
ReadFastaSequence	Procédure permettant la lecture d'une séquence fasta	Main	fasta_io.c
ReadMulCons	Procédure permettant la lecture et la conversion des blocs en profil. Appelle read_alignement, CreateWeight et CreateCons	Main	mulcons.c
sNewSeq2Keep	Création d'un variable pour retenir la séquence dans l'arbre	Main	minimier.c
unitary_matrix	Initialise à zéro une matrice de fréquences sauf au niveau des diagonales où la valeur est de un	CreateCons	Nmatrix.c

3.4.1 La fonction Main() de smartconsalign.c

La fonction principale `Main()` de smartconsalign.c fait partie de ces fonctions préexistantes au programme *SmartConsAlign*. Elle permet entre autre de diriger et de lancer les différentes applications du « package » *SmartConsAlign*. C'est également une des fonctions que j'ai dû modifier pour insérer le module *Caliseq*. En tant que « chef d'orchestre » de l'ensemble du programme, la fonction `Main()` (pour « principal » en anglais) est à l'interface entre les directives de l'utilisateur, communiquées via les paramètres d'entrées, et la transmission de ces directives aux différentes applications du programme. C'est d'ailleurs dans cette fonction que sont déclarées la plupart des variables utilisées tout au long du programme. La fonction `Main()` se charge dans un premier temps de récupérer tous les arguments optionnels (comme par exemple « –N » ou « –g » etc…) en entrée du programme et de les traiter à l'aide d'une fonction prédéfinie dans les librairies spécifiques du langage C, `getopt()`. Des documentations existent pour cette librairie, je ne la décrirai donc pas. L'appel à cette fonction `getopt()` se fait alors au moyen d'une structure conditionnelle multiple[23] de type « Switch/Case » : pour chaque paramètre optionnel lu et retourné au programme par `getopt()`, des instructions sont associées. Ainsi, pour introduire *Caliseq*, il m'a fallu définir des instructions pour le cas où `getopt()` rencontrerait les options « –N », « –B », « –Z », « –L » et « –T », un ancien paramètre ne sers plus pour l'alignement protéique, mais est encore utilisé pour l'alignement nucléiques. Pour la plupart des instructions, il s'agit de déclarer des marqueurs (aussi appelés balises) : la propriété qu'une variable soit vide ou remplie est alors utilisée. En conséquent, lorsque `getopt()` renvoie « –N », une balise signalant que le programme doit utiliser l'outil *Caliseq* est activé. Cette manœuvre est suivie par la récupération et sauvegarde dans des variables appropriés, les noms des deux fichiers lus par le programme (ali_file et cons_file). Comme la balise signalant que *Caliseq* doit être utilisé est « activée », le programme peut entre alors dans la partie du programme spécifique à l'utilisation de *Caliseq*. Dans cette partie, la première tâche effectuée est la lecture du fichier d'entrée cons_file et de la création d'une « super variable » – ou structure au sens stricte du terme en langage C, elle sera détaillé ultérieurement – contenant l'ensemble des informations qui y sont contenues tel que les séquences, les noms des séquences, la longueur des séquences, mais également le profil au sens matrice consensus de l'alignement, etc. C'est l'appel à la fonction `ReadMulCons()` qui permet de réaliser aussi bien la déclaration, que le remplissage de cette « super variable ». Une fois le traitement

[23] L'exécution des instructions du code d'un programme se fait de façon linéaire. Pour modifier cette linéarité, on a recours à deux types de structures de contrôles principaux : les boucles lorsqu'on a affaire à une instruction qui doit être répétée, et les structures conditionnelles lorsqu'on a à faire face à plusieurs choix possibles. Lorsqu'une série d'instruction est usitée un nombre important de fois, il est parfois judicieux de les regrouper en fonctions (ou procédures).

du jeu de bloc effectué, le programme bifurque entre l'utilisation de *Caliseq* en mode normal ou l'utilisation en mode recherche sur banque. Cette bifurcation dépend là encore d'un balise, remplie ou non, selon que `getopt()` a rencontré ou non le paramètre « –B ». D'un point de vu algorithmique, le cas le plus simple est celui du premier mode, même s'il figure après le cas du mode de recherche sur banque dans le corps du programme comme cela est présenté dans l'**Algorithme** 7.1 de la fonction `Main()`.

Ainsi, pour le **premier mode**, celui d'alignement simple du jeu de blocs sur une séquence ou un pré alignement de séquences, le fichier ali_file est envoyé à la procédure `AlignToFreq()`. Celle-ci est assez similaire à la procédure `ReadMulCons()` dans la mesure où `AlignToFreq()` renvoie une autre « super variable » contenant toutes les informations nécessaires du fichier d'alignement ali_file. Les deux « super variables » (l'un provenant du traitement de cons_file et l'autre de ali_file) sont alors passé en arguments à la fonction `Calign()` (*Caliseq* dans son premier mode) qui réalise et affiche l'alignement du jeu de blocs sur la ou les séquence(s) cibles.

L'algorithme pour le **mode de recherche sur banque** est plus complexe du fait que le programme sauvegarde dans un arbre (sorte de liste hiérarchisée) en fonction du score d'alignement, l'ensemble des séquences qu'il a identifié en tant que P450s par le jeu de blocs. La fonction `Main()` appelle des fonctions présentes dans le fichier minimier.c pour mettre en place et construire cette arbre. Par ailleurs, comme un z-score est appliquée sur les scores afin de filtrer les séquences, la procédure d'alignement des blocs sur les séquences cibles est réalisée deux fois : (i) la première fois sur l'ensemble des séquences de la base de séquences afin de récupérer chaque score pour un calcul de moyenne, d'écart type, et finalement du z-score (ii) puis une seconde fois uniquement sur les séquences retenus dans l'arbre. Le calcul du z-score a été précédemment expliqué dans le manuscrit. Il est à noté que l'arbre permet la sauvegarde des séquences classées en fonction de leur score. Le nombre de séquences sauvegardé dans l'arbre dépend du nombre d'éléments (de feuilles) défini en paramètre d'entrée par l'utilisateur. Lorsque ce nombre est faible, seules les séquences présentant le meilleur score d'alignement seront sauvegardées dans l'arbre. D'un point de vu algorithmique, chaque enregistrement du fichier ali_file (banque de séquences au format fasta) est lu indépendamment successivement, affecté chaque fois à une variable. Cette variable qui contient la séquence est envoyé en argument avec la « super variable » issu du traitement du fichier cons_file à la fonction `Caliseq()`. `Caliseq()` réalise alors l'alignement et attribue un score d'alignement pour la séquence. Selon le score obtenu, la séquence et son score respectif sont enregistrés dans l'arbre. Par ailleurs, les scores et le nombre de séquences sont aussi retenus et cumulés dans d'autres variables pour pouvoir calculer la

moyenne et l'écart type, et serviront ultérieurement pour le calcul du z-score. Une fois que toutes les séquences de la banque traitées, les séquences de chaque feuille de l'arbre sont soumises à nouveau à la fonction `Caliseq()` avec la « super variable » représentant le jeu de blocs. C'est au cours de ce second passage qu'un z-score est calculé pour chaque séquence de l'arbre et comparé à un seuil défini par l'utilisateur en paramètre d'entrée (ou un seuil égale à 0 par défaut). De cette comparaison, dépendra l'affichage de l'alignement de cette séquence avec le jeu de blocs.

Fonction Main() :

```
/* Récupération des paramètres en entrée */
Pour tous les arguments récupéré en paramètre d'entrée par la fonction getopt(), faire :
/* Tous les cas ne sont pas présentés ici */
   cas 'N' :
     balise du mode Caliseq mise activée
   cas 'B' :
     variable read_bank prend la valeur du nombre maximal de feuilles pour l'arbre
   cas 'Z' :
     variable seuil pour le z-score sMinZscoreCons prend la valeur insufflé par l'utilisateur
   /* etc… */
fin pour
mémorisation du nom pour le fichier ali_file
mémorisation du nom pour le fichier cons_file
/* Partie spécifique à Caliseq, les autres parties ne sont pas présentées */
Si (balise du mode Caliseq activé) alors :
/* Lecture du fichier cons_file et enregistrement dans une super variable multicons */
   multicons récupère les informations en sortie de ReadMulCons(cons_file)
/* Lancement de Caliseq en mode de recherche sur banque selon le remplissage ou non de read_bank */
    Si (read_bank est remplie) alors :
      init_minimier(read_bank) /* Création de l'arbre au nombre de feuilles défini dans read_bank */
      Tant qu'il reste des enregistrements curseq dans le fichier ali_file, faire :
/* Il existe des fonctions de correction pour les enregistrements non « propres » qui ne sont pas présenté */
/* Premier passage par Caliseq */
          score récupère les informations en sortie de Caliseq(curseq, multicons)
          cumul de score et de son carré, du nombre de séquence dans la banque
          seq2keep est créé par sNewSeq2Keep(curseq, score)
          seq2keep est envoyé à l'arbre par range(seq2keep)
          Si (seq2keep n'est pas conservé dans l'arbre) alors le détruire
      fin tant
      définition de lim, le nombre total de séquences conservées /* soit égale à la valeur de read_bank soit cumulé précédemment */
      calcul de la moyenne mu et de l'écart type ecart à partir des scores cumulés
      Pour les lim feuilles de l'arbre, faire :
 /* Deuxième passage par Caliseq */
          récupération de l'enregistrement contenu dans chaque feuille par get_minimier_swres()
          Caliseq(curseq, multicons, mu, ecart, deuxième passage)
      fin pour
    fin si /* fin du code pour le mode de recherche sur banque */
/* Lancement de Caliseq en mode normal selon le remplissage ou non de read_bank */
    Sinon
  /* Lecture du fichier ali_file et enregistrement dans une super variable alifreq */
     alifreq récupère les informations en sortie de ReadMulCons(cons_file)
     Calign(alifreq, multicons)
    fin sinon /* fin de Caliseq en mode normal */
    fin si /* fin de l'outil Caliseq */
```

Algorithme 7.1 Algorithme simplifié de la fonction Main() de *SmartConsAlign*. Seule la partie relative à *Caliseq* est présentée. Les arguments passés aux fonctions sont inscrits dans les parenthèses des fonctions. Les variables sont indiqués en italique.

3.4.2 La fonction ReadMulCons() de mulcons.c

La fonction `ReadMulcons()` a été développée pour lire le fichier de jeu de bloc – celui là même qui comporte le symbole « # » comme séparateur – afin de les transformer en matrice consensus ou profil. Historiquement, c'est d'ailleurs la première fonction qui j'ai écrite et qui a contraint le fichier cons_file à être écrit tel qu'il est. Dans l'objectif de sauvegarder les informations de ce fichier, il m'a fallu définir une « super variable » communément appelé « structure » en langage C. Celle-ci est désignée – on parle également de type – ConsFrags2 et regroupe un grand nombre de variables et de sous structures. La décomposition de cette « super variable » est présentée dans le Tableau A 3-2. Les variables *multicons* et *alifreq* présenté dans la fonction `Main()` sont de type ConsFrags2.

Tableau A 3-2 Décomposition de la structure ConsFrags2. Les astérisques « * » correspondent en langage C à un pointeur correspondant à une adresse mémoire. On souvent recourt aux pointeurs pour créer des tableaux de façon dynamique, mais également lors des passages en arguments.

Nom de la structure	***Contenu de la structure***		***Explications relatives***
	Type de la variable	***Nom de la variable***	
ConsFrags2	RConsensus	*rcons	rcons est un pointeur vers une structure qui permet de sauvegarder la matrice consensus. L'utilisation du pointeur est justifiée lorsque qu'une fonction est de type RConsensus et retourne une structure de type RConsensus
	Sseqs	**sfrags	sfrags est un pointeur vers un tableau de structures contenant l'ensemble des séquences. L'utilisation du pointeur est justifié également pour les mêmes raisons que précédemment
	Int	nbfrags	nbfrags est le nombre d'éléments contenu dans le tableau sfrags
	Int	realsize	realsize est utilisé pour allouer de la mémoire pour le tableau de sfrags
	Double	fgap_gap	Contient le poids attribué au gap face à un gap
	Double	Fgap_op	Contient le poids attribué à l'ouverture d'un gap
	Double	*weight	weight est un tableau de poids pour pondérer lors de la construction de matrice consensus les séquences trop identiques
Rconsensus	Float	*poids	La matrice consensus est sauvegardée dans un tableau unique à une dimension
	Char	*vocab	vocab contient la séquence de l'alphabet utilisé pour construire la matrice
	Int	len	len est la longeur du tableau poids
Sseqs	Char	**seqs	seqs est un tableau de séquences
	Char	**seqnames	seqname est le tableau des noms de séquences associés
	Int	*seqrealsize	seqrealsize est un tableau des longueurs de chaque séquence
	Int	realsize	realsize est utilisé pour allouer de la mémoire pour le tableau de sfrag
	Int	nseq	nseq est le nombre de séquence contenu dans le tableau

Pour sauvegarder les informations du fichier de blocs, j'ai donc déclaré la variable *multicons* au type *ConsFraqs2* où chaque bloc est un élément de sfraqs contenant lui-même dans un tableau toutes les séquences et noms des séquences de chaque bloc. La lecture des informations de chaque bloc a nécessité un fichier temporaire pour chaque bloc et l'appel à la fonction `read_alignement()`, fonction préexistante à *SmartConsAlign* qui lit un fichier d'alignement de séquences. Un fois les alignements récupérés, la procédure `CreateWeight()` est lancé avec *multicons* en argument pour calculer les pondérations à utiliser pour chaque séquence. Cette pondération a été mise en place pour ne pas favoriser lors de la création du profil, le résidu provenant de séquences trop identiques. Enfin, `ReadMulCons()` lance la procédure `CreateCons()` qui permet d'obtenir les profils des blocs. Une fois la variable *multicons* remplie, elle est retournée à la fonction `Main()`.

Fonction ReadMulCons(cons_file)

Déclaration de *multicons*
Déclaration d'un fichier temporaire *tmp*

/* mise en mémoire des séquences */
Faire tant qu'on atteint pas la fin du fichier cons_file :
  Recopier chaque bloc dans le fichier *tmp*
  Lire le fichier *tmp* et sauvegarder dans *mulicons->sfrags* les alignements de chaque bloc
  Retenir le nombre de bloc dans *mulicons->nbfrags*
fin faire

/* création d'une table de pondération pour chaque séquence */
muticons reçoit en sortie de CreateWeightl(*multicons*) lui-même, complété de *multicons->weight*

/* création de la matrice consensus globale de tout le jeu de bloc */
muticons reçoit en sortie de CreateCons(*multicons*) lui-même, complété de *multicons->rcons*

Algorithme 7.2 Algorithme simplifié de la fonction ReadMulCons() qui permet la lecture et la création du profil pour le jeu de bloc. La fonction reçoit en argument d'entrée le fichier cons_ali (ainsi que son emplacement pour l'ouverture du fichier) et retourne une variable de type ConsFrags2 dont tous les champs ont été complétés.

