Méthodes de découverte de nouveaux domaines dans les séquences biologiques : application à Plasmodium falciparum

Christophe Menichelli

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITE DE MONTPELLIER

En Informatique ED I2S

École doctorale : Information, Structures, Systèmes

Unité de recherche LIRMM

Méthodes de découverte de nouveaux domaines dans les séquences biologiques : application à *Plasmodium falciparum*

Présentée par Christophe MENICHELLI

Le 26 Novembre 2019

Sous la direction de Olivier GASCUEL et Laurent BRÉHÉLIN

Devant le jury composé de

Cédric NOTREDAME, Associate Professor, Univ. Pompeu Fabra, Group leader, CRG Barcelone	Rapporteur
Jacques VAN HELDEN, Professeur, Univ. Aix-Marseille, TAGC Marseille	Rapporteur
Annie CHATEAU, Maîtresse de conférences, Univ. Montpellier, CNRS, LIRMM Montpellier	Examinatrice
Isabelle FLORENT, Professeure, Muséum national d'Histoire naturelle, MCAM Paris	Examinatrice
Sophie SCHBATH, Directrice de Recherche, INRA, MaIAGE Jouy-en-Josas	Examinatrice
Olivier GASCUEL, Directeur de Recherche, CNRS, Institut Pasteur Paris	Directeur de thèse
Laurent BRÉHÉLIN, Chargé de Recherche, CNRS, LIRMM Montpellier	Co-directeur de thèse

UNIVERSITÉ DE MONTPELLIER

Table des matières

Liste des Figures **iv**

Liste des Tables **viii**

Liste des Algorithmes **x**

Introduction générale **3**

I État de l'art 7

1 Méthodes bio-informatique **9**

1.1 Notations 9
1.2 Alignement de séquences 10
1.2.1 Modèle de score d'un alignement 10
1.2.2 Alignement de paires de séquences 13
1.2.3 Alignement multiple 19
1.3 Motifs et domaines 22
1.3.1 Expressions régulières 23
1.3.2 Matrices pondérées 23
1.3.3 HMM : modèle de Markov caché 27
1.4 Découverte de nouveaux motifs 33
1.4.1 Oligo Analysis 34
1.4.2 Gibbs Sampler 35
1.4.3 MEME 36
1.5 Segmentation 36
1.5.1 Détection de point de rupture 37
1.5.2 Représentation linéaire par morceaux 38
1.6 Régression linéaire et LASSO 40

2 Le paludisme **43**
2.1 Origines de *Plasmodium falciparum* 43
2.2 Le cycle de vie 46
2.3 Un génome atypique 47
2.3.1 Le biais en A+T 47
2.3.2 Régions de faible complexité 47
2.3.3 Un protéome mal annoté 50
2.3.4 Une régulation unique de l'expression des gènes 50

II Découverte de domaines protéiques **57**

3 Les protéines et domaines protéiques **59**
3.1 Les protéines 59
3.1.1 Les composants 59
3.1.2 Les différents niveaux de structures 60
3.1.3 Bases de données de protéines 62
3.2 Les domaines protéiques 62
3.2.1 Représentation des domaines 64
3.2.2 Les bases de données de domaines protéiques 64
3.2.3 La base Pfam 65
3.2.4 InterPro 66
3.2.5 La co-occurrence des domaines protéiques 66
3.2.6 Le lien entre domaines et fonctions cellulaires 66

4 Amélioration des outils de comparaison de paires de séquences, et découverte de nouvelles familles de domaines protéiques **67**
4.1 Introduction 67
4.2 Méthode 69
4.3 Analyse du protéome de *Plasmodium falciparum* 77
4.3.1 Évaluation de la qualité des domaines 81
4.3.2 Comparaison avec les familles de domaines connues 84
4.3.3 Redondance des nouvelles familles 88
4.3.4 Comparaison avec les autres bases de domaines générées automatiquement 90
4.3.5 Test contre les versions précédentes de Pfam 91
4.3.6 Annotations fonctionnelles 94
4.3.7 Complexité algorithmique et temps de calcul 94
4.4 Discussion 96

III Découverte des domaines de régulation 99

5 Méthodes de prédiction de l'expression à partir de l'ADN 101
5.1 Méthodes basées sur de la régression linéaire . . . 102
5.1.1 Approche REDUCE . . . 102
5.1.2 Régression linéaire avec une pénalisation LASSO . . . 102
5.2 Méthodes basées sur les réseaux de neurones . . . 104
5.2.1 ExPecto . . . 105
5.2.2 Xpresso . . . 106
5.2.3 Basenji . . . 106
5.3 Prédiction de l'expression chez *Plasmodium falciparum* . . . 106

6 Découverte de longues séquences régulatrices 109
6.1 La méthode DExTER . . . 110
6.1.1 Critère d'optimisation . . . 111
6.1.2 Procédure d'exploration . . . 111
6.1.3 Apprentissage et évaluation du prédicteur . . . 119
6.1.4 Quantification de l'importance des variables . . . 121
6.2 Expériences . . . 123
6.2.1 Analyses préliminaires : Influence des paramètres et des choix de la méthode . . . 124
6.2.2 Premiers résultats : de longues séquences avec une composition spécifique permettent de prédire l'expression chez les différentes espèces . . . 128
6.2.3 La dynamique, la composition et la localisation diffèrent suivant les espèces . . . 131
6.2.4 Les domaines de régulation sont associés à des réglementations très dynamiques tout au long du cycle de vie de *Plasmodium falciparum* et ont des termes GO spécifiques . . . 142
6.2.5 Liens avec la régulation transcriptionnelle et post-transcriptionnelle chez *Plasmodium falciparum* . . . 144
6.2.6 Lien avec les marques épigénétiques . . . 147
6.3 Discussion . . . 149

Conclusion générale 155

LISTE DES FIGURES

1.1 Matrice BLOSUM62 12
1.2 Matrice produite à l'aide de l'algorithme Needleman-Wunsch 15
1.3 Matrice produite à l'aide de l'algorithme Smith-Waterman 16
1.4 Alignement produit par l'algorithme BLAST 19
1.5 Alignement multiple de séquences progressif 20
1.6 Exemple PFM 26
1.7 Exemple PPM 26
1.8 Exemple PWM 26
1.9 Exemple de scores 27
1.10 Structure d'un HMM profil 32
1.11 Détection de point de rupture 38
1.12 Représentation linéaire par morceaux 39

2.1 Distribution du risque de transmission de la malaria dans le monde 44
2.2 Phylogénie des espèces eucaryotes 45
2.3 Cycle de vie du parasite responsable de la malaria 48
2.4 Fréquence d'apparition de chaque acide aminé dans les espèces *P. falciparum*, *S. serivisiae* et l'Homme 49
2.5 Structure d'un gène eucaryote 51
2.6 Les régions régulatrices 53

3.1 Exemple de classification des acides aminés suivant leurs propriétés biochimiques 61
3.2 Les quatre niveaux d'organisation structurelle d'une protéine 63

4.1 Extrait des résultats de BLAST de la protéine Q8IKH9_PLAF7 contre UniRef50 70
4.2 Étapes principales de notre procédure 71
4.3 Densité des hits BLAST par résidu sur la protéine CDAT_PLAF7 . 73
4.4 Comparaison HMM-HMM des domaines identifiés par notre approche qui chevauchent un domaine Pfam connu 80

4.5 Nombre de domaines et FDR obtenus avec différents seuils de e-valeur et p-valeur . . . 81
4.6 Distribution des hits . . . 82
4.7 Scores de qualité mesurés . . . 85
4.8 Comparaisons HMM-HMM des nouvelles familles de domaines et des familles de domaines Pfam . . . 87
4.9 Distances (en nombre de résidus) entre les domaines adjacents . . . 89
4.10 Scores de qualités contre d'autres bases de données générées automatiquement . . . 92
4.11 Comparaisons HMM-HMM des nouvelles familles de domaines contre les familles de domaine de Pfam-B . . . 93
4.12 Similarité entre les annotations GO de pairs de familles . . . 95

5.1 Architecture des promoteurs humains . . . 104
5.2 Schéma de principe de l'architecture d'un réseau de neurones . . . 105

6.1 Exemple de demi-treillis . . . 113
6.2 Remplissage du treillis de manière itérative . . . 114
6.3 Construction du treillis de corrélation . . . 115
6.4 Treillis de corrélation . . . 116
6.5 Extrait du graph d'exploration . . . 117
6.6 Choix du λ et coefficients du LASSO . . . 120
6.7 Effet du nombre de régions . . . 125
6.8 Effet des régions uniformes ou variables . . . 126
6.9 Effet du seuil du gain de corrélation . . . 127
6.10 Effet de la taille des k-mers et de la segmentation . . . 129
6.11 Temps de calcul nécessaire par opération . . . 130
6.12 Compilation des résultats de DExTER pour prédire l'expression des gènes codants chez différentes espèces (*partie 1/3*) . . . 132
6.12 Compilation des résultats de DExTER pour prédire l'expression des gènes codants chez différentes espèces (*partie 2/3*) . . . 133
6.12 Compilation des résultats de DExTER pour prédire l'expression des gènes codants chez différentes espèces (*partie 3/3*) . . . 134
6.13 Caractéristiques des domaines identifiés dans les différentes espèces et conditions . . . 135
6.14 Corrélations des variables les plus importantes dans chaque condition/espèce à l'expression des gènes (*partie 1/3*) . . . 137
6.14 Corrélations des variables les plus importantes dans chaque condition/espèce à l'expression des gènes (*partie 2/3*) . . . 138
6.14 Corrélations des variables les plus importantes dans chaque condition/espèce à l'expression des gènes (*partie 3/3*) . . . 139

6.15 Importance des variables sélectionnées dans chaque condition 140
6.16 Importance des régions *upstream*, *downstream*, centre, ou toute la séquence, pour prédire l'expression dans les différentes conditions . 141
6.17 Conservation des domaines modérée au cours de l'évolution 143
6.18 Importance des domaines au cours du cycle de vie de *Plasmodium falciparum* 145
6.19 Analyse GSEA des variables identifiées 146
6.20 Variables identifiées dans le cycle érythrocytaire de *Plasmodium falciparum* 148
6.21 Prédictions de différentes marques épigénétiques 150
6.22 Corrélation des variables les plus importantes dans chaque condition 151

LISTE DES TABLES

1.1 Résumé des principaux algorithmes de segmentation en PLR 40

2.1 Liste des sites de méthylation et d'acétylation des histones de *Plasmodium falciparum* 54
2.2 Nombre de protéines codantes et de facteurs de transcription par espèce 55

3.1 Liste des acides aminés avec leurs codes 1 lettre et 3 lettres respectifs 60
3.2 Résumé des statistiques Pfam 65

4.1 Résumé du nombre des nouvelles occurrences de domaines identifiées par notre approche 78

5.1 Liste des symboles IUPAC 103

6.1 Matrice des fréquences observées dans chaque séquence pour chaque domaine identifié par DExTER 119

LISTE DES ALGORITHMES

1 Algorithme de Smith-Waterman. 17
2 Algorithme *forward*. 30
3 Identification des domaines potentiels 75
4 Procédure d'exploration de DExTER 118
5 Calcul du score d'importance des variables dans le LASSO 123

Introduction générale

Identifier les différentes composantes d'une séquence biologique (séquence nucléique ou séquence d'acides aminés) constitue un premier pas vers la compréhension de la biologie de l'organisme dont elle est issue. Dans ce cadre, identifier les motifs récurrents dans les séquences est une des problématiques les plus courantes en bio-informatique, que ce soit pour des séquences nucléiques (motifs de fixation de facteurs de transcription, motifs d'épissage, motif de réplication, etc.) ou des séquences protéiques (motifs fonctionnels, motifs structuraux, signaux d'adressage, etc.).

On distingue généralement deux types de problèmes lorsque l'on veut annoter une séquence biologique. Les problèmes d'*identification* consistent à utiliser une base de référence de motifs connus pour en identifier de nouvelles occurrences. Les approches dites de *découverte* consistent à découvrir de nouveaux motifs dont le nombre d'occurrences semble particulièrement élevé dans les séquences étudiées. On parle également de méthodes *non ab initio* pour les approches d'identification, et de méthodes *ab initio* pour les approches de découverte.

Nous nous intéressons dans cette thèse aux problèmes de découverte de motifs, et plus particulièrement de ce que l'on appelle des «domaines», c'est-à-dire des sous-séquences relativement grandes (plusieurs dizaines de nucléotides ou d'acides aminés) que l'on retrouve répétées dans les séquences.

Nous étudierons deux types de domaines. D'une part les domaines protéiques qui sont des composants essentiels des protéines. Il existe de nombreuses bases de données de domaines protéiques proposant chacune différentes méthodes d'identification. Les méthodes d'identification proposent généralement des résultats de bonne qualité mais ne peuvent évidemment pas identifier un domaine absent des bases de données. Concernant la découverte de nouveaux domaines, nous verrons qu'il s'agit d'une tâche qui peut s'avérer relativement difficile, d'autant plus lorsqu'on étudie une espèce éloignée des organismes modèles classiquement étudiés et dont les séquences peuplent une grande partie des bases de données existantes. Un premier apport de cette thèse est donc le développement d'une nouvelle procédure de découverte de domaines protéiques qui permet d'enrichir les bases de données existantes mais également de contrôler la qualité des nouveaux domaines identifiés.

Comme on le verra, une des problématiques essentielles de toutes les méthodes de découverte des domaines protéiques est qu'il est souvent difficile de contrôler les faux positifs.

Dans la dernière partie de ce manuscrit, nous nous intéresserons aux problématiques liées à la prédiction de l'expression des gènes. La recherche des éléments régulateurs de l'expression est un sujet relativement ancien mais qui connaît aujourd'hui un regain d'intérêt de la communauté du fait de la disponibilité de nombreuses données expérimentales et du développement des méthodes d'apprentissage statistique. Les méthodes de prédiction de l'expression des gènes reposent généralement sur l'analyse statistique de données expérimentales (fixation de protéines spécifiques, structure de la séquence ADN, ...). Cependant, des résultats récents ont confirmé l'existence de l'information de régulation de l'expression directement dans la séquence ADN. À côté des motifs de fixation «classiques» et relativement courts (de l'ordre de la dizaine de bp) des facteurs de transcription, coexistent de longues séquences (quelques dizaines ou quelques centaines de bp) dont la composition nucléotidique particulière semble influer l'expression des gènes cibles. Le second apport de cette thèse est donc de développer une nouvelle méthode pour caractériser et identifier ces grandes régions que l'on nomme «domaines de régulation».

Pour ces travaux, nous avons choisi comme sujet d'étude l'organisme *Plasmodium falciparum*, le pathogène responsable du paludisme chez l'Homme. Cette espèce représente un défi pour de nombreuses méthodes de bio-informatique du fait de son éloignement phylogénétique des autres espèces eucaryotes modèles généralement étudiées (animaux, plantes, levures). De fait, nombreuses sont ses protéines dépourvues d'annotations fonctionnelles. En outre, ses mécanismes de régulations transcriptionnelles semblent eux aussi différents des mécanismes classiquement retrouvés chez les autres espèces.

Organisation du manuscrit

Le premier chapitre de cette thèse (partie I) présente un état de l'art des méthodes et notions dont nous aurons besoin tout au long de ce manuscrit. Dans cet état de l'art nous verrons le principe et les méthodes d'alignement de séquences. En effet, une analyse classique en bio-informatique est de commencer par aligner une nouvelle séquence non annotée contre une base de données de séquences annotées afin d'identifier des homologies locales souvent marqueurs d'homologies fonctionnelles. Nous étudierons également les différentes modélisations couramment utilisées pour représenter un motif. Nous caractériserons ensuite la problématique de la découverte de nouveaux motifs et décrirons les principales méthodes classiques proposées pour répondre à ce problème. Nous parlerons ensuite des probléma-

tiques de segmentation d'un signal et nous terminerons ce chapitre par une brève présentation de la régression linéaire et de la pénalisation LASSO qui seront intensivement utilisées dans la partie III de la thèse. Le deuxième chapitre porte sur notre sujet d'étude, *Plasmodium falciparum*, le parasite responsable de la malaria chez l'Homme. Dans ce chapitre nous présenterons les origines de cette espèce puis nous verrons le cycle de vie particulier de ce parasite. Enfin, nous verrons qu'il s'agit encore d'un génome relativement mal annoté et nous présenterons certaines des caractéristiques qui le rendent si atypique. Nous verrons également que son éloignement phylogénétique des autres espèces modèles et sa composition nucléotidique si particulière, rendent souvent les méthodes classiques peu efficaces.

Le troisième chapitre (partie II) de ce manuscrit présente un état de l'art de la composition et de l'organisation des protéines ainsi que des domaines protéiques. Nous présenterons quelques bases de données de protéines et de domaines. Le quatrième chapitre présente les travaux que nous avons réalisés sur l'amélioration des comparaisons de paires de séquences protéiques et la découverte de nouvelles familles de domaines. Nous introduirons dans ce chapitre une méthode que nous avons développée pour résoudre ce problème. Nous proposerons également différents critères pour évaluer la qualité des alignements de séquences multiples produits par notre méthode et d'autres méthodes automatiques. Ces travaux ont été publiés dans la revue PlosCB [MGB18].

Le cinquième chapitre (partie III) présente un état de l'art des méthodologies proposées pour prédire l'expression d'un gène à partir de l'ADN. Le sixième chapitre présente nos travaux sur la découverte de longues séquences régulatrices de l'expression des gènes. Plusieurs études ont montré qu'il existe un lien fort entre la composition nucléotidique de régions particulières et l'expression des gènes. Nous avons donc développé une nouvelle méthode pour explorer l'espace des compositions et des sous-régions possibles. Nous avons testé cette méthode sur plusieurs espèces eucaryotes et notamment *Plasmodium falciparum* pour qui on observe des résultats relativement surprenants, laissant entrevoir d'autres mécanismes de régulation, différents des facteurs de transcriptions classiques.