3.4.3 La fonction CreateWeight() de mulcons.c

C'est l'une des dernières fonctions que j'ai écrite. À l'origine, elle a été mise en place pour modérer, assouplir le profil lorsque qu'il y a dans l'alignement de séquences, des séquences trop identiques. En effet, cette identité entre séquences se répercuterait au niveau des acides aminés où les fréquences d'apparitions à une position donnée seraient trop importantes, masquant de ce fait la présence d'un acide aminé conservé sur toutes les séquences. C'est pourquoi il est nécessaire de pondérer les séquences trop proches entre elles. Pour cela, il faut calculer les identités entre toutes les séquences et les comparer toutes ces identités entre elles, deux à deux. Le calcul de la pondération est expliqué en détail dans la section discutant de l'évolution de *Caliseq*, mais les grandes lignes sont

décrites de la façon suivante : pour chaque paire de séquence, une identité est attribuée. À partir de cette taux d'identité, il est possible de calculer pour chaque séquence, la fréquence moyenne des identités entre cette séquence et les autres séquences du jeu. Or, ce n'est pas la fréquence moyenne des identités qui est intéressante, mais son complémentaire pour à la fois défavoriser les séquences trop proches et favoriser les séquences n'ayant pas de séquences proches. `CreateWeight()` est une fonction qui n'appelle pas d'autres fonctions, mais est dotée d'un algorithme de programmation assez « fantaisistes », ne serait-ce que pour faire les identités deux à deux.

Fonction CreateWeight(*multicons* de type ConsFrags2)

```
/* Création du tableau qui contiendra les identités de chaque couple de séquence */
Déclaration d'un tableau tab_id de taille (nbseq*(nbseq-1)/2)  /* nbseq est le nombre de séquences totales dans l'alignement*/

/* Création du tableau qui contiendra les pondérations de chaque séquence. Sa dimension est donc nbseq */
Déclaration de multicons->weight de taille nbseq
Si (il n'y a qu'une seule séquence) alors /* Dans le cas d'une séquence unique, il n'est pas nécessaire de calculer les pondérations */
  multicons->weight[0]=1.0
  retourner à la fonction appelante  multicons
fin si
Initialisation à 0 de tout le tableau multicons->weight
Déclaration d'une variable idpos qui va remplir le tableau d'identité tab_id

/* Remplissage de la table d'identité */
Pour chaque séquence is allant de 0 à nbseq–1,  faire :     /* il n'est pas nécessaire de comparer la dernière séquence à elle-même */

   Pour chaque séquence js allant de is+1 à nbseq, faire :      /* la comparaison de toutes les séquences aboutit à une matrice symétrique */
      Initialisation des compteurs d'identité identity et du nombre d'acides aminés nbcomp comparé

      Pour chaque bloc, faire :   /* Parcours des blocs */

         Pour chaque position ipos dans la séquence, faire :  /* Parcours des positions */
            Incrémenter nbcomp
            Si (dans le même bloc, à la même position ipos, des séquence is et js sont identiques) alors incrémenter identity
          fin pour
       fin pour
       idpos prends la valeur identity/nbcomp  et s'incrémente /* fréquence d'identité entre deux séquences sauvegardé dans tab_id */
   fin pour
fin pour

/* Remplissage de la table des poids multicons-> weight par les fréquences d'identité cumulées */
Pour chaque case de multicons-> weight, faire :
   cumuler les fréquences associées à chaque séquence
fin pour   /* l'algorithme est simplifié ici, car en réalité, il faut parcourir tab_id par « ligne » et par « colonne » */

/* Calculer les fréquences moyennes à partir des fréquences cumuler en enregistrer son complémentaire */
Pour chaque élément i de multicons-> weight, faire :
  multicons-> weight[i] reçoit 1– (multicons-> weight[i]/(nbseq-1))
fin pour

retourner à la fonction appelante la variable multicons
```

Algorithme 7.3 Algorithme (très) simplifié de la procédure CreateWeight() qui permet la mise en place d'une pondération pour les séquences trop proches dans l'alignement. Cette fonction reçoit une variable de type ConsFrags2 et retourne cette même variable à la fonction appelante, qui peut être soit ReadMulCons() soit Alifreq().

3.4.4 La fonction CreateCons() de mulcons.c

`CreateCons()` est sans nulle doute la fonction la plus importante du programme *Caliseq* : elle réalise à elle seule la création d'une grande partie de la « matrice » de substitution (artéfact de la programmation dynamique), comme il sera vu plus tard. Cette fonction n'a cessé d'être remaniée et réorganisée, selon l'évolution subite pour le calcul du profil des blocs. Je présenterai ici la version finale en date de cette fonction, mais les différentes étapes de l'évolution de cette fonction ont été discutées dans le manuscrit. Cette fonction permet donc la création d'un profil à partir de l'ensemble des informations déjà disponibles dans la variable de type ConsFrags2 (séquences et pondérations) et sera sauvegardé dans la structure *rcons* de cette même variable. À noter que cette fonction est commune à la création du profil pour le jeu de blocs (provenant du fichier cons_file), mais aussi pour créer un profil à partir des séquences pré alignées sur la séquence cible (provenant du fichier ali_file). Il existe néanmoins quelques petites différences au niveau de l'algorithme qui seront vues ci-après. Pour construire le profil, il faut déclarer une matrice, dont les dimensions ont pour largeur (ou pour nombre de colonnes) le nombre totale de résidus d'une même séquence sur l'ensemble des blocs et pour hauteur, l'alphabet (ou l'ensemble des symboles) utilisé pour représenter la séquence. Pour mémoire, le profil présenté pour *SmartConsAlign* comportait un alphabet de 20 lettres, chaque lettre représentant un acide aminé différent. Ici, un alphabet plus conséquent a été utilisé, de 32 symboles, dont certains caractères ont été sollicités pour mémoriser d'autres informations que celle de fréquence des acides aminés. Ainsi, une des particularités de l'outil *Caliseq* a été d'incorporer directement la matrice d'insertion pour la programmation dynamique à la matrice de substitution, grâce à l'existence de ces caractères supplémentaires. Pour ce qui est de la largeur du profil, il suffisait par exemple de compter le nombre total de résidu de la première séquence pour tous les blocs. La déclaration et la création de cette matrice consensus se fait au moyen de la procédure `NewRConsenus()` du fichier cons_align.c. Une autre matrice, de similarité ente résidus, est nécessaire. Elle correspondait initialement à une matrice de similarité (de type BLOSUM ou PAM), mais finalement est devenue dans la dernière version de `CreateCons()` une simple matrice unitaire d'identité, de dimension *alphabet x alphabet*. Elle est donc initialisé à 1 pour la diagonale (lorsqu'il y a identité entre les résidus se trouvant en face) et 0 ailleurs. Cette deuxième matrice de similarité (ou de fréquence de replacement d'un caractère par un autre) est réalisée par l'appel à la fonction `unitary_matrix()` du fichier Nmatrix.c. Ces deux fonctions ne nécessitent pas de description : elles se contentent d'allouer de la mémoire pour la création d'une matrice (contenu dans un tableau à une dimension, c'est sa seule particularité, lié à son utilisation en langage C) et d'initialiser cette matrice par des 0 sauf au niveau des diagonales. Par ailleurs, ces fonctions ont été adaptées de fonctions préexistantes à

SmartConsAlign. Le remplissage de la matrice consensus s'effectue ensuite par calcul d'une simple la fréquence d'un symbole donné à une position donnée de l'alignement en séquence, et ce, pour chaque position de l'alignement. La fréquence peut être de plus réajustée en fonction de la fréquence de similarité tirée de la seconde matrice, ainsi que de la pondération calculée pour chaque séquence (cf. précédemment). Le calcul de la fréquence pour le profil des alignements issus du fichier cons_file (qu'on peut aussi nommer « PSSMs séquentielles ») est un peu plus complexe et se voit par ailleurs ajusté par un facteur supplémentaire : celui des fréquences dites « background » de chaque acide aminé, à savoir leurs fréquences générales dans toutes les protéines identifiées à ce jour. En fait, comme il sera expliqué plus tard dans le manuscrit, dans la dernière version de `CreateCons()`, le profil généré pour les blocs est construit de la même manière qu'une matrice de similarité BLOSUM. En effet, pour obtenir les poids de substitution dans une matrice BLOSUM, on calcule dans un alignement de séquence, le logarithme en base 2 des fréquences de chaque couple d'acide aminé à une position donnée, divisé par le produit des fréquences « background » de chaque acide aminé du couple. Dans `CreateCons()`, le même principe est appliqué pour l'obtention des matrices consensus. Les fréquences « background » (désignées aussi comme fréquences de fond) des acides aminés permettent entre autre de calculer des pseudo-comptes, et évitent ainsi d'avoir à soumettre à un logarithme des fréquences nulles pour certains acides aminés. Ces fréquences « background » ont été récupérées à partir des statistiques sur les acides aminés de la base de séquences Swiss-Prot. Celles-ci sont stockée provisoirement en « dur »[24] dans le corps du programme, à l'intérieur d'une fonction nommée `Getfreq()` de mulcons.c. Le fonctionnement de cette dernière est simple et ne nécessite pas une explication poussée : en recevant en argument d'entrée un symbole correspondant à un acide aminé (code à une lettre), la fonction renvoie la fréquence de l'acide aminé correspondant. Il est à noter que lors du calcul du profil, la fréquence des gaps dans l'alignement est déterminée de la même manière. Toutefois, la fréquence « background » pour les gaps (appelés aussi INDEL pour insertion/délétion) présente la particularité d'avoir été calculé sur la banque d'alignements multiples de séquences BAliBASE (http://www.igbmc.ustrasbg.fr/BioInfo/BAliBASE2/index.html) (Thompson et al, 1999). Un script python a été écrit à cet effet pour comptabiliser le nombre de gap présents ainsi que le nombre de caractères total (gap inclus) dans toute la base. Cette fréquence a été elle aussi enregistré en « dur » dans le corps du programme dans la même fonction `Getfreq()` de mulcons.c. Il est à noter que lors du compte, les gaps de type INDEL ont été différenciés des gaps alignés sur d'autres gaps.

[24] « Ecriture en dur » exprime en informatique que les données ne peuvent être modifiées qu'à l'intérieur du code. Ce n'est pas usuellement conseillé de procéder ainsi : la plupart du temps, il est préférable de lire des données en paramètre d'entrée au programme. Dans notre cas, si les valeurs de fréquences des acides aminés venaient à changer, il faudrait les changer dans le corps du programme et le recompiler.

Comme expliqué précédemment, la particularité de *Caliseq* a été d'avoir incorporé la matrice d'insertion dans la matrice consensus à l'aide des symboles supplémentaires. De plus, les deux matrices sont construites en même temps. En réalité, cette « matrice d'insertion » n'a finalement de matrice que le nom : en pratique, c'est une valeur unique qui correspond à la fréquence de gaps rencontrés à une position donnée de l'alignement des séquences dans les blocs.

Fonction CreateCons(*mutlicons* de type Consfrags2, et une *balise*)

```
/* CreateCons est appelé pour faire le profil des blocs du fichier cons_file mais aussi de l'alignement de séquence du fichier ali_file… */
/* … il lui faut donc une balise pour différencier le traitement des deux                                                              */
Pour tous les nbfrags blocs, faire :
  totSize : = Cumuler la longueur de la première séquence de l'alignement, sur tous les blocs
/* l'importance que toutes les séquences à l'intérieur d'un même bloc doivent être de même longueur vient de ce calcul */
fin pour
la fonction NewRConsenus(alphabet, totSize) permet de créer la matrice de fréquence (en 1D) dans la variable multicons->rcons
/* NewRConsenus crée une matrice de hauteur alphabet et de largeur totSize */
la fonction unit_matrix()  permet de créer et d'initialiser à 0 et 1 (sur la diagonale) une matrice de similarité
/* sa dimension est de alphabet x alphabet et correspond à une matrice de fréquence de remplacement d'un résidu par un autre */
Pour tous les nbfrags blocs, faire : /* Parcours de tous les blocs */
  Pour toutes les positions ipos d'alignement en séquences, faire : /* Parcours des positions sur alignement */
    Pour tous les symboles isymb de l'alphabet, faire :  /* Parcours des symboles*/
      Initialiser le compter sum à 0.00
       Pour toutes les séquences i de l'alignement, faire :
         Récupérer l'index de correspondance entre l'alphabet et le résidu de la séquence i à la position ipos
         Si (index est positif et que l'isymb est un caractère d'acide aminé) alors
            sum s'incrémente par le produit de la fréquence de remplacer isymb par index, par la pondération pour la séquence i
         fin si
         Si (index correspond au caractère gap et isymb au caractère utilisé pour retenir la fréquence des gap) alors
            sum s'incrémente
         fin si
         Si (index et isymb ne correspond à aucun résidu, ni gap) alors /* Parmi les 32 symboles, certains ne servent pas */
            sum ne reçoit rien
         fin si
      fin pour
      Si (la balise est activée ou isymb correspond à un symbole non utilisée) alors
          sum reçoit sa division par le nseq        /* calcul de la fréquence normale */
      fin si
      Sinon     /* Calcul du poids comme pour BLOSUM */
          sum reçoit le calcul du log en faisant appel à la fonction Getfreq(isymb) pour obtenir les fréquences « background »
      fin sinon
      Si (sum > la variable logmaxval) alors logmaxval := sum   /* logmaxval et logmin val permettent de calculer la fréquence  */
      Si (sum < la variable logminval) alors logminval := sum    /* de poids de gap pour la construction de la matrice de délétion */
      sum est mémorisé dans rcons qui passe à l'élément suivant
    fin pour
  fin pour
fin pour
muticons->fgap_gap := (logmaxval) – gap_gap_ratio * (logmaxval – logminval) /* gap_gap_ratio est donnée en paramètre …*/
muticons->fgap_op := (logmaxval) – gap_op_ratio * (logmaxval – logminval)      /* …d'entrée à SmartConsAlign par l'utilisateur */
Retourner à la fonction appelante multicons
```

Algorithme 7.4 Algorithme simplifié de la fonction CreateCons()qui est appelé soit par ReadMulCons() soit par AlignToFreq() passant tous deux à CreateCons() une structure de type ConsFrags2. CreateCons() renvoie la structure de type ConsFrags2 en ayant rempli la matrice consensus rcons ainsi que le poids de gap fgap_gap. A noter que j'ai utilisé un symbole particulier ici « := » qui signifie affectation.

3.4.5 La fonction AlignToFreq() de mulcons.c

La fonction `AlignToFreq()` réalise le même travail que `ReadMulCons()`, mais avec le fichier ali_file, à savoir la création d'un profil pour le fichier d'alignement de séquences cibles. Cette

procédure n'est appelée que en mode normale de *Caliseq*, dans la mesure où elle peut lire un alignement de séquence (alors que dans le mode de recherche de *Caliseq*, les séquences soumises sont traitées une à une). La variable utilisé pour retenir les séquences en mémoire est de type ConsFrags2, du même type que la variable *multicons* utilisé dans `ReadMulCons()` : elle s'appelle cette fois ci *alifreq*. L'architecture de la fonction `AlignToFreq()` est elle aussi vraiment similaire à celle de `ReadMulCons()`, excepté le fait qu'il n'y a plus besoin de rechercher des « # » qui délimitent les blocs, ni même de créer un fichier temporaire pour la lecture de ceux-ci : l'intégralité du fichier est lu par la fonction `read_alignement()`. `CreateWeight()` et `CreateCons()` sont ensuite appelés pour construire la matrice consensus, qui s'apparente vraiment, dans le cas d'*alifreq*, à une matrice de fréquence.

Fonction AlignToFreq(ali_file)

Déclaration de *alifreq*

/* mise en mémoire des séquences */
Lire le fichier ali_file et sauvegarder dans *alifreq->sfrags* les alignements de tout le fichier

/* création d'une table de pondération pour chaque séquence */
alifreq reçoit en sortie de CreateWeightl(*alifreq*) lui-même, complété de *alifreq->weight*

/* création de la matrice consensus globale de tout le jeu de bloc */
alifreq reçoit en sortie de CreateCons(*alifreq*) lui-même, complété de *multicons->rcons*

Algorithme 7.5 Algorithme simplifié de la fonction AlignToFreq() qui permet la lecture et la création du profil pour la séquence cible ou son pré-alignement. La fonction reçoit en argument d'entrée le fichier ali_ali (ainsi que son emplacement pour l'ouverture du fichier) et retourne une variable de type ConsFrags2 dont tous les champs ont été complétés.

3.4.6 La fonction CreateWdel() de mulcons.c

La fonction `CreateWdel()` permet de construire la troisième et dernière matrice pour la programmation dynamique (un du moins, une partie), celle des poids de délétion. En effet, la matrice de substitution est en partie déjà créé par la fonction `CreateCons()` qui crée également la matrice d'insertion. En quelque sorte, les « PSSMs séquentielles » issues des blocs correspondent à la première moitié de la matrice de substitution et il suffit de la multiplier avec la matrice consensus (ou fréquence) issu de l'alignement de la séquence cible avec ses proches, pour obtenir la matrice finale de substitution. Il en est de même pour la matrice de délétion.

Caliseq ne gère que les poids d'ouverture de gap (gap_op) et les poids de gap face à un gap (gap_gap). Le poids d'ouverture de gap peut être contrôlé en entrée du programme à partir de l'option « –g » suivi d'un coefficient pour atténuer au diminuer la valeur du gap. Le calcul et l'attribution du

poids des deux type de gaps s'effectuent quant à eux dans la procédure `CreateCons()` précédemment décrite. Leurs valeurs sont sauvegardées dans la variable de type ConsFrags2 issue des blocs. C'est d'ailleurs cette même variable qui est passée à la fonction `CreateWdel()` pour construire la matrice de délétion. Cette matrice est particulière dans l'outil *Caliseq*, car il fallait trouver un moyen pour ne pas pénaliser l'apparition de gap entre les blocs, permettant ainsi aux blocs de « coulisser » sur la séquence cible (ou le pré-alignement) sans contrainte. Afin de remplir cette condition, l'idée a été de ne pas pénaliser les l'apparition de gap en fin de chaque bloc. Ainsi, la matrice de délétion peut être schématisée comme sur la Figure A 3-6, où en fin de chaque bloc, l'apparition d'un gap entre les blocs n'entraîne aucune pénalité.

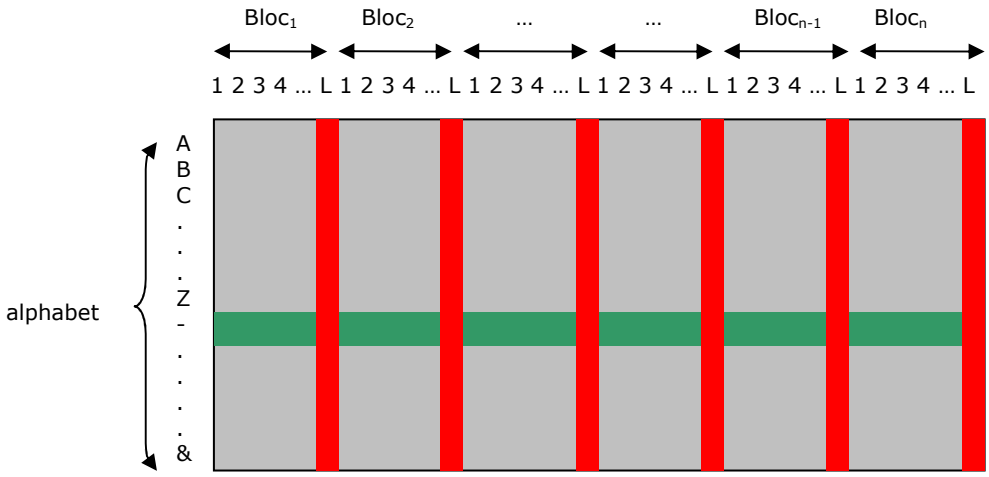


Figure A 3-6 Schéma de la matrice de délétion. La dimension de la matrice correspond en hauteur à l'alphabet utilisé (nombre de symbole) et en largeur au nombre total de résidus d'une séquence à travers tous les blocs. Ici chaque bloc est représenté par des positions (1, 2, 3, 4 …, L) où L correspond à la dernière position du bloc. En gris correspond aux parties de la matrice dont le poids est gap_op, en vert, gap_gap et en rouge, 0. Ainsi, à la fin de chaque bloc, l'apparition de gap entre les blocs n'est pas pénalisée (=0)

Cette fonction est appelée juste avant l'algorithme de programmation dynamique, soit par la fonction `Calign()` soit par la fonction `Caliseq()`. Tout comme la matrice consensus, elle a été programmée sur une matrice à 1 dimension, et est de type RConsenus (cf.Tableau A 3-2).

Fonction CreateWdel(*multicons* de type ConsFrags2)

Définition de la matrice de délétion *Wdel*
Récupération des valeurs de *gap_op* et *gap_gap* de *multicons*

/* remplissage de toute la matrice par *gap_op* */
Pour tous les éléments *i* de la matrice *Wdel*, **faire :**
  Wdel->poids[i] := *gap_op*
fin pour

/* remplissage de la ligne correspondant au symbole « – » de la matrice par *gap_gap* */
Pour tous les éléments i de la matrice correspondant au symbole « – », **faire :**
  Wdel->poids[i] := *gap_gap*
fin pour

/* remplissage de la dernière position de chaque bloc par 0 */
Pour chaque bloc, **faire :**
  Pour chaque symbole, **faire :**
    Si (on est sur la position *i* finale d'un bloc**) alors** *Wdel->poids[i]* := 0.00
  fin pour
fin pour
/* En réalité, dans le programme, j'utilise un pointeur vers l'adresse de *Wdel->poids* que je remplie au fur et à mesure */

retourner à la fonction appelante *Wdel*

Algorithme 7.6 Algorithme simplifié de la fonction CreateWdel() qui prend en entrée la variable de type ConsFrags2 provenant du fichier cons_file pour créer une matrice de délétion. Elle est appelée par Calign() ou Caliseq().

3.4.7 Les fonctions Calign() et Caliseq() de mulcons.c

`Calign()` ainsi que `Caliseq()` sont les deux fonctions qui réalisent la recherche de l'alignement optimale entre les profils des blocs et le profil de la séquence cible par un algorithme de programmation dynamique de type NWS (cf. section 2.4.1.3 page 85). Dans le cas de **`Calign()`** qui permet le lancement de l'outil *Caliseq* dans son mode normal, la réalisation de l'alignement profil-profil passe par l'établissement d'une matrice d'alignement *M[i,j]* où chaque élément de la matrice donne le score de similarité obtenu pour un sous-alignement optimal des profils séquentielles ***A*** (issu du fichier cons_file), de la position 1 à la position *i*, avec les séquences cibles (soit simple soit pré-alignées), ***B***, de la position 1 à la position *j*. Cette matrice *M[i,j]* est alors construite à partir de trois scores de similarité de sous-alignements optimaux *M[i–1,j–1]*, *M[i,j–1]*, *M[i–1,j]*. En effet, *M[i,j]* est le score maximal entre les trois scores suivants :

- *M[i–1,j–1] + W(j,i)*
- *M[i–1,j] + i*(a_i)
- *M[i,j–1]+ d*(b_j)

où *W(j,i)* est le score de similarité entre la position *i* des « PSSMs séquentielles » et la position *j* du set de séquences pré-alignées, *i*(a_i) est le poids de gap dans B contre la position *i* sur ***A*** (insertion de la

position *i* des « PSSMs séquentielles ») et enfin $d(b_j)$ est le poids de gap dans ***A*** contre la position *j* sur ***B*** (insertion d'une colonne *j* des séquences pré-alignées). Le score de similarité *W(j,i)* pour une position d'alignement *i* dans les « PSSMs séquentielles » avec la position *j* s'obtient alors par le produit du poids à la position *i* des « PSSMs séquentielles » avec le poids à la position *j* du profil pour les séquences pré-alignées. C'est pour cette raison que je comparais les « PSSMs séquentielles » à une « demi » matrice de substitution. Le score de délétion s'obtient de la même façon en multipliant le poids à la position *i* de la matrice de délétion par le poids de la position *j* de la matrice consensus pour les séquences pré-alignées. Le score d'insertion est quant à lui un peu différent, et s'obtient par addition de la fréquence de gap multiplié par le poids d'un gap contre un gap à la position *i* avec son complémentaire (1– fréquence de gap à la position *i*) multiplié par le poids d'ouverture de gap. Ainsi, on comprend mieux l'utilité d'avoir préparé les données nécessaires au calcul des trois matrices : i) la matrice de substitution, contenue dans la variable de type ConsFrags2 a été créée par la fonction `CreateCons()`, ii) la matrice d'insertion contenue dans la même variable et a été créée également par la fonction `CreateCons()` (mais n'est contenu que dans la variable contenant les « PSSMs séquentielles » et enfin, iii) la matrice de délétion qui est obtenue par appel à la fonction `CreateWdel()`. Dans le cas de la fonction **`Caliseq()`**, l'algorithme présenté ci-dessus est légèrement simplifié, puisqu'il ne s'agit plus d'un alignement profil-profil, mais d'un alignement entre un profil et une séquence simple. En effet, le fichier d'alignement de séquence ali_file, n'est plus lu dans sa totalité cette fois-ci, mais chaque enregistrement est lu l'un après l'autre au fur et à mesure du traitement. Cependant, pour pouvoir adapter l'algorithme précédent au traitement d'une séquence au lieu d'un profil il a fallu indexer la séquence cible à l'aide de la fonction `IndexSequence()` et de la désindexer une fois l'alignement réaliser par la fonction `DeIndexSequence()`. Cette indexation correspond en fait à établir une correspondance de chaque position de la séquence avec l'alphabet utilisé. Les deux fonctions sont préexistantes au programme *SmartConsAlign*, et ne seront pas détaillées.

En pratique, que ce soit pour `Calign()` ou `Caliseq()`, deux matrices d'alignements sont utilisés : une contenant les meilleurs scores selon les trois chemins possibles pour chaque élément de la matrice, et une autre matrice contenant justement les chemins empruntés. Ces deux matrices sont remplies en simultanées lors de la programmation dynamique. Une fois le remplissage des deux matrices effectuée, une fonction `Backtracking()` est appelé pour « remonter » le chemin dans la matrice des chemins empruntés, correspondant à l'alignement optimale. Cette fonction retourne dans une variable de type caractère, le chemin « optimal ». Dans la procédure `Calign()`, les deux profils et le chemin optimal sont envoyés à la fonction `PrintAln()` qui réalise l'affichage de l'alignement des

séquences contenues dans chaque bloc sur la séquence cible (ou son pré-alignement). Dans la fonction `Caliseq()` en revanche, des opérations supplémentaires sont présentes, comme la vérification de la longueur d'alignement avec le cut off sur la longueur, défini en paramètre d'entrée (option « –L ») effectuée par `LegnthSeqVSCutOff()` ou encore, le calcul du z-score qui selon la comparaison avec le seuil donnée en paramètre d'entrée (option « –Z ») permettra ou non l'affichage de l'alignement par la fonction `PrtAlnBank()` similaire à `PrintAln()` mais simplifiée.

Fonction Calign(*multicons* et *alifreq* de type ConsFrags2)

```
/* Création des 2 matrices d'alignement */
Récupération des longueurs des profils lpat et lseq
Création des deux matrices de dimension lpat x lseq /* Pathmatrix P et Scorematrix S */
Wdel := Création de la matrice de délétion par CreateWdel(multicons)
/* Algorithme de programmation dynamique */
Pour chaque position i du profil de multicons->rcons, faire :
   Pour chaque position j du profil de alifreq->rcons, faire :
      Si (on n'est pas sur la première colonne ni la première ligne) alors :
          Initialiser la variable poids de substitution wsub=0 /* cette variable permet de mémoriser les substitutions pour tout l'alphabet */
          di := ajouter à la valeur de S[i–1,j] le poids d'insertion à la position i du profil multicons->rcons
          dd := valeur de S[i–1,j]
          Pour tous les symboles de l'alphabet, faire :
              Wusb := incrémenter par le produit du poids de la position i de multicons->rcons par le poids de la position j de alifreq->rcons
              dd := incrémenter par le produit du poids de délétion Wdel[i] par le poids de la position j de alifreq->rcons
          fin pour
          ds := incrémentation par ajout de la valeur S[i–1,j–1] par Wsub
          Si (ds > dd et ds> di) alors /*cas de la substitution */
              S[i,j] := ds
              P[i,j] := 'SUB' /* on inscrit le chemin par SUB si on vient de la diagonale, INS de la gauche et DEL de dessus*/
          fin si
          Si (dd > ds et dd > di) alors /* cas de la délétion */
              S[i,j] := dd
              P[i,j] := 'DEL'
          fin si
          Si (di > ds et di > dd) alors /* cas de l'insertion */
              S[i,j] := di
              P[i,j] := 'INS'
          fin si
       fin si
       Si (on est sur la première ligne de la matrice d'alignement) alors
              S[i,j] := S[i–1,j]+ poids de gap
              P[i,j] := 'INS'
       fin si
       Si (on est sur la première colonne de la matrice d'alignement) alors
              S[i,j] := S[i,j–1]+ 0
              P[i,j] := 'DEL'
       fin si
       Si (on est sur la première case de la matrice d'alignement) alors
              S[i,j] :=  0
              P[i,j] := 'ROOT'
       fin si
   fin pour
fin pour
dmax := score de la dernière case de S
bestpath := récupère le chemin optimale par la fonction backtraking(P,lseq, lpat)
PrintAln(bestpath, multicons, alifreq, autres variables à afficher) permet d'afficher le résultat
Retourner dmax à l'application appelante
```

Algorithme 7.7 L'algorithme simplifié de Calign() est présenté ici. Il pend en arguments d'entrée les deux variables de types ConsFrags2 *multicons* issu du fichier cons_file et *alifreq* issu du fichier ali_file. En pratique, les matrices Pathmatrix et Scorematrix sont des tableaux à une dimension où les lignes de la matrice sont mises bout à bout.