Première partie

État de l'art

Chapitre 1

Méthodes bioinformatique

Comme on l'a vu en introduction, nous nous intéressons dans cette thèse à la découverte de motifs. Dans ce premier chapitre nous présenterons les modèles, et les problématiques de découverte et d'identification de motifs. Pour cela, nous commencerons par définir un ensemble de notations valables pour l'ensemble de ce document, puis nous rappellerons les principaux algorithmes d'alignement de séquences car ils sont à la base des algorithmes de découverte et d'identification de motifs. Aussi, nous présenterons les algorithmes d'un point de vue général sans qu'ils soient attachés à un type de séquence particulier. Nous parlerons donc indifféremment de séquence ADN ou protéique. Nous verrons ensuite différentes modélisations couramment utilisées pour représenter une famille de séquences et nous détaillerons comment ces modèles sont utilisés pour reconnaître de nouvelles séquences. Nous aborderons ensuite la problématique de la découverte de motifs et de domaines proprement dite. Nous verrons que, suivant le type de séquences manipulées, les méthodes de découverte peuvent être très différentes. Ce chapitre méthodologique se terminera par quelques rappels généraux de régression linéaire qui seront utilisés dans la partie III de cette thèse.

1.1 Notations

Nous donnons ici les principaux symboles que nous utilisons dans ce document.
— x, y : deux séquences protéiques ou nucléotidiques
— x_i : le i-ème symbole de x
— $|x|$: longueur de la séquence x
— X : un ensemble de séquences
— Σ : l'alphabet des symboles possibles ; par exemple pour les séquences nucléotidiques : $\Sigma = \{A, C, G, T\}$
— Σ_i : i-ème élément de Σ

— $|X|$: nombre d'éléments dans l'ensemble X, par exemple pour les séquences nucléotidiques : $|\Sigma| = 4$

1.2 Alignement de séquences

Une des analyses les plus courantes en bio-informatique est de chercher à établir si deux, ou plusieurs, séquences sont homologues en évaluant leur similarité. Pour cela, il est fréquent de réaliser un alignement de ces séquences afin d'identifier les positions conservées et les éventuelles mutations, insertions et délétion, et d'établir un score de cet alignement pour certifier ou rejeter l'homologie. Pour pouvoir réaliser un alignement, il faut répondre aux quatre problématiques qui sont :

1. quel type d'alignement : global, local ?
2. quelle fonction de score pour évaluer l'alignement ?
3. quel algorithme pour trouver l'alignement qui optimise le score ?
4. quelle méthode statistique pour évaluer la significativité du score de l'alignement ?

Dans la suite de cette section, nous commencerons par présenter la méthode classique pour évaluer la similarité de deux séquences alignées. Nous verrons ensuite les principales méthodes d'alignement de paires de séquence puis les méthodes d'alignement multiple d'une famille de séquence.

1.2.1 Modèle de score d'un alignement

Lorsque l'on compare des séquences biologiques, l'objectif est souvent de rechercher des preuves d'une histoire évolutive commune entre ces séquences. Différents processus de mutation des séquences se produisent lors de l'évolution. Ainsi il est possible qu'un élément de la séquence soit changé en un autre élément, on parle alors de substitution. Il est également possible qu'un nouvel élément soit ajouté ou retiré de la séquence, on parle alors d'insertion ou de délétion. Les événements d'insertion et de délétion occasionnent généralement des décalages dans les alignements. Ces décalages sont communément désignés sous le terme de *gaps*. Le score total d'un alignement sera alors la somme des différentes positions correctement alignées plus une pénalité pour les positions mal alignées et les gaps. En utilisant ce principe d'évaluation, nous faisons l'hypothèse que les différentes mutations d'une séquence sont indépendantes entre elles. Tous les algorithmes présentés ci-dessous utilisent ce principe d'évaluation du score.

Lorsque l'on aligne deux séquences, nous pouvons identifier des événements de substitution, c'est-à-dire le remplacement d'un nucléotide par un autre (dans

des séquences ADN), ou un changement d'acide aminé (pour des séquences protéiques). Chaque élément ayant des propriétés physico-chimiques propres, les différentes substitutions possibles n'ont pas la même probabilité de se produire. En effet, certains changements peuvent dénaturer la fonction de la séquence. À l'inverse, certains changements n'impliquent que peu de modifications de la fonction de la séquence. Nous pouvons citer par exemple le cas des acides aminés Lysine (K) et Arginine (R) qui portent tous deux une charge positive et peuvent donc parfois se substituer sans altérer la fonction de la séquence. Pour caractériser les différentes substitutions, il faut donc établir un score pour chaque paire de nucléotides ou acides aminés. Pour les acides aminés, il existe différentes mesures de similarité. Les plus couramment utilisées sont les matrices PAM et BLOSUM, que nous détaillerons ci-dessous. Pour les nucléotides, le modèle couramment utilisé est plus simple : si les nucléotides sont identiques le score est +1 sinon il est égal à -2.

Les matrices PAM (*Point Accepted Mutation*) [DE77] présentent la similarité entre acides aminés comme la probabilité qu'un acide aminé soit remplacé par un autre acide aminé dans deux séquences ayant une forte similarité. Pour calculer ces matrices, 1 572 séquences ont été groupées en 71 familles dont la similarité entre chaque séquence est d'au moins 85% au sein de chaque famille. Pour chaque famille, les séquences ont été alignées et la probabilité qu'un acide aminé i soit muté en j après un événement de mutation a été estimée à partir de ces alignements. Ces matrices sont très utilisées pour la comparaison de séquence fortement apparentées mais s'avèrent néanmoins bien moins efficaces pour la comparaison de séquences plus distantes [HH92].

Les matrices BLOSUM (*BLOcks SUbstitution Matrix*) [HH92] définissent la similarité entre acides aminés de manière à mieux rendre compte des homologies entre séquences distantes. Pour cela, ces matrices sont construites à partir de blocs conservés au sein d'alignements multiples de plusieurs séquences homologues. La valeur associée à chaque mutation possible correspond au log-ratio de la fréquence observée de substitution d'un acide aminé i en j par la fréquence attendue si la mutation était uniquement dépendante de la fréquence d'apparition de l'acide aminé j. La Figure 1.1 présente la matrice BLOSUM62 construite sur la base d'alignements réels où chaque séquence a au maximum 62% d'identité avec les autres séquences. La matrice BLOSUM62 est maintenant utilisée par défaut pour l'alignement de protéines dans l'implémentation logicielle de BLAST (voir 1.2.2.3).

Il nous reste maintenant encore à définir le score d'alignement d'un élément de Σ et d'un gap, correspondant à une insertion ou délétion. La pénalité est généralement définie soit par une fonction linéaire :

$$\gamma(g) = -gD, \tag{1.1}$$

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

FIGURE 1.1 – Matrice BLOSUM62

Cette matrice regroupe le score de substitution de chaque paire d'acides aminés. Chaque acide aminé est représenté en ligne et en colonne par sa lettre respective (voir Section 3.1). Les couleurs représentent une classification des acides aminés suivant leurs propriétés physico-chimiques.

soit à l'aide d'une fonction affine :

$$\gamma(g) = -D - (g-1)E, \tag{1.2}$$

où g est le nombre de gap, D est une constante correspondant à la pénalité d'ouverture d'un gap et E est une constante correspondant à l'extension d'un gap. Par défaut, l'implémentation de BLAST pour la comparaison de séquences protéiques utilise les valeurs $D = 11$ et $E = 1$ (pour les séquences nucléiques : $D = 5$ et $E = 2$).

Cette pénalité peut également être modélisée de la même manière que précédemment pour les substitutions, à l'aide d'un modèle probabiliste. Nous supposons que la probabilité qu'un gap se produise sur une position donnée est le produit de la fonction $f(g)$ de la taille du gap et de la probabilité combinée des éléments insérés :

$$P(g) = f(g) \prod_{i \in gap} P(x_i). \tag{1.3}$$

Cependant, les données montrent que dans la comparaison de séquences nucléiques ou protéiques, il n'existe pas de différence suffisamment significative dans la distribution des nucléotides, ou acides aminés, liés à une insertion ou une délétion. Donc le produit des $P(x_i)$ est directement fonction de g et donc $P(g) \approx \gamma(g)$.

1.2.2 Alignement de paires de séquences

Les séquences courtes ou fortement similaires peuvent être relativement facilement alignées manuellement. Cependant, les séquences biologiques étudiées en bio-informatique nécessitent très souvent l'alignement de séquences longues, très variables ou extrêmement nombreuses, qui ne peuvent pas être alignées manuellement ou par une exploration exhaustive de tous les alignements possibles. En effet, le nombre d'alignements possibles pour une paire de séquences augmente factoriellement avec la taille des séquences. Pour deux séquences de taille respective $n = |x|$ et $m = |y|$, il existe $\frac{(m+n)!}{m! \times n!}$ alignements possibles. Cela équivaut, pour deux séquences de taille identique, à $\frac{(2n)!}{(n!)^2}$, ce qui peut être approximé par $\frac{2^{2n}}{\sqrt{\pi \times n}}$ [Lan02, Edd04]. Nous commencerons par présenter les algorithmes exacts fondés sur le principe de la programmation dynamique. Ces algorithmes sont en pratique cependant, parfois trop lents pour être appliqués. Nous verrons donc par la suite des heuristiques développées pour pallier les problématiques de calculabilité.

1.2.2.1 Alignement global : algorithme de Needleman-Wunsch

L'algorithme Needleman-Wunsch a été publié en 1970 [NW70]. Cet algorithme est couramment utilisé en bioinformatique pour réaliser un alignement global de paires de séquences.

C'est un algorithme de programmation dynamique qui consiste à remplir une matrice M de dimension $(n+1) \times (m+1)$ où n et m sont les longueurs respectives des séquences x et y. La première étape consiste à remplir la première colonne et la première ligne avec les valeurs suivantes : $M(i,0) = -iD$ et $M(0,j) = -jD$, où D est une pénalité linéaire au nombre de gaps. La seconde étape consiste à remplir la matrice de la manière suivante :

$$M(i,j) = \max \begin{cases} M(i-1,j-1) + s(x_i, y_j) \\ M(i-1,j) - D \\ M(i,j-1) - D \end{cases} \tag{1.4}$$

où $s(x_i, y_j)$ est le score d'alignement des éléments x_i et y_j. Par exemple, ce score peut provenir de la matrice BLOSUM62 dans le cas de l'alignement de séquences protéiques. Le remplissage d'une case $M(i,j)$ de cette matrice s'effectue en déterminant si le score maximal de l'alignement est obtenu par l'alignement de x_i et y_j, par l'alignement de x_i avec un gap, ou par l'alignement de y_j avec un gap. La valeur dans la dernière case $M(n,m)$ correspond au meilleur score possible pour l'alignement des séquences x et y. Il est alors possible de construire un alignement suivant le principe de *traceback*. Cela consiste, en partant de la case $M(n,m)$, à identifier de quelle option est dérivé le score maximal de $M(i,j)$. Si l'option choisie était $M(i-1,j-1) + s(x_i, y_j)$, cela correspond à l'alignement de x_i et y_j. Sinon, l'option $M(i-1,j) - D$ correspond à l'alignement de x_i et un gap et l'option $M(i,j-1) - D$ correspond à l'alignement de y_i et un gap.

Il est à noter que la procédure de traceback permet d'obtenir un alignement de score maximal. Cependant il peut exister plusieurs alignements possibles pour obtenir ce score. Cela se produit lorsque, en construisant l'alignement, plusieurs options permettent d'atteindre la case $M(i,j)$ avec le même score. Dans ce cas, il nécessaire de faire un choix arbitraire. Il est également possible de considérer tous les chemins possibles en les décrivant par une structure en graphe [AE86, Hei89].

La complexité de l'algorithme est de $\mathcal{O}(nm)$. Dans la mesure où n et m sont souvent similaires, la complexité de cet algorithme est souvent notée $\mathcal{O}(n^2)$.

Exemple Prenons l'exemple de deux séquences x = NEEDLEMAN et y = NEALDLMAN. Nous devons donc créer une matrice M de taille 10×10. Nous choisissons la fonction constante $D = 4$ comme pénalité d'insertion de gap. Pour la fonction de score s nous utiliserons la matrice BLOSUM62. Après avoir rempli la première ligne et la première colonne de la manière suivante : $M(i,0) = -iD$ et $M(0,j) = -jD$. Nous pouvons ensuite calculer les éléments restants en reprenant la formule 1.4 en commençant par $M(1,1)$:

$$M(1,1) = \max \begin{cases} M(0,0) + s(N,N) \\ M(0,1) - D \\ M(1,0) - D \end{cases} = \max \begin{cases} 0+6 \\ -4-4 \\ -4-4 \end{cases} = 6 \tag{1.5}$$

		N	E	A	L	D	L	M	A	N
	0	-4	-8	-12	-16	-20	-24	-28	-32	-36
N	-4	6	2	-2	-6	-10	-14	-18	-22	-26
E	-8	2	11	7	3	-1	-5	-9	-13	-17
E	-12	-2	7	10	6	5	1	-3	-7	-11
D	-16	-6	3	6	6	12	8	4	0	-4
L	-20	-10	-1	2	10	8	16	12	8	4
E	-24	-14	-5	-2	6	12	12	14	11	8
M	-28	-18	-9	-6	2	8	14	17	13	9
A	-32	-22	-13	-5	-2	4	10	13	21	17
N	-36	-26	-17	-9	-6	0	6	9	17	27

FIGURE 1.2 – Matrice produite à l'aide de l'algorithme Needleman-Wunsch

La Figure 1.2 représente la matrice complète. On obtient donc que le score du meilleur alignement de x et y est de 27. En partant de la case $M(9,9)$, en rouge, nous remontons dans la matrice et nous obtenons le chemin affiché en jaune, ce qui correspond à l'alignement suivant :

N	E	E	-	D	L	E	M	A	N
N	E	A	L	D	L	-	M	A	N

.

1.2.2.2 Alignement local : algorithme de Smith-Waterman

Dans l'algorithme précédent, nous avons cherché à aligner les séquences entières. Cependant, il est souvent nécessaire en pratique de rechercher le meilleur alignement possible de sous-séquences issues de x et de y. C'est le cas notamment lorsque l'on suspecte deux séquences protéiques de partager un même domaine protéique, c'est-à-dire une sous-séquence conservée durant l'évolution. C'est exactement le type de problèmes auquel nous nous intéressons dans le Chapitre 3 de cette thèse. L'alignement de sous-séquences issues de x et y est appelé un alignement local.

L'algorithme de Smith-Waterman a été publié en 1981 [SW81]. Cet algorithme est fortement similaire à l'algorithme Needleman-Wunsch présenté précédemment. Il y a cependant deux différences essentielles. La première différence est qu'il existe une quatrième option pour remplir les valeurs de $M(i,j)$ qui est le choix du 0 :

$$M(i,j) = \max \begin{cases} 0 \\ M(i-1,j-1) + s(x_i, y_j) \\ M(i-1,j) - D \\ M(i,j-1) - D \end{cases} \tag{1.6}$$

Le choix du 0 correspond au départ d'un nouvel alignement local. En effet, il

		W	H	A	T	T	R	M	L	L
	0	0	0	0	0	0	0	0	0	0
W	0	11	7	3	0	0	0	0	0	0
A	0	7	9	11	7	3	0	0	0	0
T	0	3	5	9	16	12	8	4	0	0
E	0	0	3	5	12	15	12	8	4	0
R	0	0	0	2	8	11	20	16	12	8
M	0	0	0	0	4	7	16	25	21	17
A	0	0	0	4	0	4	12	21	24	20
N	0	0	1	0	4	0	8	17	20	21

FIGURE 1.3 – Matrice produite à l'aide de l'algorithme Smith-Waterman

n'est pas souhaitable d'obtenir un alignement local avec un score négatif. Il est également à noter que, maintenant, la première ligne et la première colonne de M sont initialisées à 0. La deuxième différence essentielle avec l'algorithme précédent est que le début d'un alignement local ne correspond pas nécessairement à la dernière valeur $M(n, m)$. L'alignement local commence à la plus grande valeur $M(i, j)$ de toute la matrice et on applique ensuite une procédure de traceback similaire à celle de l'algorithme Wagner-Fischer. L'alignement local s'arrête lorsque l'on arrive sur une case $M(i, j) = 0$.

Exemple Prenons l'exemple de deux séquences $x =$ WATERMAN et $y =$ WHATTRMLL. Nous devons donc créer une matrice M de taille 9×9. Pour la fonction de score s nous utiliserons la matrice BLOSUM62. Nous choisissons la fonction constante $D = 4$ comme pénalité d'insertion de gap. Après avoir initialisé la première ligne et première colonne à 0, nous pouvons remplir la matrice M en reprenant la formule 1.6 en commençant par $M(1, 1)$:

$$M(1,1) = \max \begin{cases} 0 \\ M(0,0) + s(W,W) \\ M(0,1) - D \\ M(1,0) - D \end{cases} = \max \begin{cases} 0 \\ 0 + 11 \\ 0 - 4 \\ 0 - 4 \end{cases} = 11 \tag{1.7}$$

La Figure 1.3 représente la matrice complète. On trouvera donc que le meilleur alignement local de x et y a un score de 25. En partant de la case $M(7, 6)$, en rouge, nous remontons dans la matrice et nous obtenons le chemin affiché en jaune, ce qui correspond à l'alignement local suivant :

W	-	A	T	E	R	M
W	H	A	T	T	R	M

Algorithme 1 Algorithme de Smith-Waterman.

Entrée: x, y
Sortie: score meilleur alignement local

$n = |x|$
$m = |y|$
M : une matrice de taille $n + 1 \times m + 1$
pour i de 0 à n **faire**
  $M(i, 0) = 0$
fin pour
pour j de 0 à m **faire**
  $M(0, j) = 0$
fin pour
pour i de 0 à n **faire**
  pour j de 0 m **faire**
    $M(i,j) = \max(0,\ M(i-1,j-1)+s(x_i,y_j),\ M(i-1,j)-d,\ M(i,j-1)-d)$
  fin pour
fin pour
return $\max(M)$

1.2.2.3 BLAST : *Basic Local Alignement Search Tool*

Les algorithmes présentés jusqu'ici sont considérés exacts dans le sens où ces algorithmes garantissent de trouver le score d'alignement optimal. Cependant, ils nécessitent parfois des temps de calculs importants et ne peuvent pas être appliqués en batterie sur de grosses bases de données. Pour pallier ce problème, des heuristiques telles que BLAST ont été développées afin d'accélérer le processus de recherche, tout en conservant la meilleure précision possible. BLAST a été développé par Stephen Altschul, Warren Gish et David Lipman au NCBI (*National Center for Biotechnology Information*). La publication originale [AGM+90] est parue en 1990 et est l'une des plus citées dans le monde scientifique.