Fonction Caliseq(*multicons* de type ConsFrags2et *cureseq* de type 'sequence', *balise*, *ecart*, *mu*)

```
/* Caliseq() reçoit plus d'arguments que Calign() dont curseq à la place de alifreq, balise pour savoir si on est ou non à la première lecture,
ecart et mu pour calculer le z-score*/
/* Création des 2 matrices d'alignement */
Récupération des longueurs du profil lpat et de la séquence lseq
Indexation de la séquence curseq par IndexSequence(curseq)
Création des deux matrices de dimension lpat x lseq /* Pathmatrix P et Scorematrix S */
Wdel := Création de la matrice de délétion par CreateWdel(multicons)

/* Algorithme de programmation dynamique */
Pour chaque position i du profil de multicons->rcons, faire :
   Pour chaque position j de la séquence de curseq, faire :
      Si (on n'est pas sur la première colonne ni la première ligne) alors :
          di := ajouter à la valeur de S[i–1,j] le poids d'insertion à la position i du profil multicons->rcons
          ds := ajouter à la valeur de  S[i–1,j–1] le poids de substitution multicons->rcons->poids[j]
          dd := ajouter à la valeur de  S[i–1,j] le poids de délétion Wdel[j]
          Si (ds > dd et ds> di) alors /*cas de la substitution */
                S[i,j] := ds
                P[i,j] := 'SUB' /* on inscrit le chemin par SUB si on vient de la diagonale, INS de la gauche et DEL de dessus*/
          fin si
          Si (dd > ds et dd > di) alors /* cas de la délétion */
                S[i,j] := dd
                P[i,j] := 'DEL'
          fin si
          Si (di > ds et di > dd) alors /* cas de l'insertion */
                S[i,j] := di
                P[i,j] := 'INS'
          fin si
       fin si
       Si (on est sur la première ligne de la matrice d'alignement) alors
                S[i,j] := S[i–1,j]+ poids de gap
                P[i,j] := 'INS'
       fin si
       Si (on est sur la première colonne de la matrice d'alignement) alors
                S[i,j] := S[i,j–1]+ 0
                P[i,j] := 'DEL'
       fin si
       Si (on est sur la première case de la matrice d'alignement) alors
                S[i,j] :=  0
                P[i,j] := 'ROOT'
       fin si
   fin pour
fin pour
dmax := score de la dernière case de S
bestpath := récupère le chemin optimale par la fonction backtraking(P,lseq, lpat)
curseq est désindexé par DeIndexSequence(curseq)

/* Si l'utilisateur à déterminer un seuil de longueur maximale entre le premier bloc et le dernier, alors la condition est vérifiée */
Si (LenthSeqCutOff(bestpath) est vrai) alors retourner dmax à la fonction appelante /* première sortie de la fonction */
/* Lorsqu'on lance Caliseq() pour la seconde fois */
Si (balise est activée) alors
    Calcul du z-score
    Si (z-score > seuil défini par l'utilisateur) alors
          PrtAlnBank(bestpath, multicons, alifreq, autres variables à afficher) permet d'afficher le résultat
    fin si
fin si
Retourner dmax à l'application appelante
```

Algorithme 7.8 L'algorithme simplifié de Caliseq() est présenté ici. Il pends en arguments d'entrée la variables *multicons* issu du fichier cons_file de type ConsFrags2 et la séquence *curseq* qui est un enregistrement du fichier ali_file. La construction des matrices d'alignement est identique à celle de Calign()

3.4.8 Les fonctions PrintAln() et PrtAlnBank() de mulcons.c

Dans le déroulement du programme, les fonctions `PrintAlin()` et `PrtAlnBank()` sont les dernières sollicitées. Elles réalisent toutes les deux l'affichage de l'alignement optimale à partir des deux « super variables » de type ConsFrags2, contenant les séquences à la fois pour le jeu de blocs mais aussi de la séquence cible ou pré-alignée avec ses proches, et à partir de la variable contenant le chemin optimal obtenu de la programmation dynamique. En réalité, ces deux fonctions n'affichent pas les résultats en sortie standard du programme : elles appellent des procédures prédéfinies qui réalisent cette tâche, comme `PrintFastaAli()` ou `PrintClustalAli()` du fichier readmultali.c. En effet, `PrintAlin()` et `PrtAlnBank()` sont en fait des procédures qui vont déclarer et créer un alignement virtuel en machine, et sauvegarder cet alignement dans une variable de structure sSeqs (cf. Tableau A 3-2). Pour ce faire, le nombre total de séquences présentes dans les deux « super variables » est compté, puis utiliser pour déclarer le nombre total de séquence dans la nouvelle structure sSeqs. Sachant ensuite que le chemin optimal contient les informations pour chaque position de l'alignement optimal, la longueur des séquences de l'alignement finale est ainsi déterminée et correspond à celle du chemin optimal.

Une fois déclarée, la variable recueillant l'alignement finale peut alors être complétée : à chaque position de la variable contenant le chemin optimal contient une directive pour une position de l'alignement globale. Ces directives sont de trois types : soit i (pour insertion) soit d (pour délétion) et enfin, soit s (pour substitution). La complétion de l'alignement globale dépend de ces trois directives :

- Lorsqu'il s'agit de la directive 'DEL' (ou 'd'), des gaps '–' sont placés dans l'alignement global final à la place des résidus des séquences dans les blocs, tandis que les résidus d'une même colonne de l'alignement en séquences sont recopiés dans l'alignement global finale ;
- Lorsqu'il s'agit de la directive 'INS' (ou 'i'), des gaps '–' sont placés dans l'alignement global final) la place des résidus de l'alignement en séquences, tandis que les résidus d'une même colonne de l'alignement en séquences dans les blocs sont recopiés dans l'alignement global finale ;
- Et enfin, lorsqu'il s'agit de la directive 'SUB' (ou 's'), les résidus d'une même colonne à la fois pour l'alignement de séquences dans les blocs et pour l'alignement de séquences cibles, sont recopiés dans l'alignement global finale.

A noter que la fonction `PrtAlnBank()` est une copie simplifiée de son homologue `PrintAln()` qui n'a qu'à afficher les séquences des alignements contenus dans les blocs et une séquence supplémentaire. Il y a donc moins de traitement (moins de structure de contrôle de type boucle) à faire. Je ne présenterai donc pas son algorithme.

Fonction PrintAln(*multicons* et *alifreq* de type ConsFrags2, *bestpath* et autres variables à afficher)

Récupération de la longueur *pathlen* de *bestpath*
/* Création de la variable qui contiendra l'alignement global final */
Récupération du nombre de séquences dans les blocs *nbmul*
Récupération du nombre de séquences dans l'alignement cible *nbali*
Création de la variable *sseqs* qui contiendra l'alignement global final contenant le nombre *nbmul+nbali* séquences de longueur *pathlen*
Pour tous les champs *seqnames* de *sseqs->seqnames*, **faire** /* Cette boucle a été simplifié en ici, en traitant en simultané les deux cas */
  Recopier les noms correspondant dans multicons->sfrags->seqnames et alifreq->sfrags->seqnames
fin pour

/*remplissage de l'alignement global final */
Pour toutes les positions *i* jusqu'à *pathlen*, **faire**
  Si (*bestpath[i]* est égale au caractère 's'**) alors**
    Pour toutes les séquences *id_mul* du bloc *id_frag* à la position *pmul*, **faire :**
      Recopier sur *Ssseqs-> seqs[id_mul][i]* le résidu correspondant à *multicons->sfrags[id_frag]->seqs[id_mul][pmul]*
    fin pour
    Pour toutes les séquences *id_ali* à la position *pali*, **faire :**
      Recopier sur *Ssseqs-> seqs[nbmul+id_ali][i]* le résidu correspondant *alifreq->sfrags[0]->seqs[id_ali][pali]*
      /**nbmul+id_ali* pour ne pas écraser la partie allouée aux alignements dans les blocs*/
    fin pour
    incrémenter *pmul* et *pali*
    Si (*pmul* atteint la dernière position dans un bloc**) alors**
      incrémenter *id_frag*
      remettre *pmul* à zero /*début d'un autre bloc*/
    fin si
  fin si
  Si (*bestpath[i]* est égale au caractère 'd'**) alors**
    Pour toutes les *id_mul* séquences de l'alignement global final jusqu'à la séquence *nbmul*, **faire :**
      Ssseqs-> seqs[id_mul][i] := '–'
    fin pour
    Pour toutes les séquences *id_ali* à la position *pali*, **faire :**
      Recopier sur *Ssseqs-> seqs[nbmul+id_ali][i]* le résidu correspondant *alifreq->sfrags[0]->seqs[id_ali][pali]*
    fin pour
    incrémenter *pali*
  fin si
  Si (*bestpath[i]* est égale au caractère 'i'**) alors**
    Pour toutes les séquences *id_mul* du bloc *id_frag* à la position *pmul*, **faire :**
      Recopier sur *Ssseqs-> seqs[id_mul][i]* le résidu correspondant à *multicons->sfrags[id_frag]->seqs[id_mul][pmul]*
    fin pour
    Pour toutes les *id_ali* séquences de m'alignement global final jusqu'à la séquence *nbali*, **faire :**
      Ssseqs-> seqs[nbmul+id_ali][i] := '–'
    fin pour
    incrémenter *pmul*
    Si *pmul* atteint la dernière position dans un bloc, **alors**
      incrémenter *id_frag*
      remettre *pmul* à zero /*début d'un autre bloc*/
    fin si
  fin si
fin pour
selon le format choisi par l'utilisateur
  PrintClustalAli(*sseqs*, autres variables à afficher)
  Ou par défaut PrintFastaAli(*sseqs*, autres variables à afficher)
fin selon
retourner la valeur 1 à la fonction appelante /* informe la fonction précédente que tout s'est bien déroulé */

Algorithme 7.9 Algorithme simplifié de PrintAln() qui prends le chemin optimal sous forme de chaîne de caractères *bestpath*, ainsi que les deux « super variables » de type ConsFraqs2, *multicons* et *alifreq*. La procédure

crée alors un alignement globale finale et envoie cet alignement finale en sortie standard par l'appel à deux fonctions PrintClustalAli() ou PrintFastaAli( )

3.4.9 Les autres fonctions utilisées par l'outil Caliseq

Les fonctions présentées précédemment réalisent les opérations principales de *Caliseq*, telles que les lectures des fichiers spécifiques pour *Caliseq*, les créations de matrices et l'alignement des blocs sur la séquence cible. L'outil *Caliseq* nécessite toutefois l'appel à d'autres fonctions dont l'écriture a été nécessaire pour simplifier ou clarifier le code (comme les fonctions pour déclarer des tableaux de façon dynamique en langage C) mais aussi des fonctions préexistantes à *SmartConsAlign* ou à d'autre programme, suffisamment modulaires pour être réutilisées sans avoir à modifier leur code source. Il s'agit pour la plupart des fonctions permettant la lecture des fichiers au format fasta et la sauvegarde séquences qui y sont présentes (comme `NewFastaSequence()`, `ReadFastaSequence()` ou encore `read_alignement()`) ou de leur écriture au format fasta ou clustal (`PrintFastaAli()` ou `PrintClustalAli()`). On retrouve également des fonctions destinées de créer un arbre, `Init_minimier()`, créer les éléments de cet arbre, `sNewSeq2Keep()`, le remplir, `range()`, et enfin récupérer les éléments par ordre hiérarchique, `get_minimier_swres()`. Ces quatre fonctions sont d'ailleurs assez communes à la création d'arbre en générale. En fait la particularité principale de cet arbre est de mémoriser des séquences en fonction d'un score. Pour *Caliseq*, c'était bien entendu le score d'alignement qui allait me servir classer les séquences. Le nombre de feuille de l'arbre est quant à lui déterminé par l'utilisateur en paramètre d'entrée. On peut en outre constater dans l'algorithme de la procédure `Main()`, qu'à la fin du premier tour –première fois qu'on utilise la fonction `Caliseq()`–, toutes les feuilles sont « remplies ». C'est seulement à la second utilisation de `Caliseq()`, que les séquences identifiés comme P450s sont isolées et affiché selon le z-score qui est calculé en fonction de leur score d'alignement respectif. Comme il sera vu plus tard, la création de l'arbre était importante pour éviter de relire deux fois la banque de données de séquences. Pourquoi ? Le z-score se calculant à partir de la moyenne et de l'écart type, il était nécessaire de parcourir au moins une fois toute la banque.

Il reste enfin trois autres fonctions que j'ai écrites et que je n'ai pas décrites ici : `Getfreq()`, `Bactracking()` et `LengthSeqVSCutOff()`. Ces fonctions ne présentent pas un intérêt algorithmique particulier par rapport aux autres fonctions décrites – elles ne comportent aucune contrainte liée par exemple à la gestion de séquences multiples séparées l'existence des blocs –. La première permet à la réception d'un symbole (un acide aminé codé sur une lettre ou un gap) de

retourner sa fréquence correspondante : elle repose sur une structure de contrôle type swtich/case. La seconde fonction permet à partir de la matrice d'alignement contenant les chemins, de récupérer le chemin optimal en remontant toute la matrice de la case la plus basse à droite vers la case la plus haute à gauche ('ROOT'), en suivant chaque fois les informations de directions à chaque case. Une fois le chemin récupéré, la fonction `Backtracking()` inverse la séquence (contenant les lettre i, s ou d) du chemin obtenu, pour l'avoir dans l'ordre et renvoie ce chemin à la procédure appelante. La fonction `LengthSeqVSCutOff()` permet quant à elle de vérifier si l'alignement des blocs sur la séquence cible ne dépasse pas une longueur seuil, imposée par l'utilisateur en paramètre d'entrée. Son intérêt sera justifié dans les sections à venir. L'astuce pour cette fonction est d'avoir utilisé la séquence du chemin optimal, et compter le nombre de délétions présentes en début et fin de chemin pour déterminer par opération de soustraction (entre la taille totale du chemin et le nombre de délétion), la longueur totale nécessaire pour aligner les blocs sur la séquence cible.

ANNEXE 4

Modeller : Stratégie adaptée aux blocs

« Changez vos stratégies et tactiques, mais jamais vos principes.»

John Kessel (1950 – 20XX ap JC)

Le logiciel Modeller n'a pas une nomenclature difficile pour son exécution : il suffit de lancer le programme avec le fichier de directive désiré. Ce fichier est donc très important et va être détaillé pour deux cas de figure : i) la construction d'un modèle et ii) la reconstruction des boucles.

4.1 Construction d'un modèle sous Modeller

Quelque soit l'approche, Modeller prends toujours deux fichier en entrée standard pour construire un modèle (en plus des fichiers PDB de chaque *templates*) : un fichier d'alignement dans un format particulier, correspondant dans mon cas aux alignements de P450s – c'est l'alignement des *templates* sur la séquence du P450 à reconstruire, issu de *Caliseq* après traitement – et un ficher de directive, comprenant les commandes adéquate pour Modeller.

4.1.1 Fichier d'alignement (ali)

Le fichier d'alignement utilisé par Modeller s'apparente beaucoup au format fasta, c'est pourquoi j'avais dès le départ opté pour travailler les alignements à ce format. Les principales particularités correspondent aux entêtes, mais également à certains marqueurs qu'il ne faut pas oublier en fin de séquence comme il a été à la section 2.2.1 des annexes. La Figure A 2-5 de la page 307 présente le fichier d'alignement que j'ai donné à Modeller pour la reconstruction d'un CYP 2A1 à partir de structure proche (environs 30% d'identité en séquence). Trois structures de référence ont été utilisées : un CYP 2B4 (1suo), un CYP 2A13 (2p85) et un CYP 2D6 (2fq9). C'est un cas de figure assez simple dans la mesure où les CYPs appartiennent tous à la famille des 2. Les séquences des trois structures de références ont été préalablement alignées avec la séquence de la protéine cible à construire.

4.1.2 Fichier de directives (top)

Le **fichier de directives** est le fichier contenant toutes les commandes, les renseignements nécessaire au bon fonctionnement Modeller. La possibilité en commandes est nombreuse. Le fichier se comporte comme un script shell[25] avec son lot de variables et de fonctions. Il est à noter que le « langage » de ce fichier change légèrement entre les différentes versions de Modeller. Ainsi, dans sa dernière version, Modeller 9, l'interfaçage directive/exécution du logiciel fait intervenir un script python à la place du shell, et le langage s'apparente plus à de l'objet (en terme de programmation). L'exemple de la Figure A 4-1 est le fichier de directives pour le fichier d'alignement présenté précédemment.

[25] Il s'agit d'un langage de programmation et d'interfaçage avec la machine sous unix

```
#

INCLUDE                                        # Include the predefined TOP routines
SET ALNFILE  = 'CYP1A2.ali'                    # alignment filename
SET KNOWNS   = '2p85' '1suo' '2f9q'            # codes of the templates
SET SEQUENCE = 'CYP1A2'                        # code of the target
SET ATOM_FILES_DIRECTORY = './:../atom_files'  # directories for input atom files
SET STARTING_MODEL= 1                          # index of the first model
SET ENDING_MODEL  = 100                        # index of the last model

SET TOPOLOGY_MODEL = 1, HYDROGEN_IO = on, HETATM_IO = on, WATER_IO = on
SET TOPLIB = '$(LIB)/top.lib'
SET PARLIB = '$(LIB)/par.lib'

SET DEVIATION     = 4.0               # have to be >0 if more than 1 model
SET RAND_SEED     = -12312            # to have different models from another TOP file
SET FINAL_MALIGN3D = 1
SET MD_LEVEL = 'refine_3'
# SET OUTPUT_CONTROL = 1 1 1 1 1

SET REPEAT_OPTIMIZATION = 3
CALL ROUTINE = 'model'                 # do homology modelling
```

Figure A 4-1 Exemple de fichier de directives pour le fichier d'alignement présenté précédemment.

Le fichier de directives ressemble donc à un script shell où tous les commentaires sont protégés par la balise « # », les variables sont définies par la commande « SET » et le lancement des différentes fonctions se fait par la commande « CALL ROUTINE ». Bien entendu, il en existe bien d'autres, que je serai amené à présenter plus tard. Ici, la première commande INCLUDE indique à Modeller de qu'il doit récupérer toutes les fonctions et les variables prédéfinies et préformatées au programme (sinon, il faudrait définir chaque variable utilisée). La variable ALNFILE va contenir le nom du fichier d'alignement pour Modeller. KNOWNS est une variable qui répertorie les codes PDB des structures de références utilisées (2p85, 1suo et 2f9q dans le cas présent). SEQUENCE est la variable qui correspond au nom de la protéine à construire à partir de sa séquence. Elle permet également de donner la racine des noms de fichiers qui sont générés. ATOM_FILES_DIRECTORY renseigne à Modeller où sont situés les fichiers PDB des structures *templates*. STARTING_MODEL et ENDING_MODEL correspondent au premier et au dernier modèle créés. La ligne avec TOPOLOGY_MODEL permet de prendre en compte à la fois les atomes d'hydrogène, les hétéroatomes et l'eau (qu'on peut mettre à off). DEVIATION et RAND_SEED permettent lors de nombreuses constructions de modèle une certaines variabilités entre eux. FINAL_MALIGN3D permet d'aligner structuralement le modèle résultant avec les *templates*, MD_LEVEL correspond au degré de raffinement demandé (par défaut 1) pour les boucles par un recuit simulé (boucles qui sont soit définies par l'utilisateur, soit automatiquement repéré par Modeller) et REPEAT_OPTIMIZATION permet enfin de définir le nombre de cycle d'optimisation (par défaut 1). La commande

CALL_ROUTINE se charge alors de lancer le programme. Il faut moins d'une heure pour construire un seul modèle, ce qui est relativement rapide.

4.2 Reconstruction des boucles sous Modeller

Sous Modeller, il est possible de reconstruire les boucles durant le processus de construction du modèle, vu précédemment. Il est également possible d'effectuer ce travail postérieurement à celui de la reconstruction du modèle. Dans ce dernier cas, Modeller prends également deux fichiers d'entrée, dont un seul à paramétrer, le second étant la structure au format PDB sur laquelle on effectue les opérations. Le fichier de directive est construit de manière identique à celui présenté précédemment. À la différence près que l'information sur le fichier d'alignement est remplacé par celui du fichier PDB à utiliser (généralement, un modèle issu de Modeller). Ce fichier comporte un nombre important de directives, mais les plus importantes sont celles permettant la délimitation des zones de boucles qui sont présentés à la Figure A 4-2, toujours pour l'exemple du CYP 1A2.

```
SUBROUTINE ROUTINE = 'select_loop_atoms'
  PICK_ATOMS SELECTION_SEGMENT =   '73'     '82'   , SELECTION_STATUS = 'initialize'
  PICK_ATOMS SELECTION_SEGMENT =   '89'     '95'   , SELECTION_STATUS = 'add'
  PICK_ATOMS SELECTION_SEGMENT =   '123'    '134'  , SELECTION_STATUS = 'add'
  PICK_ATOMS SELECTION_SEGMENT =   '154'    '160'  , SELECTION_STATUS = 'add'
  PICK_ATOMS SELECTION_SEGMENT =   '180'    '186'  , SELECTION_STATUS = 'add'
  PICK_ATOMS SELECTION_SEGMENT =   '201'    '226'  , SELECTION_STATUS = 'add'
  PICK_ATOMS SELECTION_SEGMENT =   '252'    '255'  , SELECTION_STATUS = 'add'
  PICK_ATOMS SELECTION_SEGMENT =   '264'    '276'  , SELECTION_STATUS = 'add'
  PICK_ATOMS SELECTION_SEGMENT =   '319'    '341'  , SELECTION_STATUS = 'add'
  PICK_ATOMS SELECTION_SEGMENT =   '403'    '421'  , SELECTION_STATUS = 'add'
  PICK_ATOMS SELECTION_SEGMENT =   '459'    '463'  , SELECTION_STATUS = 'add'
  PICK_ATOMS SELECTION_SEGMENT =   '469'    '473'  , SELECTION_STATUS = 'add'
  PICK_ATOMS SELECTION_SEGMENT =   '482'    '489'  , SELECTION_STATUS = 'add'
  RETURN
END_SUBROUTINE
```

Figure A 4-2 Création d'une fonction permettant la déclaration des régions variables à Modeller. Les numéros de résidu de début et de fin permettent de délimiter ces régions. Cas de la reconstruction d'un CYP 1A2 en utilisant mon set de bloc général

Une fois les résidus délimitant les boucles renseignés, c'est au tour des résidus délimitant les blocs : des contraintes spatiales de distances, d'angles et de torsions vont être appliquées afin de perturber un minimum l'agencement spatial au niveau de ces régions. La déclaration de ces résidus est montrée à la Figure A 4-3.

```
SUBROUTINE ROUTINE = 'special_restraints'
  SET ADD_RESTRAINTS = on
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '83'    '88'
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '96'    '122'
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '135'   '153'
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '161'   '179'
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '187'   '200'
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '227'   '251'
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '256'   '263'
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '277'   '318'
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '342'   '402'
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '422'   '458'
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '464'   '468'
  MAKE_RESTRAINTS RESTRAINT_TYPE = 'ALPHA', RESIDUE_IDS = '474'   '481'
  RETURN
END_SUBROUTINE
```

Figure A 4-3 Création d'une fonction permettant la déclaration les résidus entre lesquels des contraintes spatiales vont être appliqués : de distance, d'angles de torsion, impropres et dièdres φ, ψ, et χ, etc… Cas de la reconstruction d'un CYP 1A2 en utilisant mon set de bloc général

ANNEXE 5

Des alignements en vrac…

« C'est parce qu'on imagine simultanément tous les pas qu'on devrait faire qu'on se décourage, alors qu'il s'agit de les aligner un à un.»

Marcel Jouhandeau (1888 – 1979 ap JC)

Dans cette annexe, seront présentés quelques alignements représentatifs qui n'ont pu figurer dans les résultats, par souci de place. Les régions conservées sont indiquées par des majuscules et les régions variables par des minuscules.

```
1oxa     1   ------atVP dleSDSFHVD w--------Y STYAELRETA PVTPVRFLg- qDAWLVTGYD EAKAALSDlR LSSDPkKKyp gvevEFPAYL GFPEDVRNyf  85
1gwi     1   ------arIP ---LDPFVTD l--------D GESARLRAAG PLAAVELPgg vPVWAVTHHA EAKALLTDpR LVKDInVWga wrrgEIPADW PLIGLAN--p  81
1rom     1   -------aPS fpfSRASGPE p--------P AEFAKLRATN PVSQVKLFdg sLAWLVTKHK DVCFVATSeK LSKVR-TR-- ----QGFPEL SASGKQAAka  78
1cpt     1   --mdaratIP ehiARTVILP qgyaddeviY PAFKWLRDEQ PLAMAHIEgy dPMWIATKHA DVMQIGKQpG LFSNAeGSei lydqNNEAFM RSISGGCPhv  98
3cpp     1   ----nlapLP phvPEHLVFD fd------mY NPSNLSAGVQ EAWAVLQEsn vPDLVWTRCN GGHWIATRgQ LIREAyEDyr hfssECPFIP REAGEAYDfi  90
1e9x     1   -----msaVA lprVSGGHDE hg------hL EEFRTDPIGL MQRVRDECg- -DVGTFQLAG KQVVLLSG-S HANEFfFRag dddlDQAKAY PFMTPIFG--  84
2hpd     1   ----tikeMP qpkTFGELKN lp-------- LLNTDKPVQA LMKIADELg- -EIFKFEAPG RVTRYLSSqR LIKEAcDE-s rfdkNLSQAL KFVRDFAG-d  84
1og5     1   --------PP gptPLPVIGN i--------L QIGIKDISKS LTNLSKVYg- -PVFTLYFGL KPIVVLHGyE AVKEAlID-l geefSGRGIF PLAERANRgf  81
1pq2     1   ------klPP gptPLPIIGN m--------L QIDVKDICKS FTNFSKVYg- -PVFTVYFGM NPIVVFHGyE AVKEAlID-n geefSGRGNS PISQRITKgl  83
1nr6     1   -----gklPP gptPFPIIGN i--------L QIDAKDISKS LTKFSECYg- -PVFTVYLGM KPTVVLHGyE AVKEAlVD-l geefAGRGSV PILEKVSKgl  84
1dt6     1   --------PP gptPFPIIGN i--------L QIDAKDISKS LTKFSECYg- -PVFTVYLGM KPTVVLHGyE AVKEAlVD-l geefAGRGSV PILEKVSKgl  81
CYP3A4   1   hglfkklgIP gptPLPFLGN ---------I LSYHKGFCMF DMECHKKYg- -KVWGFYDGQ QPVLAITDpD MIKTVlVKec ysvfTNRRPF GPVGFMKS--  87

1oxa    86  atnmGTSDPP THTRLRKLVS QEFTVRRVEA MRPRVEQITA ELLDEV---- ------gDSG VVDIVDRFAH PLPIKVICEL LG------vd eAARGAFGRW  169
1gwi    82  grsmLTVDGA EHRRLRTLVA QALTVRRVEH MRGRITELTD RLLDELp--- ------aDGG VVDLKAAFAY PLPMYVVADL MG------ie eARLPRLKVL  166
1rom    79  kptfVDMDPP EHMHQRSMVE PTFTPEAVKN LQPYIQRTVD DLLEQMkqk- -----gcANG PVDLVKEFAL PVPSYIIYTL LG-------- -VPFNDLEYL  163
1cpt    99  idslTSMDPP THTAYRGLTL NWFQPASIRK LEENIRRIAQ ASVQRLl--- ------dFDG ECDFMTDCAL YYPLHVVMTA LG------vp eDDEPLMLKL  183
3cpp    91  p---TSMDPP EQRQFRALAN QVVGMPVVDK LENRIQELAC SLIESLr--- -------PQG QCNFTEDYAE PFPIRIFMLL AG------lp eEDIPHLKYL  171
1e9x    84  --egVVFDAS PERRKEMLHN AALRGEQMKG HAATIEDQVR RMIADWg--- -------EAG EIDLLDFFAE LTIYTSSACL IGkkfrdqld gRFAKLYHEL  172
2hpd    85  glftSWTHEK NWKKAHNILL PSFSQQAMKG YHAMMVDIAV QLVQKWerln adehievPED MTRLTLDTIG LCGFNYRFNS FYrdqphpfi tSMVRALDEA  184
1og5    82  givfSNGKKW KEIRRFSLMT LRNFGMGKRS IEDRVQEEAR CLVEELr--- ------kTKA SPCDPTFILG CAPCNVICSI IFhkrfdykd qQFLNLMEKL  172
1pq2    84  giisSNGKRW KEIRRFSLTT LRNFGMGKRS IEDRVQEEAH CLVEELr--- ------kTKA SPCDPTFILG CAPCNVICSV VFqkrfdykd qNFLTLMKRF  174
1nr6    85  giafSNAKTW KEMRRFSLMT LRNFGMGKRS IEDRIQEEAR CLVEELr--- ------kTNA SPCDPTFILG CAPCNVICSV IFhnrfdykd eEFLKLMESL  175
1dt6    82  giafSNAKTW KEMRRFSLMT LRNFGMGKRS IEDRIQEEAR CLVEELr--- ------kTNA SPCDPTFILG CAPCNVICSV IFhnrfdykd eEFLKLMESL  172
CYP3A4  88  a--iSIAEDE EWKRLRSLLS PTFTSGKLKE MVPIIAQYGD VLVRNLr-r- -----eaETG KPVTLKDVFG AYSMDVITST SFgvnidsln -NPQDPFVEN  177

1oxa    170 SSEILVMDPE r--------- --------ae qRGQAAREVV NFILDLVERR R-------TE PGDDLLSALI SVQDDDDg-- --RLSADELT SIALVLLLAG  241
1gwi    167 FEKFFSTQTP ---------- --------pe eVVATLTELA SIMTDTVAAK R-------AA PGDDLTSALI QASENGD--- --HLTDAEIV STLQLMVAAG  236
1rom    164 TQQNAIRTNG s--------- ------star eASAANQELL DYLAILVEQR L-------VE PKDDIISKLC TEQVKPG--- --NIDKSDAV QIAFLLLVAG  236
1cpt    184 TQDFFGVEAA r--------- --------r- -FHETIATFY DYFNGFTVDR R-------SC PKDDVMSLLA NSKLDGN--- --YIDDKYIN AYYVAIATAG  252
3cpp    172 TDQMTRPDGS ---------- ---------m tFAEAKEALY DYLIPIIEQR R-------QK PGTDAISIVA NGQVNGR--- --PITSDEAK RMCGLLLVGG  240
1e9x    173 ERGTDPLAYV dp-------- --ylpiesfr rRDEARNGLV ALVADIMNGR Ian--pptDK SDRDMLDVLI AVKAETG--- tpRFSADEIT GMFISMMFAG  257
2hpd    185 MNKLQRANPD dp-------- --ayd-enkr qFQEDIKVMN DLVDKIIADR Kas-----GE QSDDLLTHML NGKDPETg-- -ePLDDENIR YQIITFLIAG  265
1og5    173 NENIEILSSP wiqvynnfpa lldyfpgthn kLLKNVAFMK SYILEKVKEH Qes--mdmNN PQDFIDCFLM KMEKEKHnq- psEFTIESLE NTAVDLFGAG  269
1pq2    175 NENFRILNSP wiqvcnnfpl lidcfpgthn kVLKNVALTR SYIREKVKEH Qas--ldvNN PRDFIDCFLI KMEQEKDnq- ksEFNIENLV GTVADLFVAG  271
1nr6    176 HENVELLGTP wlqvynnfpa lldyfpgihk tLLKNADYIK NFIMEKVKEH Qkl--ldvNN PRDFIDCFLI KMEQENN--- -lEFTLESLV IAVSDLFGAG  269
1dt6    173 HENVELLGTP ---------- -ldyfpgihk tLLKNADYIK NFIMEKVKEH Qkl--ldvNN PRDFIDCFLI KMEQENN--- -lEFTLESLV IAVSDLFGAG  255
CYP3A4  178 TKKLLRFDFL dpfflsitvf pflipilevl nICVFPREVT NFLRKSVKRM KesrledtQK HRVDFLQLMI DSQNSKEtes hkALSDLELV AQSIIFIFAG  277

1oxa    242 FEASVSLIGI GTYLLLTHPD QLALVRADPS ALP------- ---------- -NAVEEILRY IAPP-eTTTR FAAEEVEIGG VA-IPQYSTV LVANGAANRD  321
1gwi    237 HETTISLIVN AVVNLSTHPE QRALVLSGEA EWS------- ---------- -AVVEETLRF STPTshVLIR FAAEDVPVGD RV-IPAGDAL IVSYGALGRD  317
1rom    237 NATMVNMIAL GVATLAQHPD QLAQLKANPS LAP------- ---------- -QFVEELCRY HTASalAIKR TAKEDVMIGD KL-VRANEGI IASNQSANRD  317
1cpt    253 HDTTSSSSGG AIIGLSRNPE QLALAKSDPA LIP------- ---------- -RLVDEAVRW TAPV-kSFMR TALADTEVRG QN-IKRGDRI MLSYPSANRD  332
3cpp    241 LDTVVNFLSF SMEFLAKSPE HRQELIERPE RIP------- ---------- -AACEELLRR FSLV--ADGR ILTSDYEFHG VQ-LKKGDQI LLPQMLSGLD  319
1e9x    258 HHTSSGTASW TLIELMRHRD AYAAVIDELD ELYgdgrsvs fhalrqipql eNVLKETLRL HPPL-iILMR VAKGEFEVQG HR-IHEGDLV AASPAISNRI  355
2hpd    266 HETTSGLLSF ALYFLVKNPH VLQKAAEEAA RVLvdp-vps ykqvkqlkyv gMVLNEALRL WPTAp-AFSL YAKEDTVLGG EYpLEKGDEL MVLIPQLHRD  363
1og5    270 TETTSTTLRY ALLLLLKHPE VTAKVQEEIE RVIgrnrspc mqdrshmpyt dAVVHEVQRY IDLLptSLPH AVTCDIKFRN YL-IPKGTTI LISLTSVLHD  368
1pq2    272 TETTSTTLRY GLLLLLKHPE VTAKVQEEID HVIgrhrspc mqdrshmpyt dAVVHEIQRY SDLVptGVPH AVTTDTKFRN YL-IPKGTTI MALLTSVLHD  370
1nr6    270 TETTSTTLRY SLLLLLKHPE VAARVQEEIE RVIgrhrspc mqdrsrmpyt dAVIHEIQRF IDLLptNLPH AVTRDVRFRN YF-IPKGTDI ITSLTSVLHD  368
1dt6    256 TETTSTTLRY SLLLLLKHPE VAARVQEEIE RVIgrhrspc mqdrsrmpyt dAVIHEIQRF IDLLptNLPH AVTRDVRFRN YF-IPKGTDI ITSLTSVLHD  354
CYP3A4  278 YETTSSVLSF IMYELATHPD VQQKLQEEID AVLpnkappt ydtvlqmeyl dMVVNETLRL FPIA-mRLER VCKKDVEING MF-IPKGWVV MIPSYALHRD  375

1oxa    322 P-SQFPDPHR FDVTRDTRG- ---------H LSFGQGIHFC MGRPLAKLEG EVALRALFGR FPALSLGIDA DdvvWRRSLL LRGIDHLPVR Ldg-------  403
1gwi    318 ErAHGPTADR FDLTRTSGNr ---------H ISFGHGPHVC PGAALSRMEA GVALPALYAR FPHLDLAVPA AelrNKPVVT QNDLFELPVR Lahh------  402
1rom    318 E-EVFENPDE FNMNRKWPPq d--------P LGFGFGDHRC IAEHLAKAEL TTVFSTLYQK FPDLKVAVPL GkinYTPLNR DVGIVDLPVI F---------  399
1cpt    333 E-EVFSNPDE FDITRFPNR- ---------H LGFGWGAHMC LGQHLAKLEM KIFFEELLPK LKSVELSGPP R---LVATNF VGGPKNVPIR Ftka------  412
3cpp    320 E-RENACPMH VDFSRQKVS- ---------H TTFGHGSHLC LGQHLARREI IVTLKEWLTR IPDFSIAPGA Q--iQHKSGI VSGVQALPLV Wdpattkav-  405
1e9x    356 P-EDFPDPHD FVPARYEQPr qedllnrwtW IPFGAGRHRC VGAAFAIMQI KAIFSVLLRE YEFEMAQPPE SyrnDHSKMV VQLAQPACVR Yrrrt-----  449
2hpd    364 KtIWGDDVEE FRPERFENPs ---aipqhaF KPFGNGQRAC IGQQFALHEA TLVLGMMLKH FDFEDHTNYE LdikETLTLK PEGFVVKAKS Kkiplgg---  457
1og5    369 N-KEFPNPEM FDPHHFLDEg g-nfkkskyF MPFSAGKRIC VGEALAGMEL FLFLTSILQN FNLKSLVDPK N---LDTTPV VNGFASVPPF Yqlcfipv--  461
1pq2    371 D-KEFPNPNI FDPGHFLDKn g-nfkksdyF MPFSAGKRIC AGEGLARMEL FLFLTTILQN FNLKSVDDLK N---LNTTAV TKGIVSLPPS Yqicfipv--  463
1nr6    369 E-KAFPNPKV FDPGHFLDEs g-nfkksdyF MPFSAGKRMC VGEGLARMEL FLFLTSILQN FKLQSLVEPK D---LDITAV VNGFVSVPPS Yqlcfipih-  462
1dt6    355 E-KAFPNPKV FDPGHFLDEs g-nfkksdyF MPFSAGKRMC VGEGLARMEL FLFLTSILQN FKLQSLVEPK D---LDITAV VNGFVSVPPS Yqlcfipihh  449
CYP3A4  376 P-KYWTEPEK FLPERFSKKn k-dnidpyiY TPFGSGPRNC IGMRFALMNM KLALIRVLQN FSFKPCKETQ IplkLSLGGL LQPEKPVVLK Vesrdgtvsg  473

1oxa    403 - 403
1gwi    402 - 402
1rom    399 - 399
1cpt    412 - 412
3cpp    405 - 405
1e9x    449 - 449
2hpd    457 - 457
1og5    461 - 461
1pq2    463 - 463
1nr6    462 - 462
1dt6    449 - 449
CYP3A4  474 a 474
```