BLAST commence tout d'abord par rechercher tous les *k-mers* (mots de taille k) contenus dans la première séquence x. Tous les k-mers trouvés sont ensuite stockés dans une table. On effectue ensuite la même recherche dans la seconde séquence y, en vérifiant à chaque fois, si le k-mer trouvé est déjà présent dans la première table. Si tel est le cas, on crée une paire de sous-séquences afin de commencer l'alignement (on parle alors de graines d'alignement). L'algorithme cherche alors, en partant d'une graine, à étendre l'alignement de part et d'autre en calculant à chaque fois le score de l'alignement à partir d'une matrice de substitution (BLOSUM62 par exemple).

En plus du score d'alignement, Karlin et Altschul ont décrit une méthode pour

évaluer la significativité d'un alignement local avec un score S [KA90]. Cette significativité se traduit par une e-valeur (*expected value*), c'est-à-dire le nombre attendu d'alignements locaux d'une séquence contre une base de données ayant un score de similarité supérieur ou égal au score S, si ces séquences étaient des séquences aléatoires.

Étant donné une séquence et une base de données de longueurs respectives m et n. Le nombre d'alignements locaux avec un score de similarité $\geq S$ peut être décrit par une loi de Poisson de paramètre $v = Kmne^{-\lambda S}$. Ce nombre correspond à la e-valeur E :

$$E = Kmne^{-\lambda S}. \tag{1.8}$$

Les paramètres K et λ dépendent de la distribution de probabilités *a priori* des symboles et de la matrice de scores utilisée (ex : BLOSUM62).

La probabilité de trouver exactement x alignements locaux avec un score $\geq S$ est donnée par :

$$P(X = x) = e^{-E}\frac{E^x}{x!}, \tag{1.9}$$

où E est l'e-valeur pour S. Ainsi la probabilité de trouver au moins un alignement de score $\geq S$ par chance est :

$$P(S) = 1 - P(X = 0) = 1 - e^{-E}. \tag{1.10}$$

Nous pouvons voir que la distribution de probabilité des scores suit une *loi d'extremum généralisée*. BLAST renvoie la e-valeur plutôt que la p-valeur car il est plus simple de comparer des différences d'e-valeurs que des différences de p-valeurs.

La complexité algorithmique de la méthode BLAST a été évaluée à $\mathcal{O}(nm)$, ce qui est exactement la même complexité en temps que les autres algorithmes. Cependant l'utilisation des graines d'alignement permet de fortement réduire le nombre d'alignements locaux possibles. De plus, la capacité de la méthode à évaluer la significativité statistique d'un score permet d'interrompre rapidement la phase d'extension des graines d'alignement si nécessaire. Tout ceci rend la méthode BLAST bien plus rapide en pratique. À titre d'exemple, si l'on cherche à aligner une nouvelle séquence protéique cible avec les 116 millions de séquences référencées dans la base UniProt (voir Section 3.1.3), il faut compter seulement une vingtaine de minutes sur un ordinateur de bureau contre plusieurs heures avec l'algorithme Smith-Watherman. Il est alors envisageable d'utiliser cette méthode à l'échelle d'un génome complet.

Exemple Prenons l'exemple de deux séquences : $x =$ FYWSTMIFFKCLLHSTA et $y =$ ILVSTEQYFHCLLHHQE. L'algorithme commence par référencer tous les

F	Y	W	S	T	M	I	F	F	K	C	**L**	**L**	**H**	S	T	A
I	L	V	S	T	E	Q	Y	F	H	C	**L**	**L**	**H**	T	Q	E
0	-1	-3	+4	+5	-2	-3	+3	+6	-1	+9	+4	+4	+8	+1	-1	-1

FIGURE 1.4 – Alignement produit par l'algorithme BLAST
L'algorithme commence par rechercher une correspondance exacte de taille k (3 dans cet exemple) entre les deux séquences. Ici l'algorithme trouve une correspondance sur le mot LLH, en rouge. L'algorithme étend ensuite l'alignement de part et d'autre de manière à identifier un alignement local optimal, ici le cadre bleu. La dernière ligne correspond aux scores de similarité provenant de la matrice BLOSUM62.

k-mers présents dans la première séquence puis recherche dans la deuxième séquence s'il est possible de trouver une correspondance. Dans cet exemple nous choisissons d'utiliser $k = 3$, donc l'algorithme énumère tous les mots de 3 lettres. L'algorithme trouve alors une correspondance entre les deux séquences sur le mot exact LLH, en rouge dans l'exemple Figure 1.4. L'algorithme essaye alors d'étendre l'alignement à gauche et à droite de la graine et s'arrête lorsque le score n'est plus significatif. On obtient alors un alignement local entre ces deux séquences, en bleu dans l'exemple. Nous aurions également pu utiliser le mot exact CLL comme graine d'alignement, l'alignement local obtenu serait identique.

1.2.3 Alignement multiple

Les méthodes d'alignement présentées ci-dessus permettent d'identifier ce qui est commun à deux séquences. Lorsque l'on veut identifier ce qui est conservé dans trois séquences ou plus, ces méthodes ne suffisent plus. Il faut alors recourir aux méthodes d'alignement multiple. Depuis longtemps, les biologistes produisent des alignements multiples manuellement en utilisant leur connaissance et expertise sur l'évolution des séquences. Cependant, la recherche manuelle d'un bon alignement multiple est une tâche longue, souvent erronée et surtout fastidieuse. C'est pourquoi de nombreuses recherches ont été menées pour développer une méthode automatique d'alignement multiple sur la seule base des séquences tout en essayant de tenir compte des contraintes physico-chimiques des éléments et les structures des séquences. Cependant, il s'agit d'un problème NP difficile [WJ94] qui est considéré comme un des plus difficiles en bio-informatique [LLS91, Kar93].

Pour le résoudre, de très nombreuses heuristiques d'alignement multiple ont été développées. Nous pouvons distinguer deux groupes de méthodes, les approches itératives et les approches progressives. Les approches progressives sont probablement encore les méthodes les plus utilisées actuellement. L'objectif de ces mé-

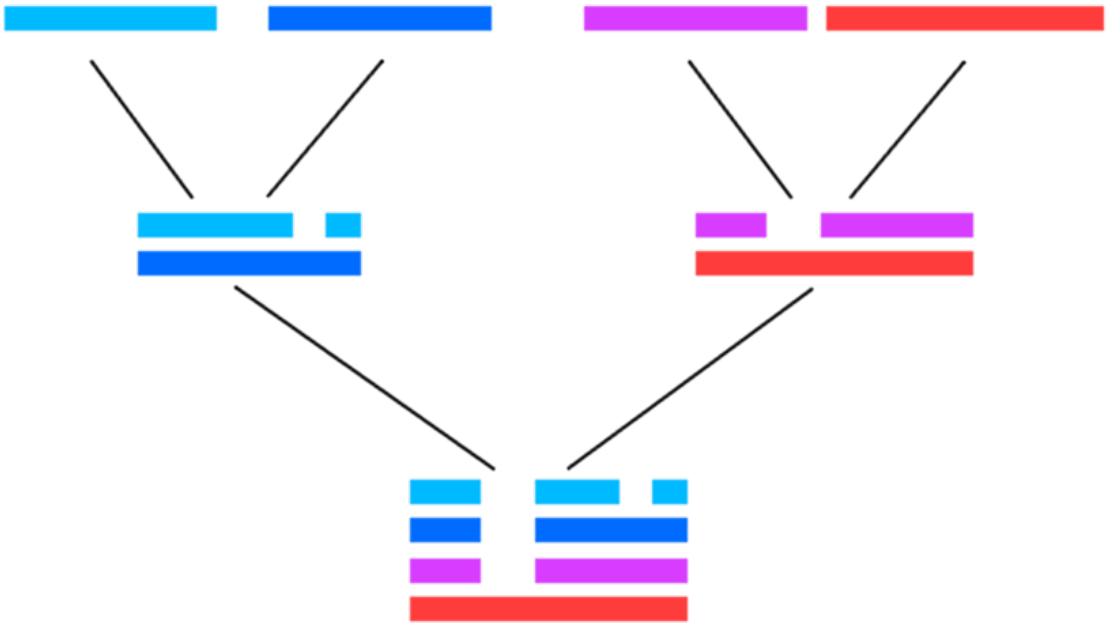


FIGURE 1.5 – Alignement multiple de séquences progressif
Chaque segment de couleur représente une séquence différente. Un alignement progressif est construit sur la base des alignements de paires de séquences suivie d'une stratégie de fusion des alignements jusqu'à obtenir l'alignement multiple complet.

thodes est de partitionner les séquences afin d'aligner les paires de séquences les plus proches, et ensuite d'assembler ces alignements progressivement jusqu'à l'alignement multiple complet (voir Figure 1.5). Un inconvénient majeur des méthodes progressives est leur aspect glouton : les alignements produits à chaque étape sont « figés », il n'est plus possible de les modifier. C'est pourquoi il existe un autre type d'approche, les méthodes itératives. Ces méthodes sont assez similaires aux méthodes progressives mais elles incluent également une étape de réalignement dans la construction de l'alignement multiple pour améliorer la qualité globale. Il existe de très nombreux comparatifs des différentes implémentations mais les auteurs s'abstiennent généralement de choisir la « meilleure » technique [HTHI95]. Nous présenterons ici trois méthodes couramment utilisées qui sont ClustalW, T-coffee et MUSCLE.

1.2.3.1 ClustalW

ClustalW est une méthode d'alignement progressif publiée par Thompson *et al.* en 1994 [THG94]. Cette méthode succède la méthode ClustalV [HBF92]. Clus-

talW est basé sur l'alignement de paires de séquences pour établir un alignement multiple.

L'algorithme de ClustalW se déroule en quatre étapes. La première étape consiste à construire une matrice de distance des paires de séquences en utilisant un algorithme de programmation dynamique comme Needleman-Wunsch. La deuxième étape convertit les scores précédents en distances évolutives en utilisant le modèle de Kimura [Kim83]. L'étape suivante consiste à construire un arbre guide en utilisant un algorithme de *neighbour joining (NJ)* [SN87]. Enfin la dernière étape aligne progressivement les séquences en suivant l'arbre précédent par ordre décroissant de similarité.

Parmi les limites de cette méthode, nous pouvons citer notamment que les alignements des sous-groupes sont figés et donc qu'il n'est pas possible de les réajuster dans l'alignement final. Aussi, l'arbre guide produit n'est pas toujours exact du fait qu'il soit construit seulement sur la base du score de similarité de chaque alignement global par paires. Enfin, l'utilisation d'un alignement global par paire peut poser des problèmes quant au positionnement des séquences de tailles très différentes.

La dernière version de ClustalW, ClustalΩ, a été publiée en 2014 [SH14]. L'amélioration essentielle de cette nouvelle version est l'utilisation de l'algorithme mBed qui permet de fortement réduire le temps et l'espace de calcul nécessaire lors de la première étape de construction de la matrice de distance. Cela permet donc un meilleur passage à l'échelle de la méthode.

1.2.3.2 T-coffee

T-coffee (*Tree-based Consistency Objective Function For alignment Evaluation*) est une méthode d'alignement progressif publié par Notredame *et al.* en 2000 [NHH00]. Cette méthode repose sur les approches classiques d'alignement multiple progressif mais permet d'éviter certains défauts de la stratégie gloutonne. Pour cela, la première étape de T-coffee est de réaliser un pré-traitement des séquences en incorporant les informations d'alignement local et d'alignement global de l'ensemble des séquences pour construire l'arbre guide de l'alignement. Une grande force de T-coffee est que cette méthode peut également inclure des informations très hétérogènes comme un alignement multiple pré-existant, des informations de structure ou encore des informations précisées par l'utilisateur, dans l'optimisation de l'arbre guide de l'alignement.

Cette méthode permet d'obtenir généralement de meilleurs résultats en terme de précision par rapport aux méthodes concurrentes telles que ClustalW. Cependant, du fait du pré-traitement nécessairement, les temps de calcul et la mémoire vive nécessaires sont également légèrement supérieurs aux autres approches [PROC14].

1.2.3.3 MUSCLE

MUSCLE (*MUltiple Sequence Comparison by Log-Expectation*) est une méthode d'alignement itératif publiée par Edgar en 2004 [Edg04b, Edg04a]. Cette méthode procède en trois étapes : la première consiste à créer un alignement progressif classique, la deuxième étape consiste à créer un nouvel alignement basé sur la distance de Kimura [Kim83] calculée sur le premier alignement, enfin la troisième consiste à partitionner l'arbre guide de la deuxième étape en deux sous-arbres, réaligner indépendamment chaque sous-arbre et les réassembler pour obtenir l'alignement complet. La troisième étape est réitérée plusieurs fois. L'algorithme s'arrête à un nombre pré-défini d'itération ou bien lorsque deux itérations consécutives n'améliorent pas l'alignement. La méthode renvoie alors le meilleur alignement obtenu.

1.3 Motifs et domaines

En bio-informatique, il est fréquent de rechercher et d'identifier des séquences, ADN ou protéiques, qui semblent conservées entre différentes espèces ou au sein du même génome. La conservation de ces séquences peut être un bon indicateur d'une pression de sélection dans ces séquences, et permet donc d'identifier des régions potentiellement essentielles au fonctionnement du génome. On pourra retrouver par exemple les sites de fixation des facteurs de transcription, qui jouent un rôle dans la régulation de l'expression des gènes, ou encore les domaines protéiques, qui jouent un rôle essentiel dans les fonctions moléculaires des protéines.

Dans la suite de ce manuscrit, nous distinguerons les séquences relativement courtes (quelques dizaines de nucléotides ou acides aminés), des séquences plus longues (plusieurs dizaines et au-delà). Le terme **motif** désignera indifféremment les motifs courts ou longs. Lorsque nécessaire, on précisera dans le texte si le motif est court ou long, et le terme **domaine** sera synonyme de motif long. Aussi, nous distinguerons les termes suivants :

- le motif, qui est la définition de la séquence conservée ;
- l'occurrence du motif, qui est la présence du motif dans une séquence spécifique ;
- la famille du motif : l'ensemble des occurrences identifiées du motif.

Les algorithmes d'alignement multiple présentés à la section 1.2.3 permettent de visualiser ou d'identifier ce qui est conservé dans une famille de séquences particulières. Souvent, la question est alors de déterminer si une nouvelle séquence appartient elle aussi à cette famille. Pour répondre à cette question, la première étape est de déterminer une modélisation adéquate permettant de résumer l'information de la famille en question. Ensuite, partant de cette modélisation, la

question est de savoir comment évaluer la similarité entre une nouvelle séquence et le modèle produit. Le processus permettant d'associer une séquence à un modèle à l'aide d'un score ou d'une probabilité sera désigné ici comme étant le processus d'identification.

Dans la section suivante, nous présentons différents types de modèles qui sont classiquement utilisés dans la littérature pour modéliser des motifs courts ou des domaines. Étant donnés un ensemble d'occurrences d'un motif particulier, ces modèles peuvent être utilisés pour modéliser ces différentes occurrences et identifier de nouvelles occurrences de la même famille. La section 1.4 s'intéresse à la problématique centrale de cette thèse, c'est-à-dire la découverte d'un nouveau motif.

1.3.1 Expressions régulières

Les expressions régulières sont des modèles issus de la théorie des langages formels. La modélisation d'un ensemble de séquences par une expression régulière consiste à représenter le consensus de l'alignement des séquences en respectant une syntaxe prédéfinie. Cette syntaxe est celle communément utilisée en informatique modulo certaines règles spécifiques aux séquences biologiques. Les expressions régulières sont des concepts mathématiques relativement simples. En effet, la lecture d'une expression régulière rend immédiatement compte des positions clés des séquences, et en particulier des propriétés physico-chimiques associées à ces positions. Dans le cadre d'une modélisation par expression régulière, le processus d'identification permettant d'associer une nouvelle séquence à la famille étudiée consiste à parcourir cette séquence et tester chaque sous-séquence pour vérifier si elle respecte ou non l'expression régulière.

1.3.2 Matrices pondérées

En bio-informatique les matrices sont couramment utilisées pour décrire un modèle de séquences biologiques. Nous présenterons ici trois modèles : PFM (*Position Frequency Matrix*), PPM (*Position Probability Matrix*) et PWM (*Position Weight Matrix*). Les PWM sont aussi communément appelés PSSM (*Position Specific Score Matrix*). Ces modèles ont été introduit initialement par Gary Stormo [SSGE82] pour remplacer l'utilisation des séquences consensus qui manquent de subtilité dans la modélisation. Gary Stormo a notamment utilisé ces modèles pour la modélisation de sites de fixation de l'ADN [Sto00]. Ces modèles se représentent par des matrices de taille $|\Sigma| \times L$ où $|\Sigma|$ est le nombre de symboles différents dans l'alphabet des séquences et L le nombre de caractères qui composent le motif.

À partir d'un ensemble X de N séquences alignées de taille L, les éléments de

la matrice M d'un PFM sont calculés de la manière suivante :

$$M_{i,j} = \sum_{x \in X} I(x_j = \Sigma_i), \tag{1.11}$$

avec $i \in [1 : |\Sigma|]$, $j \in [1 : L]$, Σ_i est un élément de l'alphabet Σ et la fonction I est une fonction indicatrice définie par :

$$I(a, \Sigma_i) = \begin{cases} 1 & \text{si } a = \Sigma_i \\ 0 & \textit{sinon.} \end{cases} \tag{1.12}$$

Une fois que nous avons créé le PFM, nous pouvons en déduire un PPM en divisant chaque valeur par le nombre total de séquences. Chaque colonne du PPM définit donc une distribution de probabilité sur Σ. L'expression de $M_{i,j}$ devient alors :

$$M_{i,j} = \frac{1}{N} \times \sum_{x \in X} I(x_j = \Sigma_i). \tag{1.13}$$

Dans un PPM, chaque position est supposée indépendante : la probabilité d'apparition d'un élément à une position donnée ne dépend ni des positions suivantes, ni des positions précédentes. Nous pouvons donc calculer la probabilité P_{PPM} qu'une séquence x soit générée à partir du PPM M de la manière suivante :

$$P_{PPM}(x|M) = \prod_{j=1}^{L} M_{r(x_j),j}, \tag{1.14}$$

avec $r(x_j)$ l'indice du caractère x_j dans Σ.