Figure A 5-1 Exemple de l'alignement Clustalw avec le jeu de *templates* ***D***. Les paramètres par défaut ont été utilisés. Les régions conservées dans la séquence (indiquées en majuscule) coïncident à peu près avec le positionnement des blocs par *Caliseq*.

```
1oxa    1   ---------- --ATVPdle- -SDSFHVDWY STYAELRETA PVTPVRFLG- QDAWLVTGYD EAKAALSDLR LSSDPKKKYP GVEVEF-PAY LGFPEDVRny  84
2hpd    1   ----tikemp qpKTFGelkn lPLLNTDKPV QALMKIADEL GEIFKFEAPg RVTRYLSSQR LIKEACDESR FDKNLSQALK FVRDFAgDGL FTSWTHEKnw  96
1gwi    1   ---------- --ARIP---- -LDPFVTDLD GESARLRAAG PLAAVELPGg VPVWAVTHHA EAKALLTDPR LVKDINVWGA WRRGEI-PAD WPLIGLAN--  80
1og5    1   --------pp gpTPLPvign iLQIGIKDIS KSLTNLSKVY GPVFTLYFGl KPIVVLHGYE AVKEALIDLG EEFSGRGIFP LAERAN-RGF GIVFSNGKkw  91
1pq2    1   ------klpp gpTPLPiign mLQIDVKDIC KSFTNFSKVY GPVFTVYFGm NPIVVFHGYE AVKEALIDNG EEFSGRGNSP ISQRIT-KGL GIISSNGKrw  93
1nr6    1   -----gklpp gpTPFPiign iLQIDAKDIS KSLTKFSECY GPVFTVYLGm KPTVVLHGYE AVKEALVDLG EEFAGRGSVP ILEKVS-KGL GIAFSNAKtw  94
CYP3A4  1   hglfkklgip gpTPLPflgn -ILSYHKGFC MFDMECHKKY GKVWGFYDGq QPVLAITDPD MIKTVLVKEC YSVFTNRRPF GPVGFM-KSA ISIAEDEE-w  97

1oxa    84  -----fATNM GTSDPPTHTR LRKLVSQEFT VRRVEAMRPR VE--QITAEL LDEV---GDS GVVDIVDRFA HP-----LPI KVICELLGVD EAARGAFGRW  169
2hpd    97  kkah--NILL PSFSQQAMKG YHAMMVDIAV QLVQKWERLN ADehIEVPED MTRLt-lDTI GLCGFNYRFN SFyrd--QPH PFITSMVRAL DEAMNKLQRA  191
1gwi    80  -----pGRSM LTVDGAEHRR LRTLVAQALT VRRVEHMRGR IT--ELTDRL LDELp--ADG GVVDLKAAFA YP-----LPM YVVADLMGIE EARLPRLKVL  166
1og5    92  keirrfSLMT LRNFGMGKRS IEDRVQEEAR CLVEELRKTK AS--PCDPTF ILGCapcNVI CSIIFHKRFD YKdqqflNLM EKLNENIEIL SSPWIQVYNN  189
1pq2    94  keirrfSLTT LRNFGMGKRS IEDRVQEEAH CLVEELRKTK AS--PCDPTF ILGCapcNVI CSVVFQKRFD YKdqnflTLM KRFNENFRIL NSPWIQVCNN  191
1nr6    95  kemrrfSLMT LRNFGMGKRS IEDRIQEEAR CLVEELRKTN AS--PCDPTF ILGCapcNVI CSVIFHNRFD YKdeeflKLM ESLHENVELL GTPWLQVYNN  192
CYP3A4  98  krlr--SLLS PTFTSGKLKE MVPIIAQYGD VLVRNLRREA ETgkPVTLKD VFGAysmDVI TSTSFGVNID SLnnpqdPFV ENTKKLLRFD FLDPFFLSIT  195

1oxa    170 SSEILVMDPE rAEQRGQAAR EVVNFILDLV ERRR-----T EPGDDLLSAL ISVQDDDDg- -----RLSAD ELTSIALVLL LAGFEASVSL IGIGTYLLLT  258
2hpd    192 NPDDPAYDEN -KRQFQEDIK VMNDLVDKII ADRKas---G EQSDDLLTHM LNGKDPETg- ----ePLDDE NIRYQIITFL IAGHETTSGL LSFALYFLVK  282
1gwi    167 FEKFFSTQTP -PEEVVATLT ELASIMTDTV AAKR-----A APGDDLTSAL IQASENGD-- -----HLTDA EIVSTLQLMV AAGHETTISL IVNAVVNLST  253
1og5    190 FPALLDYFPG tHNKLLKNVA FMKSYILEKV KEHQesmdmN NPQDFIDCFL MKMEKEKHn- --qpsEFTIE SLENTAVDLF GAGTETTSTT LRYALLLLLK  286
1pq2    192 FPLLIDCFPG tHNKVLKNVA LTRSYIREKV KEHQasldvN NPRDFIDCFL IKMEQEKDn- --qksEFNIE NLVGTVADLF VAGTETTSTT LRYGLLLLLK  288
1nr6    193 FPALLDYFPG iHKTLLKNAD YIKNFIMEKV KEHQklldvN NPRDFIDCFL IKMEQENN-- ----lEFTLE SLVIAVSDLF GAGTETTSTT LRYSLLLLLK  286
CYP3A4  196 VFPFLIPILE vLN-ICVFPR EVTNFLRKSV KRMKesrleD TQKHRVDFLQ LMIDSQNSke teshkALSDL ELVAQSIIFI FAGYETTSSV LSFIMYELAT  294

1oxa    259 HPDQLALVRA DPSALP---- ---------- ----NAVEEI LRYIAPP-ET TTRFAAEEVE IGGVAIPQYS TVLVANGAAN RDP-SQFPDP HRFDVTRDTR  338
2hpd    283 NPHVLQKAAE EAARVLvdp- vpsykqvkql kyvgMVLNEA LRLWPTAPAF SLYAKEDTVL GGEYPLEKGD ELMVLIPQLH RDKtIWGDDV EEFRPERFEN  381
1gwi    254 HPEQRALVLS GEAEWS---- ---------- ----AVVEET LRFSTPTSHV LIRFAAEDVP VGDRVIPAGD ALIVSYGALG RDErAHGPTA DRFDLTRTSG  335
1og5    287 HPEVTAKVQE EIERVIgrnr spcmqdrshm pytdAVVHEV QRYIDLLPTS LPHAVTCDIK FRNYLIPKGT TILISLTSVL HDN-KEFPNP EMFDPHHFLD  385
1pq2    289 HPEVTAKVQE EIDHVIgrhr spcmqdrshm pytdAVVHEI QRYSDLVPTG VPHAVTTDTK FRNYLIPKGT TIMALLTSVL HDD-KEFPNP NIFDPGHFLD  387
1nr6    287 HPEVAARVQE EIERVIgrhr spcmqdrsrm pytdAVIHEI QRFIDLLPTN LPHAVTRDVR FRNYFIPKGT DIITSLTSVL HDE-KAFPNP KVFDPGHFLD  385
CYP3A4  295 HPDVQQKLQE EIDAVLpnka pptydtvlqm eyldMVVNET LRLFPIA-MR LERVCKKDVE INGMFIPKGW VVMIPSYALH RDP-KYWTEP EKFLPERFSK  392

1oxa    339 G--------- HLSFGQGIHF CMGRPLAKLE GEVALRALFG RFPALSLGID ADdvvWRRSL LLRGIDHLPV RLDG------ -- 403
2hpd    382 Ps--aipqha FKPFGNGQRA CIGQQFALHE ATLVLGMMLK HFDFEDHTNY ELdikETLTL KPEGFVVKAK SKKIplgg-- -- 457
1gwi    336 Nr-------- HISFGHGPHV CPGAALSRME AGVALPALYA RFPHLDLAVP AAelrNKPVV TQNDLFELPV RLAHh----- -- 402
1og5    386 Eggnfkksky FMPFSAGKRI CVGEALAGME LFLFLTSILQ NFNLKSLVDP KN---LDTTP VVNGFASVPP FYQLcfipv- -- 461
1pq2    388 Kngnfkksdy FMPFSAGKRI CAGEGLARME LFLFLTTILQ NFNLKSVDDL KN---LNTTA VTKGIVSLPP SYQIcfipv- -- 463
1nr6    386 Esgnfkksdy FMPFSAGKRM CVGEGLARME LFLFLTSILQ NFKLQSLVEP KD---LDITA VVNGFVSVPP SYQLcfipih -- 462
CYP3A4  393 Knkdnidpyi YTPFGSGPRN CIGMRFALMN MKLALIRVLQ NFSFKPCKET QIplkLSLGG LLQPEKPVVL KVESrdgtvs ga 474
```

Figure A 5-2 Exemple d'alignement par Clustalw avec le jeu ***C***. Comme constaté sur l'alignement, Clustalw semble favoriser les mésappariements à l'insertion de Gap : les régions conservées sont longues.

```
1z8o     1   ---------- ---------- ---------- ---------- -tvpdle-sd s----f-hv- DWYRTYAELR E-TAPVTPVR f-LGQDAWLV TGYDEAKAAL  50
1gwi     1   ---------- ---------- ---------- ---------a ripldpf--- -------vt- DLDGESARLR A-AGPLAAVE lpGGVPVWAV THHAEAKALL  49
2ij2     1   ---------- ---------- ---------- -----kempq pktfgelknl pl---l-ntd KPVQALMKIA DeLGEIFKFE a-PGRVTRYL SSQRLIKEAC  60
1r9o     1   ---------- ---------- ---------- ---rgklppg ptpl------ --plqigik- DISKSLTNLS KvYGPVFTLY f-GLKPIVVL HGYEAVKEAL  57
CYP 3A4  1   malipdlame twlllavslv llylygthsh glfkklgipg ptplpflgni ls---y-hk- GFCMFDMECH KkYGKVWGFY d-GQQPVLAI TDPDMIKTVL  94

1z8o     51  sdlrlssdpk kkyp--gvev efpaylgfpe d---vr--NY FATNMGTSd- p-PTHTRLRK LVSQEFTV-- ----rrveAM RPRVEQITAE LLDEVg----  131
1gwi     50  tdprlvkdin vw-g--awrr geipadwp-l i---gl--AN PGRSMLTVd- g-AEHRRLRT LVAQALTV-- ----rrveHM RGRITELTDR LLDELpa---  129
2ij2     60  -de--srfd- kn-lsqa--- ---------l k---f-vrDF AGDGLFTSwt heKNWKKAHN ILLPSFSQ-- ----qamkGY HAMMVDIAVQ LVQKWerl--  131
1r9o     58  idlgeef-s- gr-g--i--- ---------f plae-r--AN RGFGIVFSn- g-KKWKEIRR FSLMTLRNfg mgkr----SI EDRVQEEARC LVEELrktk-  130
CYP 3A4  95  vkecysvft- nr-r--p--- ---------f g---pv--GF MKSAISIAe- d-EEWKRLRS LLSPTFTS-- ----gklkEM VPIIAQYGDV LVRNLrreae  165

1z8o     132 ds-GVVD-IV DRFAHPLPIK VICELLGVDE K----yr-GE FGRWSSeilv m--------- -------d-- ---------p e-raeqrgqa AREVVNFILD  196
1gwi     130 dg-GVVD-LK AAFAYPLPMY VVADLMGIEE A----rl-PR LKVLFEkffs t--------- -------q-- ---------- -tppeevvat LTELASIMTD  193
2ij2     132 nadEHIE-VP EDMTRLTLDT IGLCGFNYRF NsfyrdqpHP FITSMV---- ---------r aldeamnkln pddpaydenk rq-------- FQEDIKVMND  209
1r9o     130 -a-SPCD-PT FILGCAPCNV ICSIIFHKRF Dykdqqf-LN LMEKLNenik ilsspwipii --dyfpg--- ---------- --thnkllkn VAFMKSYILE  209
CYP 3A4  166 tg-KPVT-LK DVFGAYSMDV ITSTSFGVNI Dslnnpq-DP FVENTKkllr fdfld----p fflsitvfp- ------flip ilevlnicvf PREVTNFLRK  251

1z8o     197 LVERRRTE-- --p---GDDL LSALIRV-q- --DDDdgrls adeltsi--- -------ALV LLLAGFEASV SLIGIGTYLL LTHPDQLALV RRDP------  269
1gwi     194 TVAAKRAA-- --p---GDDL TSALIQA-s- --ENG-dhlt daeivst--- -------LQL MVAAGHETTI SLIVNAVVNL STHPEQRALV LSGE------  265
2ij2     210 LVDKIIADrk asg-eqSDDL LTHMLNGk-- -dPETge--- -------pld deniryqIIT FLIAGHETTS GLLSFALYFL VKNPHVLQKA AEEAarvlv-  294
1r9o     210 KVKEHQESmd mn---nPQDF IDCFLMKmek ekHNQps--- -------eft ieslentAVD LFGAGTETTS TTLRYALLLL LKHPEVTAKV QEEIervigr  296
CYP 3A4  252 SVKRMKESrl edtqkhRVDF LQLMIDSqns keTEShk--- -------als dlelvaqSII FIFAGYETTS SVLSFIMYEL ATHPDVQQKL QEEIdavlpn  341

1z8o     269 ---------- ---SALPNAV EEILRYIAPP ETTTR-FAAE EVEIGG-VAI PQYSTVLVAN GAANRDPKQF P-DPHRFDVT RDTr------ ---GHLSFGQ  344
1gwi     265 ---------- ---AEWSAVV EETLRFSTPT SHVLIRFAAE DVPVGD-RVI PAGDALIVSY GALGRDERAH GPTADRFDLT RTSg------ --nRHISFGH  343
2ij2     294 -dpvpsykqv kqlKYVGMVL NEALRLWPTA PA-FSLYAKE DTVLGGEYPL EKGDELMVLI PQLHRDKTIW GDDVEEFRPE RFE--npsai pqhAFKPFGN  390
1r9o     297 n-rspcmqdr shmPYTDAVV HEVQRYIDLL PTSLPHAVTC DIKFRN-YLI PKGTTILISL TSVLHDNKEF P-NPEMFDPH HFLdeggnfk kskYFMPFSA  393
CYP 3A4  342 k-apptydtv lqmEYLDMVV NETLRLFPIA MR-LERVCKK DVEING-MFI PKGWVVMIPS YALHRDPKYW T-EPEKFLPE RFSkknkdni dpyIYTPFGS  437

1z8o     345 GIHFCMGRPL AKLEGEVALR ALFGRFPALS LGIda---dD VVWRR-SLLL RGIDhLPVRL Dg--------  402
1gwi     344 GPHVCPGAAL SRMEAGVALP ALYARFPHLD LAVpa---aE LRNKP-VVTQ NDLFeLPVRL Ahhh------  403
2ij2     391 GQRACIGQQF ALHEATLVLG MMLKHF-DFE DHT----nyE LDIKE-TLTL KPEG-FVVKA Kskk--ipl-  450
1r9o     394 GKRICVGEAL AGMELFLFLT SILQNF-NLK SLVdpkn-lD TTPVVnGFAS VPPF-YQLCF Ipihh-----  455
CYP 3A4  438 GPRNCIGMRF ALMNMKLALI RVLQNF-SFK PCKet-q-iP LKLSL-GGLL QPEKpVVLKV Esrdgtvsga  503
```

Figure A 5-3 Exemple d'alignement par HHpred avec le jeu **F**. Les gaps sont plus nombreux et les régions conservées moins longues.

```
1z8o     1   ---------- ---------- ---------- ---------- ----tvp--d le------sd s----f--hv -d---WYRTY AELRE-t-AP VTPVRF--lg  34
1gwi     1   ---------- ---------- ---------- ---------- --aripl--d pf-------- --------vt -d---LDGES ARLRA-a-GP LAAVELp-gg  33
1n97     1   ---------- ---------- ---------- ---------- -mkrlsl--r -eaw-p--yl kdl--q--q- -d---PLAVL LAWGRah--P RLFLPL--pr  40
1cpt     1   ---------- ---------- ---------- -----mda-r atipe-h--- --iartvilp qg---y--a- -ddevIYPAF KWLRDeq--P LAMAHIegy-  48
1izo     1   ---------- ---------- ---------- ---------- phdks----- l----d--ns lt---l--lk -e---GYLFI KNRTErynSD LFQARL--lg  38
2cib     1   ---------- ---------- ---------- -------a-l prvsgghde- -h---g--hl ee---f--rt -d---PIGLM QRVRDel-GD VGTFQL--ag  43
1odo     1   ---------- ---------- ---------- ---------- ---alvl--d pt-------- --------ga -d---HHTEH RTLRE-g-GP ATWVDV--lg  31
2ij2     1   ---------- ---------- ---------- -------k-e mpqpktf--g el---k--nl pl---l--nt dk---PVQAL MKIADel-GE IFKFEA--pg  44
1po5     1   ---------- ---------- ---------- ----g--k-l ppgpsplpv- -l---g--nl lq---mdr-k -g---LLRSF LRLREky-GD VFTVYL--gs  45
1r9o     1   ---------- ---------- ---------- ---rg--k-l ppgptpl--- ---------- --plqig-ik -d---ISKSL TNLSKvy-GP VFTLYF--gl  41
CYP 3A4  1   malipdlame twlllavslv llylygthsh glfkk--l-g ipgptpl--p fl---g--ni ls---y--hk -g---FCMFD MECHKky-GK VWGFYD--gq  78

1z8o     35  QDAWLVTGYD EAKAALsdlr lssdpkkky- p--------- gvevefpayl gfped---vr --ny------ f-atnmgtsd -p-PTHTRLR Klvsqe----  106
1gwi     34  VPVWAVTHHA EAKALLtdpr lvkdinvw-- g--------- awrrgeipad wp-li---gl --an------ p-grsmltvd -g-AEHRRLR Tlvaqa----  103
1n97     41  FPLALIFDPE GVEGAL-lae -gt-t-ka-- tf-------- q--------- ---yr---a- lsrl------ t-grglltdw -g-ESWKEAR Kalkdp----  97
1cpt     49  DPMWIATKHA DVMQIG-kqp -glfs-na-- egseilydqn n--------- ---ea---f- mrsisggcph v-idsltsmd -p-PTHTAYR Gltlnw----  120
1izo     39  KNFICMTGAE AAKVFYd--t drf-q--rqn a--------- l--------- ---pkrv-qk --sl------ fgvnaiqgmd -g-SAHIHRK Mlflsl----  97
2cib     44  KQVVLLSGSH ANEFFFra-g dddl--dq-- a--------- k--------- ---a------ ---------- ---------- --yPFMTPIF G---------  80
1odo     32  VQAWSVSDPV LLKQLLtssd vskdarahw- p--------- afgevvgtwp ---la---lw --va------ --venmftay -g-PNHRKLR Rlvapa----  99
2ij2     45  RVTRYLSSQR LIKEAC-de- -srfd-kn-- lsq------- a--------- ---lk---f- vrdf------ a-gdglftsw theKNWKKAH Nillps----  104
1po5     46  RPVVVLCGTD AIREALvdqa eafsgrgk-- i--------- a--------- ---vv---dp --if------ q-gygvifan -g-ERWRALR Rfslatmrdf  108
1r9o     42  KPIVVLHGYE AVKEALidlg eef-s-gr-- g--------- i--------- ---fplae-r --an------ r-gfgivfsn -g-KKWKEIR Rfslmt----  100
CYP 3A4  79  QPVLAITDPD MIKTVLvkec ysvft-nr-- r--------- p--------- ---fg---pv --gf------ m-ksaisiae -d-EEWKRLR Sllspt----  136

1z8o     106 -ftv------ rrveAMRPRV EQITAELLDE Vg-----ds- GVVD-IVDRF AHPLPIKVIC ELLgvde-k- ------yr-g -ef------g r-wsseilvm  175
1gwi     103 -ltv------ rrveHMRGRI TELTDRLLDE Lpa----dg- GVVD-LKAAF AYPLPMYVVA DLMgiee-a- ------rl-p -rl------k v-lfekffst  173
1n97     97  -flp------ knvrGYREAM EEEARAFFGE Wr------g- EERD-LDHEM LALSLRLLGR ALFgkpl--- -----s---p -sl------a e-hal-----  158
1cpt     120 -fqp------ asirKLEENI RRIAQASVQR Lldf----d- GECDfMTDCA LYYPLHVVMT ALG-v-ped- -----de--p -lm------l k-ltq-----  185
1izo     97  -mtp------ ---pHQKRLA ELMTEEWKAA Vtrwe--ka- DEVV-LFEEA KEILCRVACY WAG-v----- -plketev-k -er------a d-dfidmvda  167
2cib     80  -lrg------ eqmkGHAATI EDQVRRMIAD Wg--e-a--- GEID-LLDFF AELTIYTSSA CLIgkk---- ---------- ---frdqldg r-faklyhel  148
1odo     99  -fsa------ rrvdAMRPAV EAMVTGLVDR Laelp--ag- EPVD-LRQEL AYPLPIAVIG HLMgvpq-d- ------rr-d -gf------r a-lvdgvfdt  171
2ij2     104 -fsq------ qamkGYHAMM VDIAVQLVQK Werl---nad EHIE-VPEDM TRLTLDTIGL CGFnyrf-ns f---yrdqph -pf------i t-smv-----  176
1po5     109 g--------- mgkrSVEERI QEEARCLVEE Lrksk-g--- ALLD-NTLLF HSITSNIICS IVFgk----- ---------- ---rfdyk-d p-vflrlldl  174
1r9o     100 -lrnfgmgkr ----SIEDRV QEEARCLVEE Lrktk---a- SPCD-PTFIL GCAPCNVICS IIFhkrf-dy k---dqqf-l -nl------m e-klneniki  177
CYP 3A4  136 -fts------ gklkEMVPII AQYGDVLVRN Lrrea-etg- KPVT-LKDVF GAYSMDVITS TSFgvni-ds l---nnpq-d -pf------v e-ntkkllrf  213

1z8o     175 ---------- -------d-- ---------- ----p----e -rae----qr gqaAREVVNF ILDLVErr-- rte------- -p-----gdd lLSALIRV-q  217
1gwi     173 ---------- -------q-- ---------- ---------- tppe----ev vatLTELASI MTDTVAak-- raa------- -p-----gdd lTSALIQA-s  214
1n97     158 ---------k aldrimaq-t rsplalldl- --aae----a r--------- ---FRKDRGA LYREAEal-- iv-------- ---------- -HPPLSHL--  206
1cpt     185 ---------d f--------- ---------f gveaa----r r--------- ---FHETIAT FYDYFNgf-- tvd--rr-s- cp-----kdd vMSLLANSkl  231
1izo     168 fg-av----- g--------- ---------- ---------- prhw----kg rraRPRAEEW IEVMIEda-- ragllkt-t- -------sgt aLHEMAFH--  215
2cib     149 er-gt----d ---------p --------l- --ayvdp--- ----ylpies frrRDEARNG LVALVA--di mng--rianp -----dr-d mLDVLIAVk-  204
1odo     171 ---------- -------t-- ---------- ---------- ldqa----ea qanTARLYEV LDQLIAak-- rat------- -p-----gdd mTSLLIAArd  213
2ij2     176 ---------r aldeamnk-l npddpay-d- --enk----r q--------- ---FQEDIKV MNDLVDki-- iad--rk-a- sg--eq-sdd lLTHMLNGk-  236
1po5     175 ff-qs----f -slissfs-s --------q- --vfelfsgf lkhfp---gt hrqIYRNLQE INTFIG--qs vek--hratl dp-snp-r-d fIDVYLLRme  246
1r9o     178 ls-spwipii --dyfpg--- ---------- ---------- -thn----kl lknVAFMKSY ILEKVKeh-- qes--md-m- n----n-pqd fIDCFLMKme  235
CYP 3A4  214 df-ld----p fflsitvf-p --------f- --lip----i levl----ni cvfPREVTNF LRKSVKrm-- kes--rl-e- dt-qkh-rvd fLQLMIDSqn  280

1z8o     217 ---------- --ddddgrls adeltsi--- ---------- -----alvll lAGFEASVSL IGIGTYLLLT HPDQLALVRR Dp-------- ----------  269
1gwi     214 ---------- --eng-dhlt daeivst--- ---------- -----lqlmv aAGHETTISL IVNAVVNLST HPEQRALVLS Ge-------- ----------  265
1n97     206 ---------- ---------- ---------- -------pre ralseavtll vAGHETVASA LTWSFLLLSH RPDWQKRVAE S--------- ----------  250
1cpt     231 ---------- ----dgn--- -------yi- -------ddk yinayyvaia tAGHDTTSSS SGGAIIGLSR NPEQLALAKS D--------- ----------  280
1izo     215 ----tqedgs ---------- --------ql dsrmaaieli nvl------- -RPIVAISYF LVFSALALHE HPKYKEWLRS Gn-------- ----------  267
2cib     204 -aet------ g-tprfs--- ---------- --------ad eitgmfismm fAGHHTSSGT ASWTLIELMR HRDAYAAVID Eldelyg--- -dgrsvsfha  271
1odo     214 d--------- -------rls peelrdt--- ---------- -----lllmi sAGYETTVNV IDQAVHTLLT RPDQLALVRK Ge-------- ----------  261
2ij2     236 ---------- d-petge--- -------pl- -------dde niryqiitfl iAGHETTSGL LSFALYFLVK NPHVLQKAAE Eaarvlv--d p-vp--sykq  302
1po5     247 k--d------ k-sdpss--- -------ef- -------hhq nliltvlslf fAGTETTSTT LRYGFLLMLK YPHVTERVQK Eieqvigsh- r-pp--aldd  315
1r9o     236 k--e------ k-hnqps--- -------ef- -------tie slentavdlf gAGTETTSTT LRYALLLLLK HPEVTAKVQE Eiervigrn- r-sp--cmqd  304
CYP 3A4  281 s--k------ e-teshk--- -------al- -------sdl elvaqsiifi fAGYETTSSV LSFIMYELAT HPDVQQKLQE Eidavlpnk- a-pp--tydt  349

1z8o     269 ----SALPNA VEEILRYIAP PE-tttr-FA AEEVEIg-g- VAIPQYSTVL VANGAANRDp kQFP-DPHRF DVTRDTr--- ---------- --------gh  339
1gwi     265 ----AEWSAV VEETLRFSTP TS-hvlirFA AEDVPVg-d- RVIPAGDALI VSYGALGRDe rAHGpTADRF DLTRTSg--- ---------- -----n--rh  338
1n97     250 ---eEAALAA FQEALRLYPP AW-i-ltrRL ERPLLg-e- DRLPPGTTLV LSPYVTQRL- -HFP-DGEAF RPERFL--ee rg-tp---s- -----g--ry  326
1cpt     280 ---pALIPRL VDEAVRWTAP VK-s-fmrTA LADTEVr-g- QNIKRGDRIM LSYPSANRDe eVFS-NPDEF DITRF----- ---------- ------pnrh  351
1izo     267 ---sREREMF VQEVRRYYPF GP-f-lgaLV KKDFVWn-n- CEFKKGTSVL LDLYGTNHDp rLWD-HPDEF RPERFAe--- -----reen- -----lf-dm  344
2cib     272 lrqiPQLENV LKETLRLHPP LI-i-lmrVA KGEFEVq-g- HRIHEGDLVA ASPAISNRIp eDFP-DPHDF VPARYEq--- ----p--rqe dllnrw--tw  355
1odo     261 ----VTWADV VEETLRHEPA VK-hlplrYA VTDIALpdg- RTIARGEPIL ASYAAANRHp dWHE-DADTF DATRTVk--- ---------- --------eh  333
2ij2     303 vkqlKYVGMV LNEALRLWPT AP-a-fslYA KEDTVLg-ge YPLEKGDELM VLIPQLHRDk tIWGdDVEEF RPERFE--np --sai--pq- -----h--af  385
1po5     316 rakmPYTDAV IHEIQRLGDL IPfg-vphTV TKDTQFr-g- YVIPKNTEVF PVLSSALHDp rYFE-TPNTF NPGHFLdang --alk--rn- -----e--gf  399
1r9o     305 rshmPYTDAV VHEVQRYIDL LP-tslphAV TCDIKFr-n- YLIPKGTTIL ISLTSVLHDn kEFP-NPEMF DPHHFLdegg --nfk--ks- -----k--yf  388
CYP 3A4  350 vlqmEYLDMV VNETLRLFPI AM-r-lerVC KKDVEIn-g- MFIPKGWVVM IPSYALHRDp kYWT-EPEKF LPERFSkknk --dni--dp- -----y--iy  432

1z8o     340 LSFGQG---- IHFCMGRPLA KLEGEVALRA LFGRFpal-s LGida----D D-VVw-rr-s lllRGIDHLP VRLD-g---- ----- 402
1gwi     339 ISFGHG---- PHVCPGAALS RMEAGVALPA LYARFphl-d LAvpa----A E-LRn-kp-v vtqNDLFELP VRLA-h-hh- ----- 403
1n97     327 FPFGLG---- QRLCLGRDFA LLEGPIVLRA FFRRF-rl-d PL-------P F-PRv-la-q vtlRPEGGLP ARPR-e---- ----- 385
1cpt     352 LGFGWG---- AHMCLGQHLA KLEMKIFFEE LLPKL-ksve LS-------G P-PRl-vatn fvgGPK-NVP IRFT-k-a-- ----- 412
1izo     345 IPQGGGhaek GHRCPGEGIT IEVMKASLDF LVHQI-ey-d VPe-q--s-L H-YSl--a-r mpsLPES-GF VMSG---irr k---- 411
2cib     356 IPFGAG---- RHRCVGAAFA IMQIKAIFSV LLREY-ef-e MAqppe---S YRNDh-skm- vv-QLAQPAA VRYR-r-r-- -t--- 420
1odo     334 LAFGHG---- VHFCLGAPLA RMEVTLALES LFGRFpdl-r LAdpa----E E-LPp-vp-s lisNGHQRLP VLLH--a--- ----- 396
2ij2     386 KPFGNG---- QRACIGQQFA LHEATLVLGM MLKHF-df-e DHt-----nY E-LDi-ke-t ltlKPEG-FV VKAK-s-kk- -ipl- 450
1po5     400 MPFSLG---- KRICLGEGIA RTELFLFFTT ILQNF-si-a SPvppedi-D LTPRes-gv- gn-V-PPSYQ IRFL-a-r-- h---- 465
1r9o     389 MPFSAG---- KRICVGEALA GMELFLFLTS ILQNF-nl-k SLvdpk-n-L D-TTp-vvng fasVPPF-YQ LCFI-p-ihh ----- 455
CYP 3A4  433 TPFGSG---- PRNCIGMRFA LMNMKLALIR VLQNF-sf-k PCket--q-I P-LKl-sl-g gllQPEKPVV LKVE-s-rdg tvsga 503
```