À partir du PPM précédent, nous pouvons maintenant construire un PWM. Pour convertir un PPM en PWM, il faut calculer *l'odds ratio* de chaque élément du PPM. Pour cela, il est nécessaire de définir un modèle nul b (*background*) exprimant la distribution a priori des éléments de l'alphabet. Un modèle simple est de considérer que tous les symboles de Σ sont équiprobables. Cependant, en pratique nous utiliserons plutôt les probabilités observées chez l'organisme étudié. Les éléments du PWM se calculent alors de la manière suivante :

$$M_{i,j} = \log\left(\frac{\frac{1}{N} \times \sum_{x \in X} I(x_j = \Sigma_i)}{b_i}\right), \tag{1.15}$$

avec b_i la probabilité a priori associée au symbole Σ_i. De la même manière que dans un PPM, dans un PWM la probabilité de chaque position est supposée indépendante. Le score S_{PWM} d'une séquence x pour un PWM s'obtient de la manière suivante :

$$S_{PWM}(x|M) = \sum_{j=1}^{L} M_{r(x_j),j}. \tag{1.16}$$

Une séquence aléatoire, dont la probabilité est plus élevée dans le modèle nul aura donc un score négatif. Inversement, une séquence ayant une probabilité plus élevée dans le PPM aura un score positif.

Afin d'éviter d'obtenir une probabilité nulle dans notre PPM (et donc un score de PWM de $-\infty$), il est souvent nécessaire d'appliquer une correction au préalable sur le PFM pour pouvoir calculer la probabilité d'une nouvelle séquence. Une correction simple est d'ajouter un pseudo-compte (ou estimateur de Laplace) à chaque élément de M dans le PFM. Ajouter un pseudo-compte revient à ajouter de fausses séquences dans notre alignement qui, à partir de nos connaissances des séquences protéiques et nucléiques, permettent de modéliser simplement tous les événements qui pourraient se produire. Cependant cette correction reste rudimentaire. Afin d'inclure une meilleure information a priori, il existe une méthode plus sophistiquée, que nous ne détaillerons pas ici, basée sur l'utilisation de modèles de Dirichlet [BHK$^+$93].

Pour déterminer si une nouvelle séquence appartient à la famille de séquence utilisée pour générer ces modèles, nous devons maintenant évaluer la significativité de ces scores. Pour cela, la méthode standard consiste à estimer la distribution de scores de séquences aléatoires, et d'en déduire une p-valeur correspondant à la probabilité d'obtenir un score aussi bon que le score observé avec une séquence aléatoire. Si la p-valeur est inférieur à un certain seuil, usuellement 5% ou 1%, nous pourrons en conclure qu'il est peu probable d'obtenir une séquence aussi bonne par le modèle aléatoire, et donc que, vraisemblablement, cette nouvelle séquence appartient à la famille en question. Toute la difficulté de cette approche est d'avoir une bonne estimation de la distribution de séquences aléatoires, et donc de choisir un modèle aléatoire approprié. Nous pouvons citer par exemple les méthodes décrites dans les références [ZJL$^+$07] et [TV07] qui utilisent des modèles de Markov d'ordre 1 pour évaluer un modèle aléatoire. Notons qu'il existe d'autres approches plus sophistiquées pour estimer les p-valeurs. Mais elles sont rarement utilisées en pratique du fait de la complexité de calcul. En effet, estimer la p-valeur à partir d'une PWM est un problème NP difficile [TV07, ZJL$^+$07].

Exemple Considérons un exemple de six séquences d'ADN : $X = \{$TACGAT, TATAAT, TATAAT, GATACT, TATGAT, TATGTT$\}$. En reprenant les définitions précédentes, nous pouvons construire le PFM (Figure 1.6). À partir de ce PFM, nous pouvons facilement déduire différentes règles caractérisant la famille de séquences étudiées comme par exemple le fait que la position 2 est nécessairement un A, ou encore que la position 4 est soit un A soit un G. Nous pouvons ensuite construire le PPM correspondant en divisant chaque élément du PFM par $|X|$ (Figure 1.7). Nous pouvons alors en déduire que la probabilité de voir la lettre A en $5^{\text{ème}}$ position est d'environ 67%. Nous pouvons voir également que sans correction

	1	2	3	4	5	6
A	0	6	0	3	4	0
C	0	0	1	0	1	0
G	1	0	0	3	0	0
T	5	0	5	0	1	6

FIGURE 1.6 – Exemple PFM
PFM obtenu avec l'ensemble de séquences $X = \{$TACGAT, TATAAT, TATAAT, GATACT, TATGAT, TATGTT$\}$.

	1	2	3	4	5	6
A	0	1	0	0.5	0.67	0
C	0	0	0.17	0	0.17	0
G	0.17	0	0	0.5	0	0
T	0.84	0	0.84	0	0.17	1

FIGURE 1.7 – Exemple PPM
PPM obtenu à partir du PFM de la Figure 1.6.

du PFM, toute séquence avec un T, un C, ou un G en 2ème position aura une probabilité nulle. Pour construire le PWM, nous considérons une distribution a priori équiprobable des lettres donc $b_A = b_C = b_G = b_T = 0.25$ (Figure 1.8). En utilisant ces modèles, nous pouvons maintenant évaluer le score (ou la probabilité) de différentes séquences et sous l'hypothèse d'une distribution de score, nous pourrons identifier les séquences homologues à la famille X.

	1	2	3	4	5	6
A	$-\infty$	2	$-\infty$	1	1.42	$-\infty$
C	$-\infty$	$-\infty$	-0.58	$-\infty$	-0.58	$-\infty$
G	-0.58	$-\infty$	$-\infty$	1	$-\infty$	$-\infty$
T	1.74	$-\infty$	1.74	0	-0.58	2

FIGURE 1.8 – Exemple PWM
PWM obtenu à partir du PPM de la Figure 1.7 en considérant un modèle nul uniforme et aucune correction.

séquence	S_{PFM}	P_{PPM}	S_{PWM}
TACGAT	25	4.78×10^{-2}	7.58
TATAAT	29	2.36×10^{-1}	9.9
GATACT	22	1.21×10^{-2}	5.58
TATGAT	29	2.36×10^{-1}	9.9
TATGCT	26	6.00×10^{-2}	7.9
TTTGCT	20	0	$-\infty$

FIGURE 1.9 – Exemple de scores
Scores de différentes séquences avec les différents modèles des Figures 1.6, 1.7 et 1.8

1.3.3 HMM : modèle de Markov caché

Un inconvénient majeur des PMW présentés précédemment est que l'on ne peut pas modéliser des *gaps* de tailles variables. Pour gérer cela, il existe d'autres modèles plus appropriés comme notamment les HMM. Un HMM (*Hidden Markov Model*, ou en français modèle de Markov caché) est, comme les PPM, un modèle probabiliste d'une famille de séquences. Les HMM ont été introduit initialement dans les années 60-70 par Baum et ses collaborateurs [BP, BE, BS, BPSW, Bau]. Les HMM sont particulièrement connus pour leur application dans la reconnaissance de la parole, de l'écriture manuscrite, ... et en bio-informatique. Les applications classiques des HMM peuvent se diviser en deux catégories : les problèmes de classification et les problèmes de segmentation. Les problèmes de classification sont nombreux, on trouve par exemple l'identification d'une famille de protéines à partir d'une séquence d'acides aminés [HKMS93]. Pour ces problèmes, on manipule généralement un ensemble de HMM, un pour chaque classe à reconnaître. La résolution des problèmes de classification consiste à calculer la probabilité de génération d'une séquence par chacun des HMM de la bibliothèque et d'assigner cette séquence au modèle le plus probable. Pour les problèmes de segmentation, on trouve des problèmes tels que la localisation des régions codantes et non codantes d'une chaine de nucléotides [KBM$^+$94] ou la recherche des îlots CpG [Bir87]. Ces problèmes de segmentation reposent sur la recherche du chemin ayant la probabilité maximale de générer cette séquence et seront plus amplement développés dans la section 1.3.3.3.

Dans les sections suivantes, nous commencerons par présenter le formalisme général d'un HMM. Nous verrons ensuite comment entraîner et utiliser les HMM pour modéliser une famille de séquences.

1.3.3.1 Définition d'un HMM

Un HMM peut s'apparenter à un automate probabiliste [Cas90]. Il est définit par une structure composée d'états et de transitions, et par un ensemble de probabilités sur les transitions. La différence essentielle est que pour un automate probabiliste, la génération d'un élément s'effectue lorsqu'une transition est empruntée tandis que pour un HMM, la génération s'effectue lorsqu'un état est emprunté. De plus, dans un HMM on associe à un état non pas un élément mais à une distribution de probabilité de générer chaque élément. Précisément, un HMM est un automate probabiliste caractérisé par deux processus stochastiques : un premier processus interne non observable (d'où le *hidden*) du déplacement d'état en état en respectant les transitions autorisées par la topologie du modèle ; et un second processus externe observable de la génération d'une observation dans chaque état du HMM.

Formellement, un HMM à N états est défini par un quadruplet (Σ, E, T, G) avec :

- Σ un alphabet fini de symboles (par exemple {A,T,G,C} pour les séquences nucléiques) ;
- E l'ensemble des états $\{e_0, e_2, \ldots, e_{N+1}\}$. Deux de ces états sont dits «muets», c'est-à-dire qu'ils ne génèrent aucun symbole et n'ont donc pas de probabilités de génération associées. Ce sont deux états spéciaux, *start* et *end*, qui servent respectivement à débuter et conclure une séquence ;
- T une matrice $|E| \times |E|$ indiquant les probabilités de transition entre les états : on note $T(e_i, e_j)$ la probabilité de transition de l'état e_i à l'état e_j ;
- G une matrice $|E| \times |\Sigma|$ indiquant les probabilité de génération associées aux états : on note $G(a, e_i)$, avec $a \in \Sigma$, la probabilité de générer le symbole a dans l'état e_i. On a une distribution de probabilités sur les symboles dans chaque état, et donc : $\forall e \in E : \sum_{a \in \Sigma} G(a, e) = 1$.

La définition présentée ici n'est pas la définition originale proposée par Baum [BPSW]. Elle diffère par l'introduction des états muets *start* et *end*. Cependant, cette définition est couramment utilisée dans la plupart des applications, dont la modélisation des séquences biologiques.

1.3.3.2 Probabilité de génération d'une séquence

Pour calculer la probabilité de générer une séquence de symbole $x = x_1 x_2 \ldots x_{|x|}$ à l'aide du HMM H, on doit calculer la probabilité de génération de x à travers tous les chemins (ou séquence d'états) possibles de H et faire la somme de ces probabilités. La probabilité de générer la séquence x par le chemin $c = e_0 e_2 \ldots e_{|x|+1}$, où e_0 correspond au *start* et $e_{|x|+1}$ correspond au *end*, est définie de la manière

suivante :

$$P(x,c) = T(e_0,e_1)\prod_{i=1}^{|x|} G(x_i,e_i)\times T(e_i,e_{i+1}). \tag{1.17}$$

La probabilité de générer la séquence x avec le HMM H est alors obtenue en faisant la somme sur l'ensemble des chemins possibles, notée $\mathcal{C}$:

$$P(x|H) = \sum_{c\in\mathcal{C}} P(x,c). \tag{1.18}$$

Nous pouvons maintenant calculer la probabilité de génération d'une séquence par un HMM. Cependant, cette solution n'est pas applicable en pratique du fait de sa complexité car si N est le nombre d'états du HMM, alors le nombre de chemins possibles pour générer une séquence de longueur L est de l'ordre de N^L dans le pire des cas. Étant donné que le nombre d'opérations nécessaires au calcul d'un chemin est de l'ordre de L, nous pouvons évaluer la complexité du calcul (1.18) comme étant en $\mathcal{O}(LN^L)$. À titre d'exemple, en prenant seulement une séquence de 100 symboles et un HMM à 10 états, on obtient déjà 10^{102} opérations nécessaires. Heureusement, pour résoudre ce problème, il existe une procédure de programmation dynamique bien plus efficace : l'algorithme *forward-backward* [Rab89].

On considère la variable *forward* α définie par :

$$\alpha_i(e) = P(x_1\dots x_i, e_i = e), \tag{1.19}$$

où $\alpha_i(e)$ exprime la probabilité d'avoir généré la séquence $x_1\dots x_i$ en partant de l'état *start* et d'être arrivé à l'état e pour générer le i-ème symbole. Ce calcul peut être effectué récursivement, voir Algorithme 2. La complexité de cet algorithme est de l'ordre de $\mathcal{O}(N^2L)$. En reprenant l'application numérique précédente ($N = 5$ et $L = 100$), on obtient alors seulement 2 500 opérations nécessaires pour le calcul de $P(S|H)$.

De la même manière, il est possible d'effectuer ce calcul « à l'envers », on parle alors de l'algorithme *backward*. On considère alors la variable *backward* β définie par :

$$\beta_i(e) = P(x_{i+1}\dots x_{|x|}, e_i = e|H) \tag{1.20}$$

où $\beta_i(e)$ exprime la probabilité de générer la séquence $x_{i+1}\dots x_{|x|}$ en partant de l'état e et en arrivant sur l'état *end*. Ce calcul peut également être effectué récursivement de façon analogue à l'algorithme *forward*. Sa complexité est identique à l'algorithme *forward*, soit $\mathcal{O}(N^2L)$.

1.3.3.3 Recherche du chemin de probabilité maximale

Un problème classique lorsque l'on travaille avec des HMM consiste à trouver la séquence d'états du HMM ayant la probabilité maximale de générer une

Algorithme 2 Algorithme *forward*.

Entrée: séquence x, HMM H
Sortie: $P(x|H)$
  pour tout $e \in E$ **faire**
    $\alpha_1(e) = T(e_0, e)G(x_1, e)$
  fin pour
  pour i de 2 à $|x|$ **faire**
    pour tout $e \in E$ **faire**
      $$\alpha_i(e) = \left(\sum_{f \in E} \alpha_{i-1}(f) \times T(f, e) \right) G(x_i, e)$$
    fin pour
  fin pour
  $$P(x|H) = \sum_{e \in E} \alpha_{|x|}(e)T(e, e_{N+1})$$

séquence donnée. Ce qui nous intéresse ici n'est pas tant la valeur de la probabilité maximale mais le chemin qui permet d'obtenir cette probabilité. On appelle ce chemin le chemin de Viterbi. Pour rechercher ce chemin, nous avons besoin d'un algorithme efficace car une recherche exhaustive de tous les chemins possibles pour identifier celui ayant la probabilité la plus élevée souffre des mêmes limites combinatoires que dans le cas du calcul de la probabilité de génération d'une séquence. Pour résoudre ce problème, il existe un autre algorithme de programmation dynamique, l'algorithme de Viterbi [Rab89]. Cet algorithme est assez proche de l'algorithme *forward-backward*. La principale différence résulte de la maximisation des probabilités attachées aux états précédents au lieu du calcul de la somme de ces probabilités. La complexité de l'algorithme de Viterbi est identique à la complexité de l'algorithme *forward-backward*, soit $\mathcal{O}(N^2L)$.

1.3.3.4 Apprentissage d'un HMM

Jusqu'à présent nous avons vu les différents algorithmes utilisés pour résoudre les problèmes de classification et de segmentation. Cependant, nous partions du principe que nous disposions déjà d'un HMM construit et paramétré de manière à modéliser une famille de séquence. Dans le cas le plus favorable, le HMM recherché peut être construit directement à partir des connaissances a priori sur le sujet. C'est notamment le cas dans la référence [KBM+94], dans laquelle les auteurs modélisent des séquences d'ADN de l'espèce *E. coli*. Ils exploitent un certain nombre de connaissances pour apprendre la structure (nombre d'états, chemins autorisés) du HMM et les distributions de probabilités pour segmenter les séquences d'ADN en région codante et non-codante. Malheureusement, il est relativement rare de

disposer de suffisamment de connaissances pour apprendre un HMM de cette manière. Pour cela, il existe des algorithmes d'apprentissage que l'on applique à un ensemble de séquences représentatives des séquences que l'on souhaite modéliser.

On peut différencier deux cas de figure d'apprentissage d'un HMM, lorsque la structure est connue ou non. Lorsque la structure est connue, le problème consiste à «entraîner» le HMM. C'est-à-dire ajuster les probabilités de génération et de transition de manière à expliquer au mieux les séquences d'apprentissage. Ce problème est NP-difficile [AW92] mais dispose d'heuristiques classiques telles que l'entraînement de Viterbi et l'entraînement de Baum-Welch, tous deux de complexité $\mathcal{O}(KN^2T)$, avec N le nombre d'états, T la taille totale des séquences d'apprentissage et K le nombre d'itérations de l'algorithme. L'entraînement de Baum-Welch est issu de la méthode générale d'*Expectation maximisation* [DLR77] qui est une procédure d'apprentissage permettant de maximiser la vraisemblance des séquences d'apprentissage lorsqu'il y a des variables cachées dans le modèle.

Lorsque la structure est inconnue, le problème d'apprentissage est plus difficile puisqu'en plus de paramétrer la structure, il faut également déduire cette structure des séquences d'apprentissage. Plusieurs approches ont été proposées pour résoudre ce problème. Nous pouvons notamment citer l'approche par généralisation (ou fusion) d'états dont le principe est de construire le HMM le plus spécifique puis de le généraliser par des étapes successives de fusion d'états [SO94a]. Une autre approche consiste à la spécialisation (ou fission) d'états dont le but est de spécialiser un HMM très général en scindant successivement des états ou transitions [TS92].

1.3.3.5 Les HMM profils

Il existe une spécialisation des HMM dédiée à l'étude des séquences biologiques : les HMM profils. Ces modèles sont couramment utilisés pour la modélisation de familles de séquences et la recherche d'homologie. Les HMM profils font maintenant partie des outils classiques de la bio-informatique [Edd95, DREKJM98].