Figure A 5-4 Exemple d'alignement par HHPred, sur un jeu global ***G*** de *templates*.

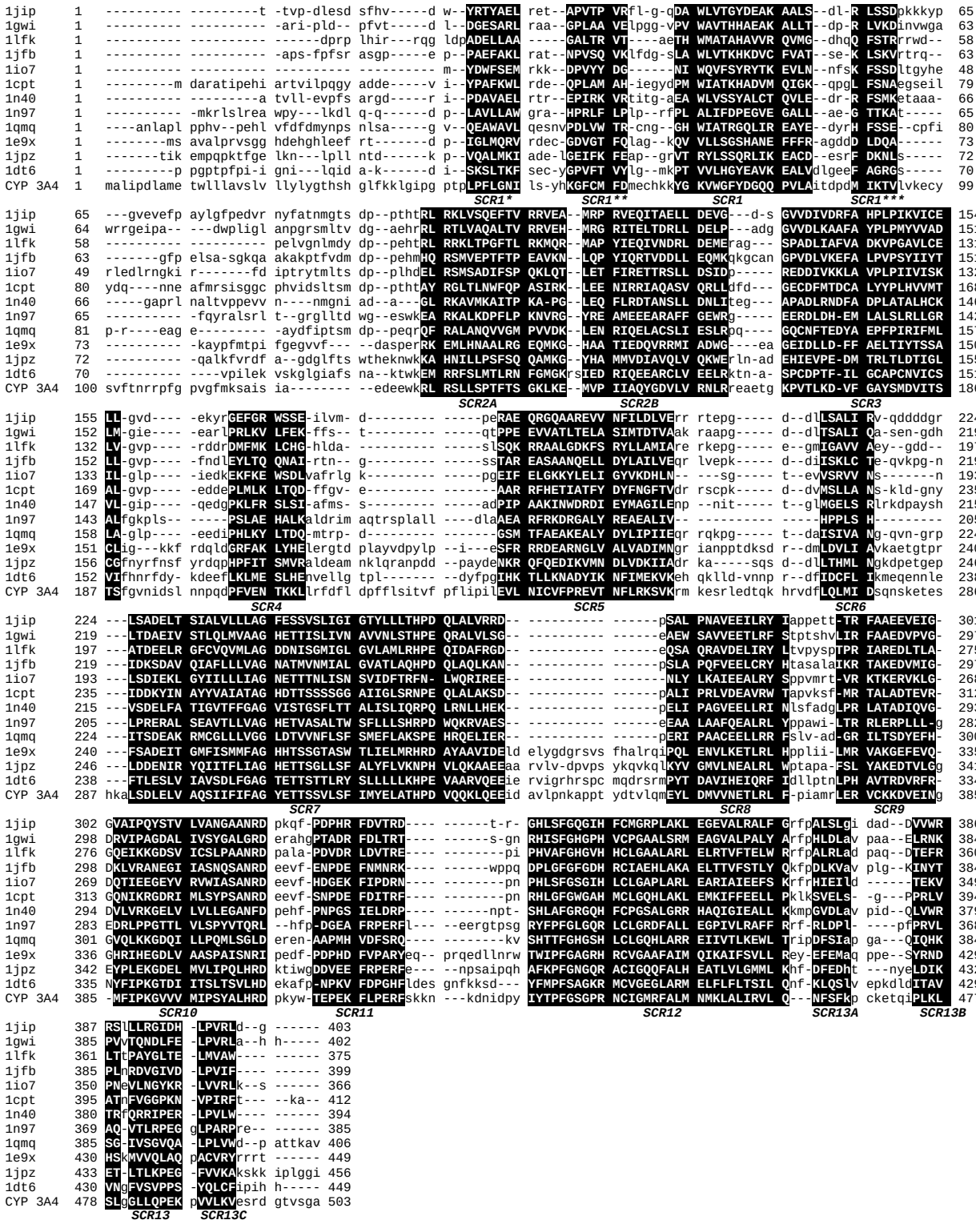


Figure A 5-5 Exemple d'alignement réalisé par Fugue/Orchestrar. 12 templates ont été imposés au programme : $P450_{eryF}$ (1jip), CYP 154C1 (1gwi), $P450_{OxyB}$ (1lfk), $P450_{nor}$ (1jfb), CYP 119 (1io7), $P450_{terp}$ (1cpt), CYP 121 (1n40), CYP 175A1 (1n97), $P450_{cam}$ (1qmq), CYP 51 (1e9x), $P450_{BM3}$ (1jpz), CYP 2C5 (1dt6). Les SCRs (régions structuralement conservées par Orchestrar) sont indiquées par des blocs noirs. La nomenclature utilisée pour la numérotation est la même que celle de P. Jean. Certains blocs n'ont jamais été identifiés avec les autres méthodes, comme le SCR1*** par exemple.

```
Conf: 923466899999999999999999851454301039989876055864675148988 99
Pred: CCHHHHHHHHHHHHHHHHHHHHHHHHHHCCCCCCHHCCCCCCCCHHHHHHHHHCCCCHHHH
  AA: MALIPDLAMETWLLLAVSLVLLYLYGTHSHGLFKKLGIPGPTPLPFLGNILSYHKGFCMF
              10        20        30        40        50        60

Conf: 999998369933997789689997298999999726874434445310011457753784
Pred: HHHHHHHCCCCEEEEECCEEEEECCHHHHHHHHHCCCCCCCCCCCCCEEECCCCCCEEEE
  AA: DMECHKKYGKVWGFYDGQQPVLAITDPDMIKTVLVKECYSVFTNRRPFGPVGFMKSAISI
              70        80        90       100       110       120

Conf: 695389988522063101111134699999888999999752125765433189998633
Pred: CCCHHHHHHHHHCCCCHHHHHHHHHHHHHHHHHHHHHHHHHHCCCCCCCCCHHHHHHHHH
  AA: AEDEEWKRLRSLLSPTFTSGKLKEMVPIIAQYGDVLVRNLRREAETGKPVTLKDVFGAYS
             130       140       150       160       170       180

Conf: 222257653024334440134589987455665542100000000012331132210012
Pred: HHHHHHHHHCCCCCCCCCHHHHHHHHHHHHHHHHHHHHHHHCCCCHHHHHHCCCCCCCCC
  AA: MDVITSTSFGVNIDSLNNPQDPFVENTKKLLRFDFLDPFFLSITVFPFLIPILEVLNICV
             190       200       210       220       230       240

Conf: 379999999999999855421055532001788999876321101335899999999998
Pred: CHHHHHHHHHHHHHHHHHHHHHCCCCCCCHHHHHHHHHHHHHHHHHCCCCCCCHHHHHHHH
  AA: FPREVTNFLRKSVKRMKESRLEDTQKHRVDFLQLMIDSQNSKETESHKALSDLELVAQSI
             250       260       270       280       290       300

Conf: 621111045367878999987530007999998755420127888977874022468998
Pred: HHHHHHCCCHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCCCCCCHHHHHCCCHHHHH
  AA: IFIFAGYETTSSVLSFIMYELATHPDVQQKLQEEIDAVLPNKAPPTYDTVLQMEYLDMVV
             310       320       330       340       350       360

Conf: 875531012246311213761223765676555511504421633326434300345233
Pred: HHHHHHHCCCCCCCCEEECCCEEECCCCCCCCCCEEECCHHHHCCCHHHCCCCCCCCCCCC
  AA: NETLRLFPIAMRLERVCKKDVEINGMFIPKGWVVMIPSYALHRDPKYWTEPEKFLPERFS
             370       380       390       400       410       420

Conf: 456652122100015805552876358999999999998762288746888677520243
Pred: CCCCCCCCCCEEECCCCHHHCCCCCHHHHHHHHHHHHHHHHHEEEEECCCCCCCCCCCCCC
  AA: KKNKDNIDPYIYTPFGSGPRNCIGMRFALMNMKLALIRVLQNFSFKPCKETQIPLKLSLG
             430       440       450       460       470       480

Conf: 78965997299999658421169
Pred: EEEECCCCEEEEEEECCCCCCCC
  AA: GLLQPEKPVVLKVESRDGTVSGA
             490       500
```

Figure A 5-6 Résultat de prédiction de SSE par le logiciel PSIPRED sur la séquence du CYP 3A4. Dans la nomenclature PSIPRED, H correspond à une structure en hélice α, E à une structure en brin β, et c à un coil (ou boucle). Sur la séquence, les SSE attribués dans le fichier PDB du CYP 3A4, sont figurés, en rouge pour les hélices α et en jaune pour les brins β. PSIPRED prédit les SSE avec une fiabilité de 50 à 60%.