Les HMM profils sont souvent utilisés pour modéliser les propriétés révélées par un alignement multiple : les positions conservées et les positions ayant une forte probabilité d'engendrer une insertion ou une délétion. Les HMM profils permettent donc d'obtenir un modèle probabiliste permettant d'intégrer la totalité des informations d'un alignement multiple. Pour cela, à chaque position p de l'alignement correspond une position p dans le HMM profil qui contient trois états (voir Figure 1.10) :

— un état *Match*, noté M_p, qui contient la distribution de probabilité de génération de chaque élément de Σ à la position p ;
— un état *Insert*, noté I_p, qui modélise l'insertion éventuelle d'un élément à la position p. La probabilité d'insertion se traduit par la probabilité de la transition $T(M_p, I_p)$. L'insertion de plusieurs éléments se traduit par une

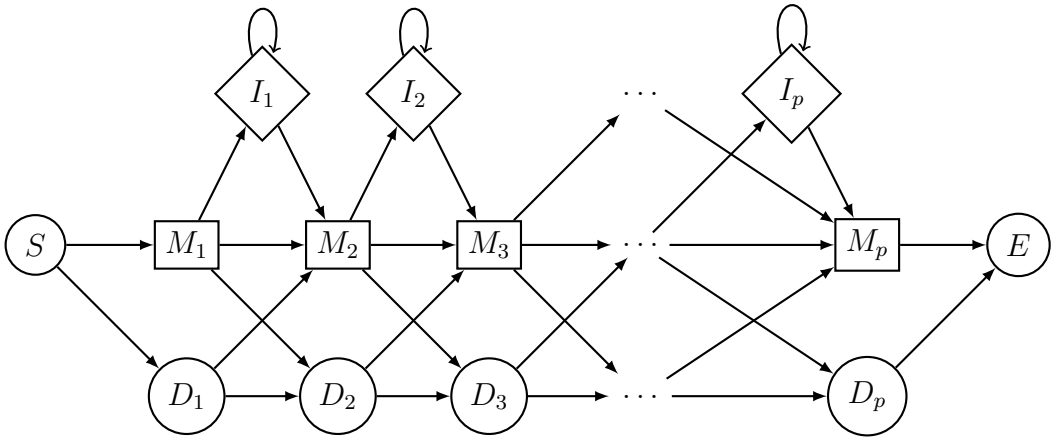


FIGURE 1.10 – Structure d'un HMM profil
Les états S et E correspondent respectivement aux états *start* et *end*. Les états M correspondent aux états *Match*. Les états I et D correspondent respectivement aux états d'insertion et de délétion.

boucle $T(I_p, I_p)$. Une fois les insertions terminées, on oblige la modélisation à reprendre dans l'état M_{p+1} par une transition $T(I_p, M_{p+1})$;

— un état *Delete*, noté D_p, qui représente la délétion éventuelle d'un élément à la position p. Les états de délétion sont muets, c'est à dire qu'ils ne peuvent pas générer de symbole. L'état de délétion permet donc de «contourner» un état *Match* pour modéliser une séquence ayant subie une délétion. Le coût d'ouverture de la délétion se traduit par la probabilité de transition de l'état *Match* précédent vers l'état de délétion, notée $T(M_{p-1}, D_p)$. Le coût d'extension de la zone de délétion se définit par les probabilités de transition vers l'état de délétion, ou *Match*, suivants $T(D_p, D_{p+1})$ et $T(D_p, M_{p+1})$.

Cette définition n'est pas exactement celle proposée par [HKMS93, KBM+94] car elle ne considère pas les transitions entre les états *Delete* et les états *Insert*. Cependant, il s'agit de la définition utilisée par la majorité des base de données de domaines protéiques et par la structure manipulée par le logiciel HMMER (voir plus bas).

Du fait de leur similarité avec les alignements multiples, il est à noter que, la plupart du temps, les HMM profils sont appris à partir d'un alignement multiple préalablement optimisé par les outils dédiés (MUSCLE, ...) plutôt qu'en utilisant les approches du type EM évoquées plus haut [Edd03]. Pour entraîner un nouveau HMM profil à partir d'un alignement multiple de séquences, on commence par identifier toutes les positions conservées dans l'alignement (celles avec une majorité de non gaps). Chacune des positions conservées deviendra alors un état *match* du

HMM profil. Ensuite, les probabilités de transition entre les états sont estimées à partir de l'alignement. La dernière étape consiste à appliquer un algorithme de lissage des probabilités afin d'attribuer une probabilité faible mais non nulle aux différentes émissions du modèle qui n'ont jamais été observés dans l'alignement. Cela permet d'éviter des erreurs de prédictions dues au manque d'information dans l'alignement fourni lors de l'apprentissage. Par exemple, si dans une nouvelle séquence un acide aminé n'a jamais été observé à une position donnée, alors la probabilité que le modèle puisse générer cette séquence sera toujours nulle. Ce lissage peut être effectué en utilisant simplement un pseudo-compte ou en utilisant des mixtures de Dirichlet [BHK+93, SKB+96]. C'est de cette manière que sont par exemple entraînés les HMM profils dans la suite HMMER [Edd98].

HMMER est un logiciel qui offre de nombreuses fonctionnalités liées à la manipulation des HMM profils pour l'analyse de séquences. HMMER a été publié par Sean Eddy. Avec ce logiciel, on peut notamment construire un HMM profil à partir d'un alignement multiple, aligner une séquence sur un HMM pour résoudre un problème de segmentation ou encore rechercher des séquences homologues à un HMM pour résoudre un problème de classification.

1.4 Découverte de nouveaux motifs

On a vu dans la section précédente différents modèles qui peuvent être utilisés pour modéliser des motifs ou des domaines de grande taille. À partir d'un ensemble d'occurrences de ces motifs, on a vu qu'il était possible d'apprendre un modèle qui peut ensuite être utilisé pour identifier de nouvelles occurrences de la même famille. La question à laquelle on s'intéresse ici, et qui est le sujet central de cette thèse, est celui de la découverte de nouveaux motifs sans occurrences connues.

Étant donné un ensemble de séquences, le problème qui se pose est de découvrir un motif ayant un nombre d'occurrence élevé parmi les séquences. On a vu à la section 1.2.2.3, qu'étant données deux séquences, les algorithmes d'alignement locaux tel que BLAST, permettent d'identifier une sous-séquence conservée par les deux séquences. Lorsqu'on a plus de deux séquences par contre, il faut se tourner vers d'autres types de méthodes pour identifier les parties conservées.

Dans la littérature il existe de très nombreuses méthodes de découverte de nouveaux motifs. En effet, suivant les données biologiques que l'on traite, la problématique n'est pas toujours la même et il est nécessaire d'adapter la méthodologie. Nous pouvons distinguer différentes caractéristiques qui peuvent varier d'un problème à l'autre :

— de combien de séquences est ce que l'on dispose ?
— quelle est la fréquence du motif recherché ? est-il présent strictement une fois dans chaque séquence ? peut-il apparaître plusieurs fois ? peut-il être

absent de certaines séquences ?
— est ce que l'on dispose de séquences négatives ?
— comment évaluer la pertinence d'un motif ?

Les problèmes peuvent être alors fondamentalement très différents. Nous n'allons donc pas énumérer les méthodes pour chaque combinaison de caractéristique. Cependant, les méthodes de découverte peuvent être classées suivant l'approche computationnelle utilisée. Certaines méthodes utilisent des approches d'algorithmique du texte en énumérant de manière exhaustive tous les mots possibles et en comparant les fréquences d'apparition. D'autres méthodes en revanche utilisent des modèles probabilistes de la séquence où les paramètres du modèle sont estimés en utilisant le principe de maximum de vraisemblance, ou une inférence Bayésienne.

Les méthodes provenant de l'algorithmique du texte garantissent dans certains cas l'optimalité globale de la solution en énumérant de manière exhaustive toutes les solutions possibles. Cette exploration peut être possible pour certains problèmes grâce à l'utilisation de structures de données appropriées comme les arbres des suffixes [Sag98]. Ces méthodes sont donc plutôt appropriées pour trouver des motifs courts et dont chaque élément du motif correspond à la présence d'un nucléotide (ou acide aminé) particulier. Cependant, ces méthodes se révèlent rapidement perfectibles lorsque le motif recherché comprend des insertions ou des positions faiblement contraintes. C'est notamment le cas des sites de fixation des facteurs de transcription. Dans ce cas, il sera nécessaire de recourir à un post-traitement des résultats en utilisant par exemple une méthode de *clustering* afin de former des motifs complexes à partir des résultats de la recherche.

Les méthodes probabilistes reposent sur l'utilisation d'un modèle probabiliste comme un PWM par exemple. Ces méthodes ont été développées pour rechercher la présence de motif plus long ou plus complexe que les approches basées sur l'algorithmique du texte. En revanche, ces méthodes ne garantissent généralement pas d'obtenir l'optimal global du fait que le problème d'optimisation est la plupart du temps NP difficile. Nous allons présenter ci-dessous trois méthodes classiques de la bio-informatique qui sont *Oligo Analysis*, le *Gibbs sampler* et *MEME*.

1.4.1 Oligo Analysis

Oligo Analysis est un algorithme de découverte de motif développé par Jacques van Helden *et al.* [vHACV98]. Cette méthode utilise les principes de l'algorithmique du texte. Cet algorithme fait partie de la suite logiciel RSAT [NCMCM$^+$18] dédiée à la détection et l'analyse des éléments régulateurs des génomes. L'algorithme consiste à énumérer tous les motifs d'une taille donnée présents dans un ensemble de séquences (habituellement nommée «séquences positives»). Il évalue ensuite si la fréquence d'apparition d'un motif particulier est plus élevée qu'attendu par hasard ou qu'observée dans un autre ensemble de séquences (habituellement nommée

«séquences négatives»). Cet algorithme a ensuite été étendu pour ajouter la possibilité d'intégrer des gaps (positions non conservées) dans les motifs [vHRCV00]. Un inconvénient majeur de cet algorithme est le fait qu'il ne considère que des motifs fortement contraints (chaque position est attribuée à un seul nucléotide) et ne permet donc pas de considérer des motifs plus complexes comme des expressions régulières ou des PWM. La conséquence de cela est donc un manque de sensibilité ou la production de nombreux artéfacts. Un post-traitement de ces résultats est donc souvent recommandé. Pour autant, cet algorithme est à la base de très nombreux autres algorithmes permettant de compenser cela. Nous pouvons citer par exemple la référence [Tom99] qui utilise des chaines de Markov pour estimer les probabilités d'apparition des motifs dans un ensemble de séquences négatives, et sa suite l'algorithme YMF (*Yeast Motif Finder*) [ST00] ou encore la référence [BJVU98] qui intègre l'utilisation des expressions régulières.

1.4.2 Gibbs Sampler

Le *Gibbs Sampler* (ou *Échantillonage de Gibbs*) est un algorithme probabiliste de type MCMC (chaîne de Markov Monte-Carlo) de découverte de motifs proposé par Laurence *et al.* [LAB+93]. Cet algorithme permet de trouver un motif de longueur prédéfinie k le plus «similaire» dans un ensemble de N séquences. Cela suppose donc de connaître au préalable la longueur k du motif que l'on cherche mais surtout que chaque séquence contienne exactement une occurrence du motif que l'on cherche. Le principe de cet algorithme est d'échantillonner aléatoirement une sous-séquence x_i de longueur k dans chaque séquence i et les aligner pour construire une PWM. Ensuite pour chacune des N séquences, on calcule le score de cette PWM à chaque position de la séquence i et on choisit une nouvelle sous-séquence en tirant au «hasard» suivant la distribution des scores de la PWM (les positions avec les meilleurs scores auront plus de chance d'être choisies). Cette nouvelle sous-séquence x_i remplace celle précédemment choisie pour la séquence i. L'opération est ensuite répétée jusqu'à ce que les sous-séquences de la PWM soient stables. L'algorithme est probabiliste et peut donc être répété plusieurs fois afin d'essayer de trouver le motif optimal. Il existe plusieurs optimisations de cette algorithme. Nous pouvons citer par exemple *AlignACE* (***Align****s Nucleic* ***A****cid* ***C****onserved* ***E****lements*) qui est spécialisé pour la découverte de motifs conservés dans un ensemble de séquences d'ADN [RHEC98]. Cette nouvelle version permet notamment de proposer non pas un mais plusieurs motifs en masquant les motifs déjà identifiés de manière itérative.

1.4.3 MEME

MEME (***M**ultiple **E**M for **M**otif **E**licitation*) est un algorithme développé par Bailey et Elkan [BE95] basé sur un algorithme EM (*Expectation Maximization*). La stratégie de cet algorithme est d'utiliser les k-mers qui apparaissent dans l'ensemble de séquences au lieu et place des séquences elles-mêmes comme point de départ. Cette stratégie permet notamment de relâcher la contrainte que chaque séquence contient exactement une occurrence du motif recherché. Afin de modéliser les k-mers qui ne seraient pas une instance du motif recherché MEME utilise un modèle de Markov d'ordre 0 de séquences aléatoires. Pour ne plus utiliser un modèle background, Redhead et Bailey ont développé une suite nommée DEME (***D**iscriminatively **E**nhanced **M**otif **E**licitation*) [RB07]. DEME utilise alors un ensemble de séquences *négatives* fournies par l'utilisateur pour découvrir une PWM qui permet de discriminer les deux ensembles de séquences. L'algorithme de DEME ne repose plus sur l'algorithme EM mais sur la méthode du gradient conjugué.

1.5 Segmentation

Lorsque l'on étudie une séquence biologique (ADN ou protéine) il est fréquent de rechercher la présence de motifs ou domaines appartenant à une famille connue dans cette séquence. Pour cela, nous disposons généralement d'un modèle (PWM ou autres) et en utilisant ce modèle, il est alors possible d'attribuer un score à chaque position de la séquence pour déterminer si le motif est présent ou pas. Un autre type d'analyses possibles d'une séquence est de chercher à identifier dans cette séquence des régions ayant une composition homogène. On parle alors de problème de segmentation. La segmentation est une méthode d'analyse qui consiste à diviser une séquence en segments discrets dans le but de révéler les propriétés sous-jacentes de la séquence. Il existe de nombreuse applications de la segmentation dans les problématiques de bio-informatique. Par exemple, les îlots CpG, des régions de l'ADN enrichies en dinucléotide CG [Bir87, GGF87], sont souvent identifiés à l'aide d'algorithmes de segmentation basés sur une fenêtre coulissante [WL04, TJ02, PM02, RLB00]. Une autre problématique est la segmentation des données de microarrays d'hybridation en génomique comparative. On trouvera une introduction à cette problématique et aux solutions proposées dans la thèse de Franck Picard [Pic05]. Un autre problème de segmentation en bio-informatique est la différenciation de séquences codantes et non codantes dans l'ADN. Les méthodes de segmentation tirent parti du fait que les compositions nucléotidiques de ces deux types de séquences sont habituellement très différentes [LB98]. Pour ce type de problème, les HMM présentés plus haut sont souvent utilisés, et la segmentation est réalisée via l'algorithme de Viterbi.

En dehors des HMM, on pourra distinguer deux grands types de méthodes de segmentation. En effet, la segmentation peut être vue comme la recherche de point de rupture dans la séquence, c'est-à-dire un changement brutal de caractéristiques comme la moyenne ou la variance des valeurs locales. Mais la segmentation peut également être assimilée à une problématique de reconstruction du signal en utilisant une représentation plus simple comme la régression linéaire par morceaux afin de faire émerger des caractéristiques locales de la séquence. Il existe également d'autres approches que nous ne détaillerons pas mais qui sont très utilisées en traitement du signal, comme les transformées de Fourier [AFS93, KCPM01], ou la décomposition en ondelettes [CF99, CGKC11].

1.5.1 Détection de point de rupture

La détection de point de rupture est un problème classique en analyse du signal qui correspond à plusieurs problèmes : détecter les changements de caractéristiques de la séquence, les localiser et ensuite analyser individuellement les segments obtenus.

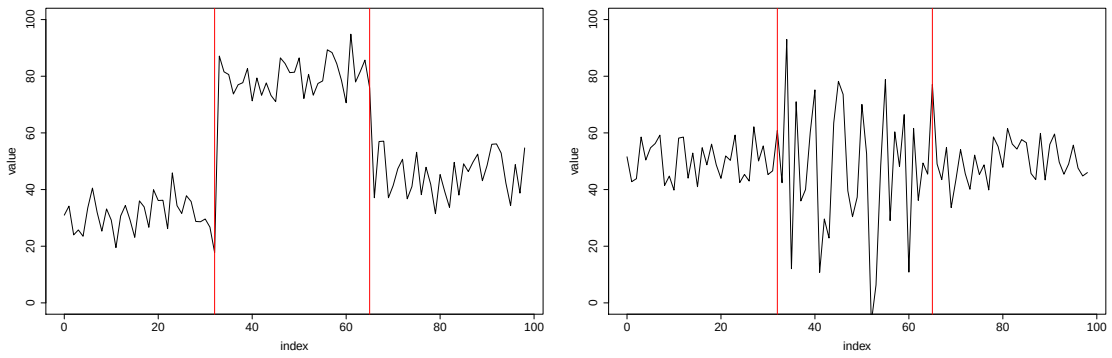
En analyse statistique, le problème de détection de point de rupture est un problème visant à estimer les instants où un signal présente des changements dans la distribution des valeurs. Classiquement, on réalise la détection de point de rupture pour un signal ayant des changements dans la moyenne. De manière plus générale, on peut s'intéresser à n'importe quelle statistique ou caractéristique calculable localement, comme la variance par exemple. Cette caractéristique est représentée par le paramètre Θ et peut être estimée localement pour chaque élément x_i d'une séquence $X = (x_1 \dots x_n)$ où $i \in [1 : n]$. On note Θ_i la valeur de la caractéristique associée à chaque x_i. La rapidité du phénomène conduit à le qualifier de rupture, c'est-à-dire que la transition d'état se fait dans un intervalle inférieur à la fréquence d'échantillonnage. Si l'on observe $\Theta_i \neq \Theta_{i+1}$ alors x_i est un point de rupture (voir exemple Figure 1.11).

Dans ce cadre, la segmentation conduit à la détermination d'un signal « constant » par morceaux. La détection de rupture se ramène alors au « débruitage » du signal. Par exemple, si on considère que le paramètre Θ est la moyenne, nous pouvons décomposer chaque variable comme :

$$X_i = \Theta_i + \epsilon_i$$

avec Θ_i la moyenne de la distribution de X_i et ϵ_i une variable aléatoire de moyenne nulle et de variance finie, semblable à du bruit.

Détecter les ruptures consiste donc à déceler la présence d'un changement brutal et le localiser. Cependant, déterminer l'existence d'une rupture est d'autant plus difficile que la rupture n'est pas forcément caractérisée par un décalage de



(a) (b)

FIGURE 1.11 – Détection de point de rupture
Dans la figure (a) le paramètre Θ est la moyenne et dans la figure (b) le paramètre Θ est la variance. Les traits rouges verticaux marquent la position du point de rupture. Dans chacun de ces deux exemples, on obtient alors trois segments homogènes suivant leur moyenne et leur variance respectivement.

grande amplitude entre Θ_i et Θ_{i+1} par rapport à la variance des observations. De même, lorsque les changements sont progressifs, la détection de rupture peut s'avérer problématique, soit parce que la méthode risque d'identifier plusieurs ruptures successives, soit la rupture sera mal localisée ou ne sera tout simplement pas détectée. La complexité du problème augmente fortement lorsque le nombre de ruptures et leurs positions sont inconnus. De même, le choix du paramètre Θ nécessite des hypothèses fortes sur la distribution des observations.

Pour plus de détails, le lecteur peut notamment se référer à [BN93] sur la détection d'une rupture unique, [CG14] qui présente l'ensemble des méthodes paramétriques, [BD93] pour une présentation des approches non paramétriques et enfin [DMS14] pour une présentation plus générale des méthodes d'analyse des séries temporelles.

1.5.2 Représentation linéaire par morceaux

Une façon simple de modéliser une série temporelle est d'utiliser une représentation linéaire par morceaux, ou PLR (*Piecewise Linear Representation*). Le principe de cette représentation est d'approximer une série temporelle de taille n en utilisant K segments, voir Figure 1.12. Cette représentation a été très fréquemment utilisée en raison du côté intuitif de la méthode et de ses nombreux avantages [LSL$^+$00, HM99, KJM95]. Entre autres problématiques la PLR a notamment été utilisée pour :

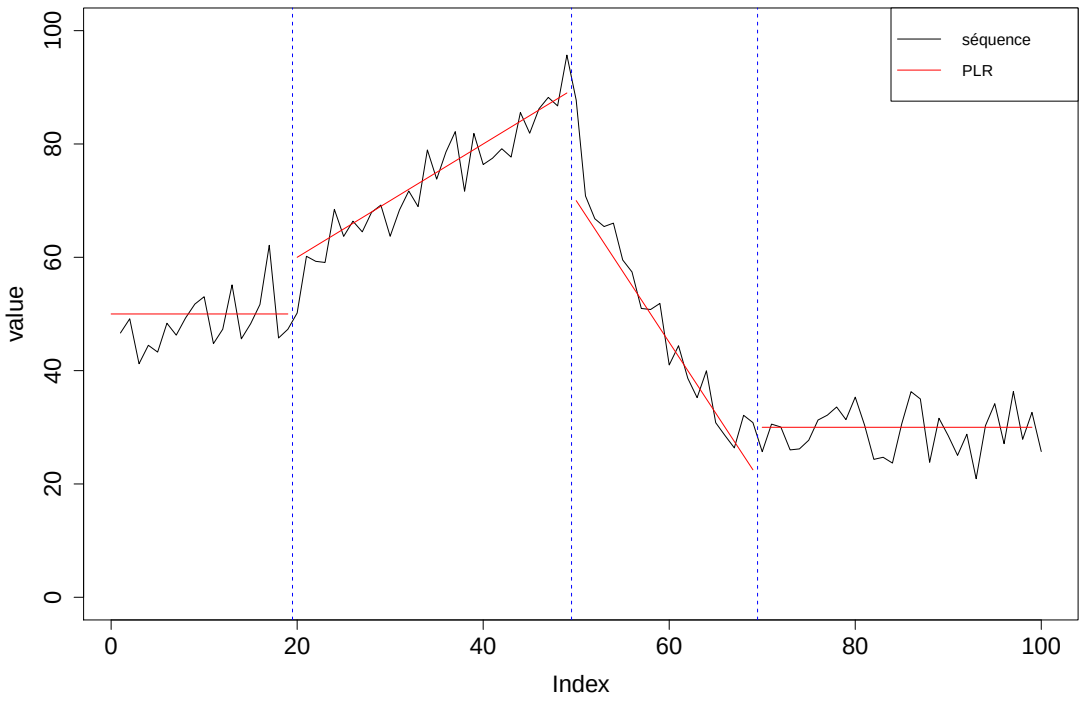


FIGURE 1.12 – Représentation linéaire par morceaux
Dans cet exemple, nous avons quatre segments ici représentés en rouge. Chaque segment correspond à une régression linéaire locale de la séquence. Les lignes bleues verticales symbolisent le changement de segment.

— la réduction de dimension pour la recherche rapide de similarité exacte [KCPM01],
— l'exploration simultanée de série temporelle [LSL+00],
— le *clustering* et la classification de séries [KP98],
— la détection de point de rupture [SO94b, GS01].

Du fait de cette multitude d'applications, un grand nombre d'approches ont été proposées pour cette problématique [Ram72, DP73, HG97]. D'un point de vue général, on peut distinguer trois classes de problèmes de PLR associés à un critère d'optimisation :

— segmenter en utilisant exactement K segments tout en minimisant l'erreur totale cumulée,
— segmenter de manière à ce que l'erreur maximale commise par chaque segment soit inférieure à un seuil prédéterminée tout en minimisant le nombre

	Critère				
Algorithme	K segments	erreur max	erreur max cumulée	Complexité	Références
fenêtre coulissante	non	oui	non	$\mathcal{O}(\frac{n}{K}n)$	[ISHS83, KJM95, QWW98] [SZ96, VVV97, WW00]
bottom-up	oui	oui	oui	$\mathcal{O}(\frac{n}{K}n)$	[HM99, KP99] [KP98, KS97]
top-down	oui	oui	oui	$\mathcal{O}(n^2K)$	[Dud73, DP73, GS01] [LSL+00, LYC98, PLC99]

TABLE 1.1 – Résumé des principaux algorithmes de segmentation en PLR

de segments utilisés,
— segmenter de manière à ce que l'erreur totale cumulée de tous les segments soit inférieure à un seuil prédéterminé tout en minimisant le nombre de segments utilisés.

De même, les différents algorithmes qui ont été développés pour résoudre ce problème peuvent être rassemblés en trois catégories d'approches qui sont :
— approche par fenêtre coulissante : un segment est développé jusqu'à ce que l'erreur commise dépasse un certain seuil,
— approche *top-down* : la série est approximée par un seul segment puis ce segment est partitionné et la procédure est répétée jusqu'au critère d'arrêt,
— approche *bottom-up* : chaque point de la série est un segment puis les segments sont fusionnés successivement jusqu'à atteindre le critère d'arrêt.

La Table 1.1 présente un résumé des algorithmes, des conditions d'utilisation, la complexité de chacun ainsi que quelques références où ce type d'algorithme a été utilisé. Cependant, le choix de l'algorithme ne dépend pas uniquement de la complexité. En effet, il n'est pas toujours facile de déterminer un bon critère de segmentation et donc le choix de celui-ci dépendra de chaque problème et des connaissances a priori du sujet. Un autre élément à prendre en compte dans le choix de l'algorithme est le fait que les données soient générées en temps réel (par exemple un électrocardiogramme) ou non. Dans le cas de données acquises en temps réel, part la définition même des algorithmes, seule la segmentation à base de fenêtre coulissante pourra être utilisée.

1.6 Régression linéaire et LASSO

Le modèle de régression linéaire est un outil statistique habituellement mis en œuvre pour l'étude de données multidimensionnelles. Le modèle se définit par une variable quantitative Y dite à «expliquer» que l'on cherche à mettre en relation avec p variables $X_1 \dots X_p$ dites «explicatives». Les données proviennent d'un échantillon de taille n. On notera x_{ij} la valeur de la j-ème variable du i-ème exemple et y_i la valeur de la variable Y pour le i-ème exemple. On dira alors que le modèle

est de régression linéaire s'il vérifie :

$$y_i = \mathrm{f}(x_1, \ldots, x_p) = \beta_0 + \sum_{j=1}^{p} (\beta_j \times x_{ij}) + \epsilon_i \tag{1.21}$$

où :

— $\beta_0 \ldots \beta_p$ sont des paramètres inconnus à estimer,

— ϵ_i est l'erreur résiduelle associée à l'exemple i.

Habituellement, les paramètres β du modèle sont estimés par minimisation de la somme quadratique des erreurs (notée SSE pour *Sum of Squarred Error*) entre y_i et sa prédiction :

$$\mathrm{SSE(f)} = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (y_i - (\beta_0 + \sum_{j=1}^{p} \beta_j \times x_{ij}))^2 \tag{1.22}$$

Dans le cas de problème en grande dimension (cas où $p \gg n$), la méthode SSE fonctionne mal car elle a tendance à faire du sur-apprentissage. Dans ce cas, il existe d'autres méthodes d'estimation qui rajoutent une contrainte sur les valeurs des β de manière à limiter ces problèmes. La méthode LASSO (***L**east **A**bsolute **S**hrinkage and **S**election **O**perator*) est une de ces approches. Cette méthode a été publié en 1996 par Robert Tibshirani [Tib96]. La méthode LASSO consiste à estimer les paramètres β en favorisant les $\beta = 0$ et donc en éliminant de fait certaines des variables prédictives. Plus précisément on cherche à résoudre le problème d'optimisation suivant :

$$\underset{\beta_0 \ldots \beta_p}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} ((y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \text{ sous la contrainte } \sum_{j=1}^{p} |\beta_j| \leqslant t \tag{1.23}$$

où t est le paramètre qui contrôle le niveau de contrainte des coefficients. Il s'agit là d'une pénalisation des coefficients β en norme l_1 qui favorise les coefficients à zéro. Outre qu'elle évite les dangers du sur-apprentissage (lorsque le paramètre de contrainte t est judicieusement choisi) le fait que cette approche mette un certain nombre de β à 0 permet de sélectionner un sous-ensemble de variables explicatives et simplifie donc l'interprétation du modèle. C'est cette dernière propriété qui nous mènera à considérer le LASSO comme une méthode de choix dans le Chapitre 6, dans la troisième partie de cette thèse.

Chapitre 2

Le paludisme

Le paludisme, ou malaria, est la maladie parasitaire la plus répandue dans le monde. En 2016, l'OMS (Organisation Mondiale de la Santé) estime qu'il y a eu 216 millions de cas de paludisme dans 106 pays. Le paludisme est responsable du décès de 445 000 personnes en 2016 dont 91% ont eu lieu en Afrique selon l'OMS. Près de la moitié de la population mondiale est exposée au risque de contracter la malaria. La grande majorité des cas et des décès dûs à cette maladie surviennent en Afrique sub-saharienne. Les personnes les plus vulnérables sont les enfants de moins de 5 ans, les femmes enceintes, les personnes atteintes du sida, les voyageurs, ... Les régions impaludées (où sévit le paludisme) sont essentiellement intertropicales, dans des niches écologiques propices à la reproduction des moustiques, voir Figure 2.1.

Le paludisme est dû à des parasites du genre *Plasmodium* transmis à l'Homme par des piqûres de moustiques *Anopheles* femelles infectées. Il existe 5 espèces de parasites responsables du paludisme chez l'Homme. Le *Plasmodium falciparum* est de loin le plus mortel. *Plasmodium falciparum* est le parasite du paludisme le plus répandu sur le continent africain. Il est responsable de la plupart des cas mortels dans le monde. *P. vivax* est le parasite prédominant hors d'Afrique.

2.1 Origines de *Plasmodium falciparum*

Les parasites responsables du paludisme sont des eucaryotes unicellulaires appartenant au genre *Plasmodium* du phylum des *Apicomplexa*. Les apicomplexes sont des parasites intracellulaires dont la majorité sont des agents phathogènes d'espèces métazoaires. Du point de vue phylogénétique, les apicomplexes font parties du genre *Chromalveolata* et plus précisément de la division *Alveolata* (voir Figure 2.2).

L'espèce *Plasmodium falciparum* responsable du paludisme chez l'Homme, a évolué à partir du genre *Laverania* (un sous-genre des *Plasmodium*) qui infectait les

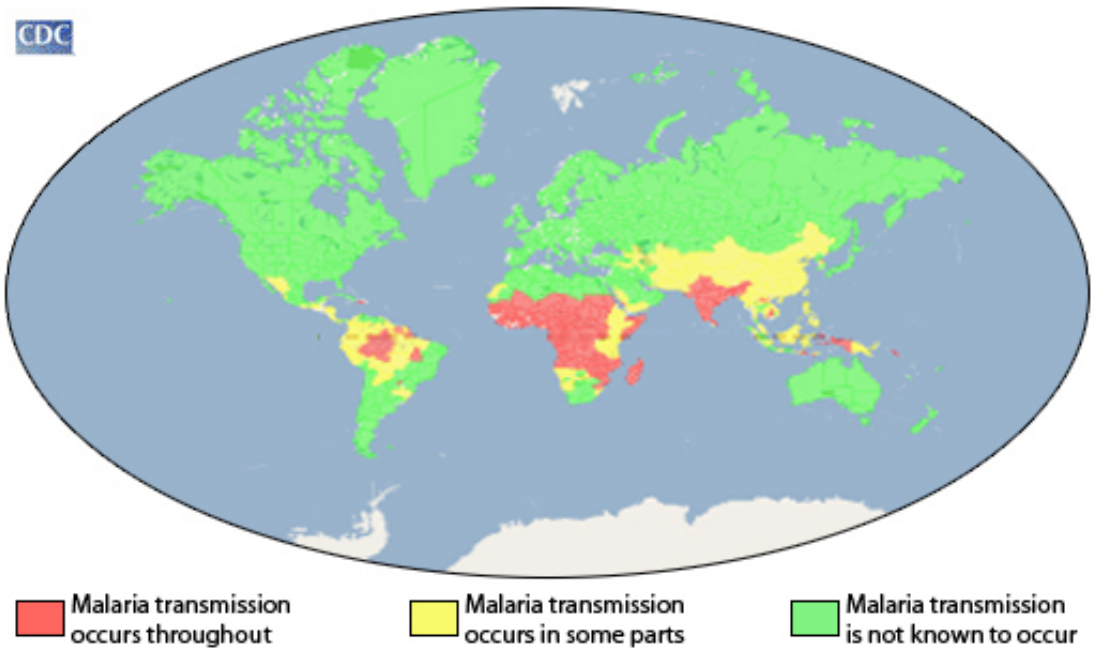


FIGURE 2.1 – Distribution du risque de transmission de la malaria dans le monde
Source : CDC

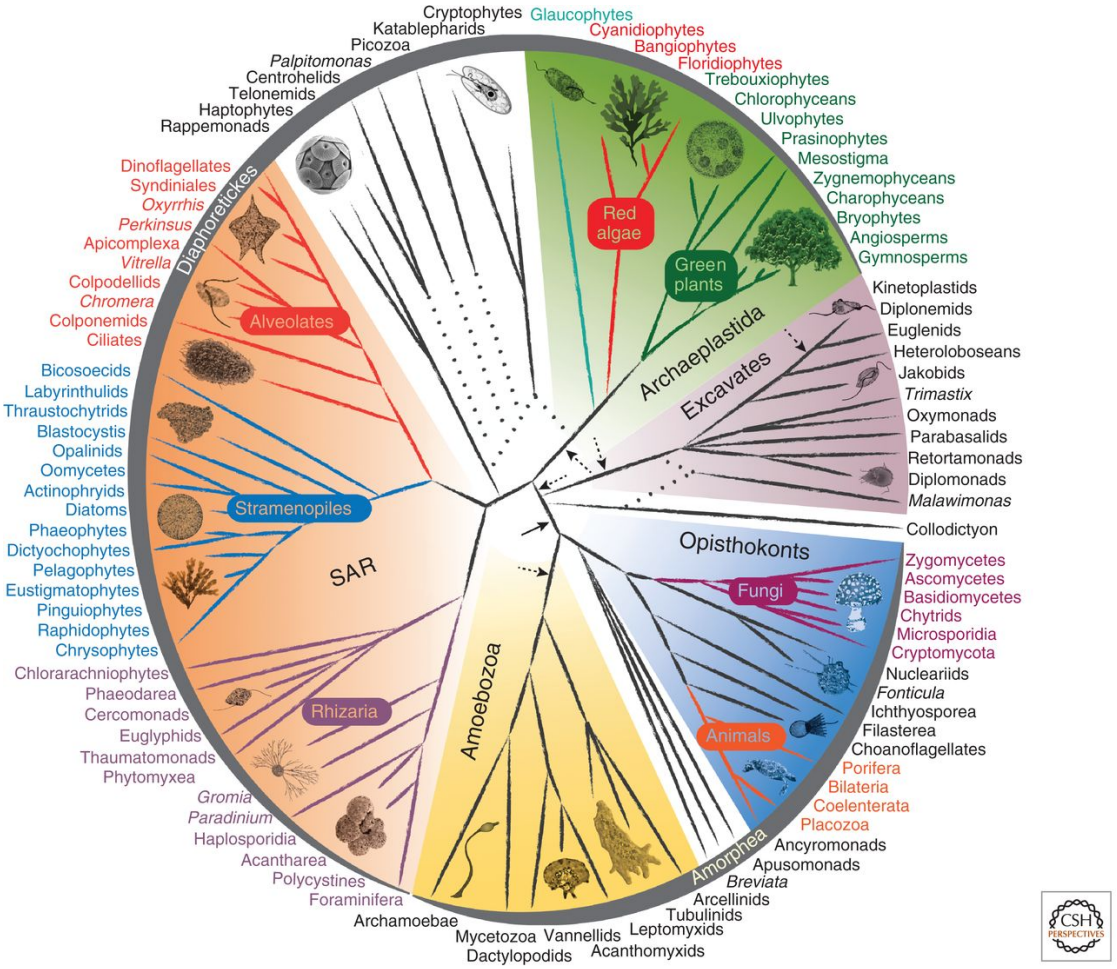


FIGURE 2.2 – Phylogénie des espèces eucaryotes
Plasmodium falciparum appartient à la division des *Alveolates* et plus précisément au phylum *Apicomplexa*. Les plantes font parties de la division des *Archaeplastida* et les animaux et champignons font partis de la division *Opisthokonts*. Le plus proche ancêtre commun entre ces espèces remonte aux origines des espèces eucaryotes. Source : Fabien Burki [Bur14]

grands singes en Afrique. Il existe au moins sept parasites du genre *Laverania* qui infectent naturellement les grands singes mais seul *Plasmodium falciparum* s'est adapté pour infecter l'Homme. Cependant l'histoire évolutive de *Plasmodium falciparum* est sujette à de nombreux débats du fait du peu de données génomiques dont on dispose sur les autres espèces de *Laverania*. Pour tenter de résoudre cette question, Otto *et al.* [OGC+18] ont réalisé plusieurs séquençages complets de génomes de chimpanzés et de gorilles infectés par un parasite *Plasmodium*. En utilisant ces séquençages, ils ont pu assembler *de novo* les génomes de référence pour six espèces de *Laverania* : *P. praefalciparum*, *P. blacklocki*, *P. adleri*, *P. billcollinsi*, *P. gaboni* et *P. reichenowi*. En utilisant ces génomes les auteurs ont alors recherché les événements de spéciation au sein du genre *Laverania* en comparant les divergences génétiques pour estimer la période de chaque événement. Ainsi ils ont pu estimer que le genre *Laverania* est apparu il y a 0.7 - 1.2 million d'années. Ils ont également pu estimer l'émergence du *Plasmodium falciparum* chez l'Homme il y a environ 40 000 à 60 000 ans et pu établir que celle ci ne provient pas d'un unique événement de transmission, comme suggéré jusqu'à présent.

2.2 Le cycle de vie

Le cycle de vie des parasite du genre *Plasmodium* est complexe, voir Figure 2.3. Il se compose d'une phase de multiplication sexuée chez le moustique femelle du genre *Anopheles* et une phase de multiplication asexuée chez l'Homme. L'Homme est considéré comme l'hôte définitif.

Chez l'Homme, les parasites se développent dans un premier temps dans les cellules du foie puis ensuite dans les globules rouges du sang (érythrocyte). Dans le sang, des couvées successives se développent et détruisent les globules rouges en libérant d'autres parasites (des mérozoïtes) qui continuent ce cycle en envahissant d'autres globules. L'étape de développement dans le sang est l'étape qui cause les principaux symptômes du paludisme et cause de sévères anémies. Les parasites peuvent également boucher les petits vaisseaux sanguins ce qui peut avoir des conséquences mortelles notamment au niveau du cerveau. La durée du cycle érythrocytaire est de 7 à 15 jours pour le *Plasmodium falciparum*. Après l'infection, il ne reste aucun dépôt parasitaire. Ce parasite infecte environ 10% des globules rouges.

Lors de la piqûre d'un moustique du genre *Anopheles*, les parasites à l'étape du cycle érythrocytaire (les gamétocytes) sont prélevés par le moustique et ceux-ci démarrent un nouveau cycle de développement différent chez le moustique. Après 10 à 18 jours, le parasite se trouve sous forme de sporozoïtes dans les glandes salivaires du moustique. Ainsi, lorsque le moustique contaminé pique un autre humain, les parasites sont injectés avec la salive du moustique et commencent

un nouveau cycle parasitaire chez l'Homme. Bien que le moustique fasse office de vecteur de contamination en transportant le parasite d'un humain à l'autre, celui-ci ne souffre pas de la présence du parasite.

2.3 Un génome atypique

Dans la suite de cette thèse, nous allons nous intéresser particulièrement à l'étude du génome de *Plasmodium falciparum* et plus particulièrement de la souche *Isolate 3D7*. Le génome de *Plasmodium falciparum* a été publié en 2002 [GHF$^+$02]. Il s'agit de la première espèce séquencée de la famille *Plasmodium* (séquençage par le Sanger Center). La publication du génome de *Plasmodium falciparum* a permis de révéler des caractéristiques communes avec les autres parasites de la malaria comme un génome de taille similaire (23.3 Mb) et un nombre de gènes (environ 5400) comparable. Mais le génome de *Plasmodium falciparum* présente également de nombreuses caractéristiques atypiques comme le biais particulier en nucléotides A+T (adénide et thymine), la présence de nombreuses régions de faible complexité, le faible nombre de facteurs de transcription ou encore la présence de nombreux gènes orphelins. Il existe des bases de données dédiées à des espèces particulières. La base de référence pour *Plasmodium falciparum* est la base PlasmoDB [BBC$^+$03]. Plus généralement, pour l'ensemble des pathogènes il existe la base EuPathDB [ABB$^+$10].

2.3.1 Le biais en A+T

La principale particularité du génome de *Plasmodium falciparum* est son taux très élevé en nucléotides A+T atteignant jusqu'à 90% dans certaines régions [GHF$^+$02]. Par comparaison, la majorité des organismes eucaryotes modèles possèdent un taux de A+T qui fluctue entre 40% et 60%. Il est à noter que ce biais nucléotidique se traduit également par un biais de composition en acides aminés des protéines de l'organisme [SH00]. En effet, on constate chez *Plasmodium falciparum* une surabondance des acides aminés asparagine (N), isoleucine (I) et lysine (K), (voir Figure 2.4), qui codent à eux seuls plus de 35% du protéome de *Plasmodium falciparum*. Certains auteurs [SH00, BLR$^+$04] ont suggéré que ce biais proviendrait d'une pression d'origine nucléique car le biais de composition en nucléotides diffère selon la position des codons sur le gène.

2.3.2 Régions de faible complexité

Une autre caractéristique du génome de *Plasmodium falciparum* est le fait que ces protéines contiennent de longues régions de faible complexité, composées de

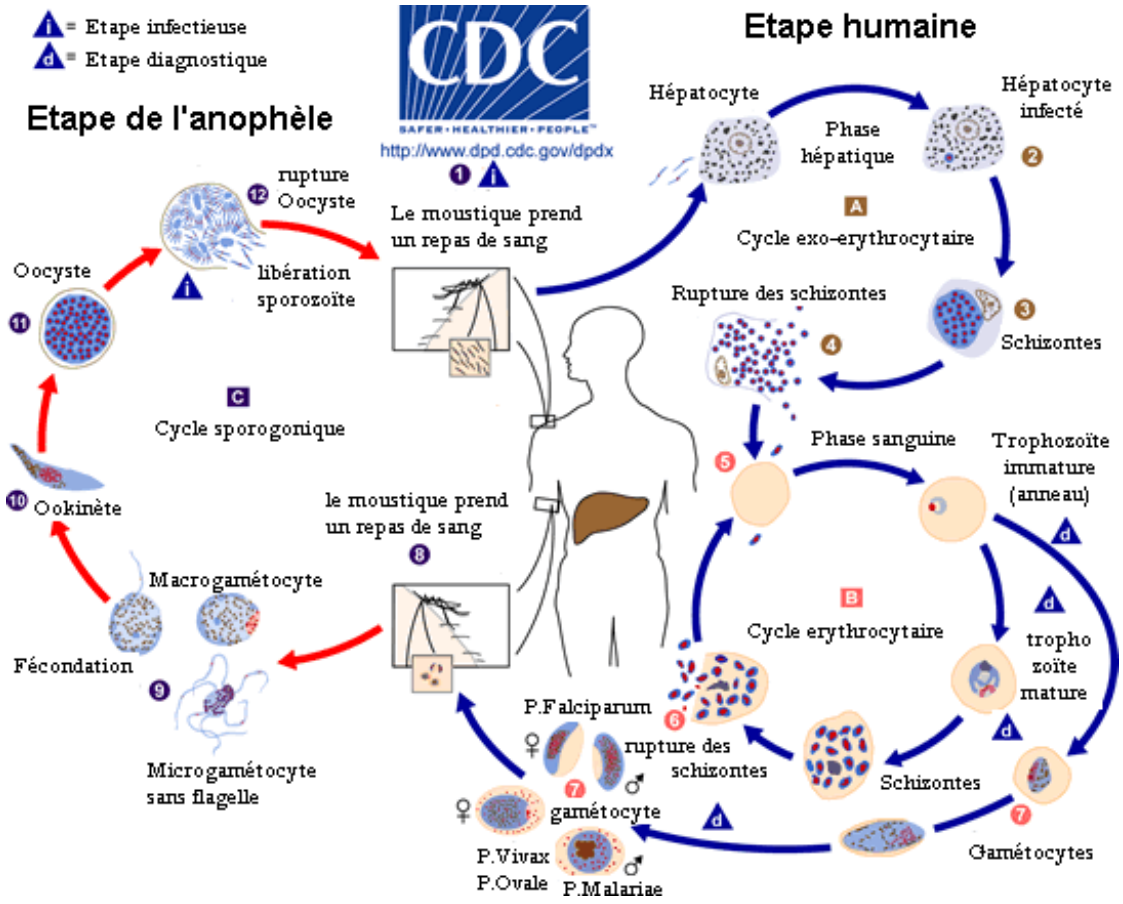


FIGURE 2.3 – Cycle de vie du parasite responsable de la malaria
Le cycle de vie de ce parasite implique deux hôtes : l'Homme et le moustique femelle *Anopheles*. Son cycle de vie se décompose en trois grandes étapes : (A) la multiplication dans les cellules du foie humain, (B) la multiplication dans les globules rouges, (C) la multiplication chez le moustique. Source : CDC

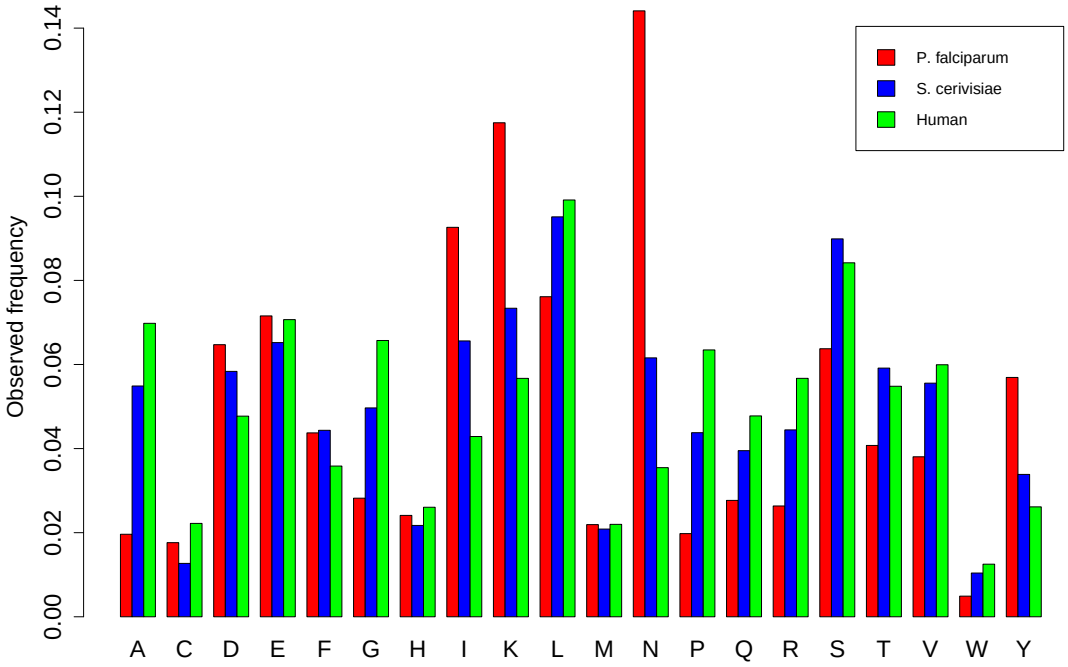


FIGURE 2.4 – Fréquence d'apparition de chaque acide aminé dans les espèces *P. falciparum*, *S. serivisiae* et l'Homme
On constate une surabondance des acides aminés N, I et K chez *Plasmodium falciparum* qui provient du biais en nucléotides A+T chez cet organisme.

quelques acides aminés répétés en tandem. Un grand nombre de protéines de *Plasmodium falciparum* sont plus longues que leur homologue dans les autres espèces du fait de l'abondance de ces régions de faible complexité. Une séquence de faible complexité simple consiste à la répétition d'un seul acide aminé tel que les répétitions d'asparagine (N) que l'on retrouve dans environ 25% des protéines de *Plasmodium falciparum*. On retrouve également un grand nombre de répétition plus complexe allant de deux à une vingtaine d'acides aminés et jusqu'à une centaine dans de rares cas [DNMO17]. Ces séquences de faible complexité n'ont généralement pas de fonction connue. Pour autant, certains auteurs [XF03] suggèrent que ces régions proviendraient d'une adaptation primaire ayant une fonction au niveau nucléique. La caractérisation fonctionnelle de ces insertions de faible complexité reste encore un sujet d'étude ouvert.

2.3.3 Un protéome mal annoté

Une autre particularité du protéome de *Plasmodium falciparum* est la couverture restreinte de ses annotations fonctionnelles. Comme on le détaillera au chapitre suivant, 38% des protéines de *Plasmodium falciparum* sont dépourvues d'annotations, un taux bien plus élevé que pour la plupart des autres organismes modèles. Plusieurs explications sont possibles. Tout d'abord le biais en acides aminés et la présence de régions de faible complexité rendent évidemment plus difficile l'identification de séquences homologues chez les autres espèces, et donc l'annotation fonctionnelle des protéines de *Plasmodium falciparum*. En outre, du fait de sa position phylogénétique très éloignée de la plupart des organismes modèles (voir Figure 2.2) il est également possible que *Plasmodium falciparum* ait développé un protéome spécifique, que l'on ne retrouve pas dans les autres grands groupes eucaryotes d'où sont issus la plupart des organismes modèles.

2.3.4 Une régulation unique de l'expression des gènes

On retrouve chez *Plasmodium falciparum* la structure de gène typique des eucaryotes avec (voir Figure 2.5) les exons, les introns et des régions flanquantes non traduites (UTR : *UnTranslated Regions*). Le nombre d'exons ainsi que leur taille varie d'un gène à l'autre. Chez *Plasmodium falciparum* on retrouve en moyenne 2,39 exons par gène d'une longueur moyenne de 949 nucléotides (bp) et composés d'environ 24% de guanine ou cytosine (G+C) [GHF$^+$02]. Les introns sont les séquences qui séparent les exons dans le gène [CGBR77, BMS77] et que l'on retrouve uniquement dans le pré-ARNm. Ces régions sont transcrites puis retirées lors de la phase de maturation de l'ARN (voir plus bas). Chez *Plasmodium falciparum* les introns ont une longueur moyenne de 179 bp et sont composés d'environ 13,5%

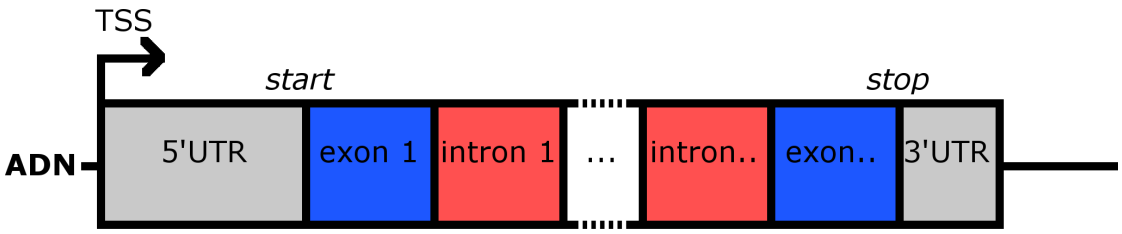


FIGURE 2.5 – Structure d'un gène eucaryote
Le gène s'étend du codon d'initiation *start* jusqu'au codon *stop*. La flèche représente le site d'initiation de la transcription (TSS), en gris les régions non traduites (UTR), en bleu les exons et en rouge les introns.

G+C. Enfin, chaque gène possède deux UTR. En amont du gène on note cette région 5'UTR et en aval la région 3'UTR. Ces régions sont de taille variable suivant les gènes. La région 3'UTR est définie entre la fin du dernier exon et la position du codon *stop* du gène. La région 5'UTR est définie entre la position du site d'initiation de la transcription (TSS : *Transcription Start Site*) et le codon *start* du gène. Il est intéressant de noter que les gènes possèdent généralement plusieurs TSS. C'est notamment le cas chez l'Homme où ces sites alternatifs de départ de la transcription sont utilisés pour la spécialisation des différents types cellulaires [RH18]. Il a été montré que *Plasmodium falciparum* contrôle le choix du TSS d'un gène au travers d'un mécanisme de sélection du TSS en fonction des étapes de son cycle de vie. Ce mécanisme semble notamment lié à la composition nucléotidique de la région promotrice [ACK+16].

Chez *Plasmodium falciparum* différents mécanismes de régulation transcriptionnelle sont mis en œuvre pour que le parasite puisse accomplir son cycle de vie et interagir avec son hôte, comme le montrent de nombreuses études [GHF+02, BLP+03, LRJF+04, LVC+05, DS06, GKG+09, HCK+10]. Certains gènes pourront être exprimés à chaque instant tandis que d'autres, ayant une tâche plus spécifique n'interviendront qu'à des instants précis. En particulier, son cycle intra-erythrocytaire de 48h est caractérisé par une chorégraphie finement coordonnée où l'expression de chaque gène culmine à un moment précis [BLP+03, LRZB+03]. Ce sont ces changements de niveau d'expression qui permettent de modifier les caractéristiques de l'organisme.

Dans la suite de cette section, nous présenterons les différents mécanismes de régulation que l'on retrouve chez la majorité des espèces eucaryotes ainsi que les particularités propres à *Plasmodium falciparum*. Dans le Chapitre 6, nous nous intéresserons à la découverte de nouvelles régions régulatrices chez *Plasmodium falciparum*.

2.3.4.1 La transcription

La transcription est la première étape de l'expression d'un gène. Ce processus consiste à la lecture d'une région ADN, le gène, pour produire un pré-ARNm (ARN messager précurseur). On peut décomposer ce processus en trois étapes :

— l'initiation de la transcription, l'ARN polymérase II (pour les gènes codants) se fixe au TSS (*Transcription Start Site*),
— l'élongation, le gène est lu et copié en pré-ARNm,
— la terminaison, la lecture s'arrête à la fin du gène.

2.3.4.2 Les régions régulatrices

On distingue généralement deux types de régions impliquées dans la régulation de la transcription. Ces régions régulent principalement l'étape d'initiation. Une première région importante est la région promotrice (*Promoter*, ou promoteur), voir Figure 2.6. Cette région se situe proche du TSS et contient des séquences ADN spécifiques qui permettent de fixer l'ARN polymerase ainsi que des protéines spécifiques de régulation appelées facteurs de transcription (voir plus bas). Il n'existe pas de définition stricte permettant de délimiter clairement le promoter mais dans la littérature, et donc en particulier dans les espèces les plus étudiées (plantes et animaux), on retrouve cependant plusieurs caractéristiques notables. Chez l'Homme par exemple, la présence des îlots CpG, des régions de 500 à 4 000 bp qui sont relativement enrichies en dinucléotides CG, environ 65% contre 40% en moyenne dans le génome. Chez l'Homme 70% des régions promotrices contiennent un îlot CpG [DB11]. Un autre élément important présent dans la région promotrice est la TATA-box. C'est une séquence ADN TATA que l'on retrouve à 25bp en amont du TSS. Cette séquence joue un rôle important dans le positionnement de l'ARN polymerase II [Her93]. Il semble que l'on retrouve également cette TATA-box chez *Plasmodium falciparum* mais le biais en A+T rend l'identification difficile [RSdCREMC$^+$05].

Une seconde région importante est la région amplificatrice (*Enhancer*), voir Figure 2.6. Cette région ADN peut fixer des protéines pour stimuler la transcription d'un gène. Un gène peut posséder plusieurs enhancers qui sont généralement situés relativement loin sur le gène (jusqu'à 100 000bp), voire même se situer sur un chromosome différent. Cependant les repliements de l'ADN permettent la proximité spatiale des deux éléments [PBD$^+$13]. Le Consortium FANTOM (Fonctional ANonTation Of the Mammaliangenome) [LHS$^+$15] définit environ 38 000 enhancers confirmés expérimentalement chez l'Homme. Chez *Plasmodium falciparum*, il semble que les enhancers soient assez rares, et la base PlasmoDB ne référence que 2 enhancers dont la caractérisation fonctionnelle reste à confirmer.

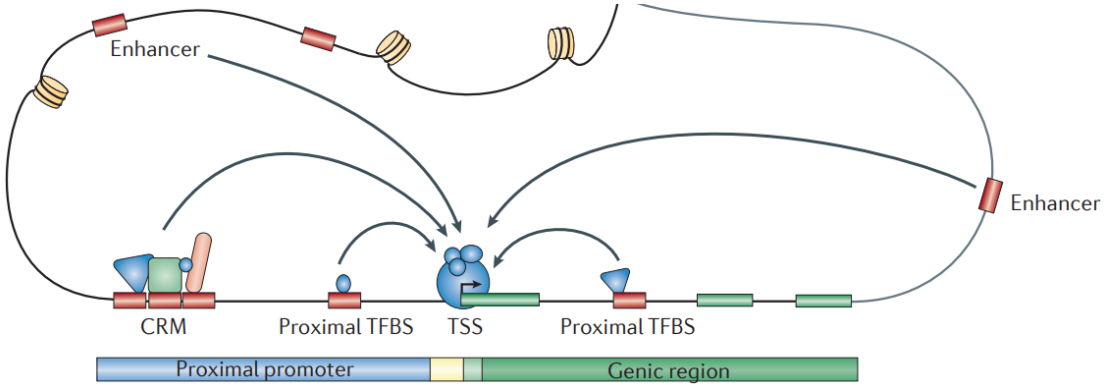


FIGURE 2.6 – Les régions régulatrices
La région promoter et les régions enhancers sont particulièrement importantes pour la régulation de la transcription. Ces régions contiennent notamment des sites de fixation (TFBS) pour des protéines spécifiques (TF) impliquées dans la régulation. Figure adaptée de Boris Lenhard [LSC12].

2.3.4.3 Les marques épigénétiques

Les marques épigénétiques sont des modifications de la chromatine qui se produisent sans modifier la séquence ADN. Ces modifications sont naturelles et essentielles à l'organisme et lorsqu'elles ne sont plus contrôlées, celles-ci peuvent engendrer des comportements anormaux de la cellule et de graves problèmes chez l'organisme [MA16]. Il existe plusieurs sortes de modifications épigénétiques telles que la méthylation de l'ADN, ou les modifications des histones.

La méthylation de l'ADN est une modification épigénétique qui permet de réguler l'expression des gènes. La méthylation de l'ADN consiste à l'ajout d'un groupe méthyle sur les cytosines (C) des dinucléotides CpG [SB08]. Chez l'Homme, la distribution des méthylations est bimodale : une grande partie des gènes est fortement méthylée tandis qu'une minorité des gènes n'est pas méthylée. Il a été montré que la fonction de la méthylation de l'ADN est de réprimer l'activité des promoters [SB08]. Les auteurs de la référence [MWR+08] ont montré une corrélation négative entre la transcription des gènes et le taux de méthylation pour les gènes dont le promoter est riche en CpG chez l'Homme. Il semble que ce mécanisme soit également présent chez *Plasmodium falciparum* mais on note certaines différences comme le fait qu'un seul brin d'ADN est méthylé et que la transcription des gènes corrèle plutôt avec la méthylation des exons [PFH+13].

Les modifications d'histones sont des modifications post-traductionnelles (PTM : *Post Translational Modifications*) qui ont lieu sur les queues des histones et affectent la structure de la chromatine. Certains auteurs pensent que ces modifications régulent l'expression des gènes en modifiant l'organisation du génome en

Histone	Méthylation	Acétylation
H2A		
H2A.Z	K27	K24, K29, K34
H2B		
H2Bv		K13, K19
H3.3		K9, K14
H3	K4, K9, K14, K36, R17	K9, K14, K18, K27
H4	K5, K16, R17	K8, K12

TABLE 2.1 – Liste des sites de méthylation et d'acétylation des histones de *Plasmodium falciparum*

régions actives où l'ADN est accessible pour la transcription, et en régions inactives où l'ADN est compacté et donc moins accessible pour la transcription. Il existe différentes modifications possibles qui seront associées plutôt à la promotion ou plutôt à la répression de la transcription. La Table 2.1 présente la liste des PTM connues chez *Plasmodium falciparum*. L'acétylation des queues d'histone est principalement impliquée dans le processus d'activation des gènes tandis que la méthylation est impliquée à la fois dans l'activation et la répression des gènes. On remarque qu'il existe chez *Plasmodium falciparum* deux variants d'histone H2A.Z et H2Bv. Il a été proposé que ces variants ont été développés par le parasite pour mieux se fixer à son génome riche en A+T qui rend l'ADN moins flexible [HSAS$^+$13, PSL$^+$13]. Un autre fait surprenant est l'absence de l'histone *linker* H1 dans son génome [MFC$^+$06].

2.3.4.4 Les facteurs de transcription

Les facteurs de transcription (TF) sont des protéines qui permettent de réguler la transcription des gènes. Les TF permettent de contrôler quand, où et comment l'ARN polymerase va se fixer au TSS [LT00]. Les TF reconnaissent des séquences ADN spécifiques localisées dans les promoters et les enhancers. Ces séquences sont relativement courtes, généralement 10 à 30 nucléotides et sont appelées les TFBS (*Transcription Factor Binding Site*). Les TF peuvent reconnaître différents TFBS. On modélise généralement l'ensemble des TFBS à l'aide d'un PWM (voir Section 1.3.2). La fixation est réalisée par le repliement de domaines de liaison du TF de façon à présenter une protubérance ou une structure flexible qui entrera en contact avec l'ADN. Ces contacts se font dans le grand sillon de l'ADN, souvent par l'intermédiaire d'une structure hélice α, avec des liaisons hydrogènes et des interactions de van der Waals. De manière plutôt surprenante compte tenu de la complexité du cycle de vie de *Plasmodium falciparum*, l'analyse de son génome a révélé un très faible pourcentage de protéines assignées à la régulation de la trans-

Espèce	#protéines	#TF	%
P. falciparum	5 449	69	1, 27
S. cerevisiae	6 436	247	3, 83
D. melanogaster	13 993	1 489	10, 64
M. musculus	20 534	1 675	8, 15
A. thaliana	27 387	2 296	8, 38
H. sapiens	20 412	1 639	8, 03

TABLE 2.2 – Nombre de protéines codantes et de facteurs de transcription par espèce
On remarque que le génome de *Plasmodium falciparum* contient un faible nombre de TF comparé aux autres espèces eucaryotes. Il est cependant à noter que ces chiffres proviennent de différentes sources de données (YEASTRACT [TMP+18], OnTheFly [SLS+14], PlantDB, PlasmoDB, [LJC+18], Uniprot) et sont souvent discutés et mis à jour mais les ordres de grandeur restent sensiblement les mêmes.

cription, environ $1,3\%$, comparé aux autres espèces eucaryotes (voir Table 2.2). Parmi les TF que l'on retrouve chez *Plasmodium falciparum*, on retrouve la majorité des TF généraux partagés par les espèces eucaryotes, à l'exception de certains membres du complexe TFIID pourtant essentiels à l'initialisation de la transcription [CPM+05]. On retrouve également une famille de TF spécifique chez *Plasmodium falciparum*, la famille ApiAP2 qui contient une vingtaine de membres que l'on ne retrouve que chez les espèces *Apicomplexa* [BBIA05].

2.3.4.5 La régulation post-transcriptionnelle

Au niveau des ARN, il existe différents mécanismes de contrôle de l'expression des gènes et de maturation du pré-ARNm. On retrouve notamment l'épissage alternatif, les protéines à domaines de liaison ARN (RBP : *RNA binding protein*) et les mécanismes d'exportation du transcrit vers le cytoplasme. La séquence codante de l'ADN joue également un rôle dans la régulation post-transcriptionnelle chez certains eucaryotes [RH05].

Une des premières étapes de la maturation de l'ARN est l'épissage alternatif. L'épissage des ARNm permet d'obtenir les transcrits matures pouvant être transportés vers le cytoplasme pour y être traduits. Durant cette étape, les introns présents dans le pré-ARNm sont retirés et les exons sont reliés entre eux. Cependant, suivant les exons choisis, il peut apparaître plusieurs produits d'épissage différent à partir du même pré-ARNm, donnant ainsi naissance à des protéines similaires mais dont la fonction peut être différente. C'est un mécanisme très important chez les

métazoaires car il permet de générer de la diversité à partir d'un gène unique. Par exemple, les auteurs de la référence [LBS+07] ont montré que 95% des gènes chez l'Homme sont soumis à l'épissage alternatif. L'épissage alternatif a été décrit pour quelques gènes de *Plasmodium falciparum* [KNHK91, PBJ+98, SPR+04, vLPJ+01] mais du fait du faible nombre d'exons par gène (environ $2,39$ exons par gène chez *Plasmodium falciparum* [GHF+02] contre 8,8 exons par gène chez l'Homme [SCK04]), il semblerait que ce mécanisme ne soit pas lié à une forte augmentation de la diversité des protéines chez *Plasmodium falciparum* car il existe assez peu de combinaisons possibles et beaucoup seraient aberrantes [YLMR19].

Lors de son séjour dans le cytoplasme, la vie d'un ARNm (le temps durant lequel il pourra servir de matrice pour l'expression d'une protéine) est déterminée par sa stabilité. Ainsi contrôler la stabilité d'un ARNm constitue un moyen de moduler l'expression d'une protéine. Cette régulation fait intervenir une classe de protéines spécifiques appelée *RNA Binding Proteins* (RBP). La majorité des RBP vont rechercher des séquences spécifiques et vont venir se fixer directement sur l'ARN [RKC+13, LQLM10, AOA06]. Certaines RBP peuvent également reconnaître des séquences spécifiques au travers de la fixation de petits ARN non codants, des ARN interférents, et cette reconnaissance peut éventuellement conduire à la dégradation de l'ARN [SCFK14]. Cependant, les auteurs de [MHKK14] ont montré qu'il n'existe pas d'ARN interférent chez *Plasmodium falciparum*. Les régions non traduites (UTR) des ARNm possèdent également des éléments responsables de la stabilité des transcrits [WAJ97, RKC+13, SCFK14, GHT14] qui interagissent spécifiquement avec des facteurs de la classe RBP. Chez *Plasmodium falciparum*, il semble que ce mécanisme soit notamment impliqué dans la traduction retardée de la protéine Pbs21, qui est produite seulement dans les stades zygote et ookinète alors que son ARNm est transcrit au stade gametocyte [STS96].

Deuxième partie

Découverte de domaines protéiques

Chapitre 3

Les protéines et domaines protéiques

3.1 Les protéines

Les protéines sont des macromolécules naturelles que l'on retrouve chez tous les êtres vivants. Les protéines sont essentielles pour la structuration et le fonctionnement des cellules. Suivant leur nature, elles peuvent assurer différentes fonctions comme par exemple :

— la catalyse de réactions chimiques (les enzymes),
— la structuration de tissus (le collagène),
— l'enroulement de l'ADN (les histones),
— la régulation de la transcription (les facteurs de transcription),
— *etc.*

En réalité, la majorité des fonctions cellulaires est assurée par les protéines. La caractérisation des fonctions des protéines est donc une tâche essentielles en biologie moléculaire et en bioinformatique pour la compréhension du fonctionnement d'un organisme.

3.1.1 Les composants

Les protéines sont composées de composés organiques, appelés acides aminés. Un acide aminé est une molécule contenant un groupe amine, un groupe acide carboxylique et une chaîne latérale. Lorsque les groupes amine et acide carboxylique sont liés au même carbone, on parle alors d'acide aminé. Il existe 20 acides aminés (Table 3.1) dans les protéines qui diffèrent uniquement par leur chaîne latérale. Les chaînes latérales confèrent des propriétés biochimiques différentes aux acides aminés. Il existe de nombreuses classifications possibles des acides aminés

Nom	Code 3 lettres	Code 1 lettre
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartate ou acide aspartique	Asp	D
Cystéine	Cys	C
Glutamate ou acide glutamique	Glu	E
Glutamine	Gln	Q
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Méthionine	Met	M
Phénylalanine	Phe	F
Proline	Pro	P
Sérine	Ser	S
Thréonine	Thr	T
Tryptophane	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

TABLE 3.1 – Liste des acides aminés avec leurs codes 1 lettre et 3 lettres respectifs

suivant leurs propriétés biochimiques (voir par exemple Figure 3.1). On désignera également les acides aminés par le terme résidu, qui est un terme plus général désignant une partie de molécule qui reste inchangée après être entrée dans la composition d'un polymère (ici une protéine). Ce terme est couramment utilisé en biologie moléculaire.

3.1.2 Les différents niveaux de structures

On distingue quatre niveaux d'organisation structurelle d'une protéine, voir Figure 3.2. La séquence des résidus liés ensemble par des liaisons peptidiques constitue la **structure primaire** de la protéine. Pour la grand majorité des protéines intra-cellulaires, cela consiste en une chaîne polypeptidique unique et linéaire [Fer99]. Les interactions des liaisons hydrogènes de la chaine polypeptidique causent des repliements de la chaîne d'acides aminés formant ainsi dans certaines régions des éléments de structures locales : les hélices α et les feuillets

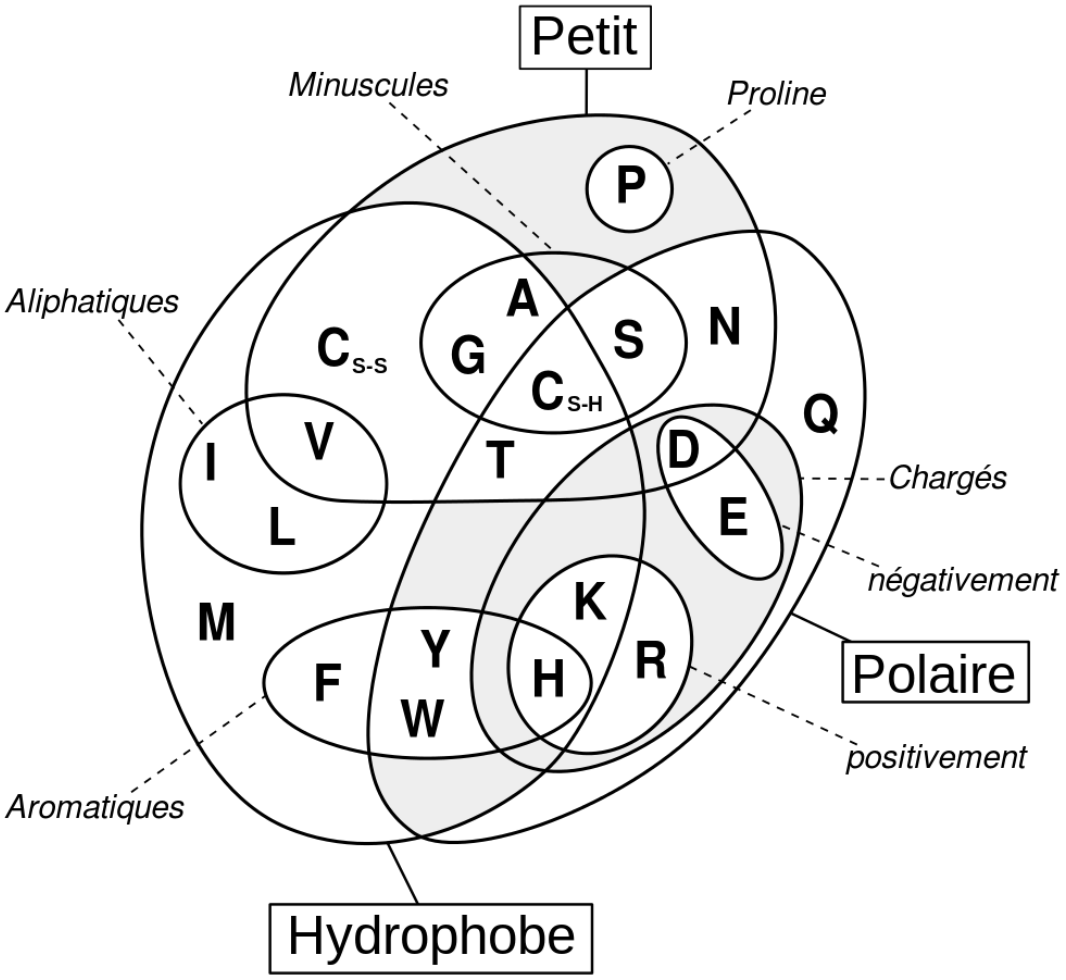


FIGURE 3.1 – Exemple de classification des acides aminés suivant leurs propriétés biochimiques
Source : Wikipedia d'après [Tay86].

β. La **structure secondaire** de la protéine fait référence alors à la décomposition en éléments de structures secondaires reliés par des régions non structurées. Le repliement tridimensionnel de la protéine est appelé la **structure tertiaire**. Cela consiste à modéliser l'emplacement de tous les atomes de la protéine dans sa conformation spatiale. Il est à noter que la densité du repliement d'une protéine est souvent très proche d'un cristal de manière à ce que sa structure soit rigide [Fer99]. Lorsque deux chaînes polypeptidiques, ou plus, s'assemblent pour former un polymère (ou complexe protéique), on parle alors de **structure quaternaire**.

3.1.3 Bases de données de protéines

Il existe de nombreuses sources de données concernant les protéines. À titre d'exemple nous présentons rapidement ici les bases UniProt, PDB et EMDB.

— La base de données UniProt [The19] vient d'une collaboration européenne entre l'European Bioinformatics Institute (EMBL-EBI), le Swiss Institute of Bioinformatics (SIB) et Protein Information Resource (PIR). Cette base de données regroupe toutes les séquences protéiques séquencées en deux bases de données distinctes Swiss-Prot et TrEMBL. Dans la version 2019_06, Swiss-Prot est constituée de 560 537 séquences protéiques annotées et vérifiées manuellement et TrEMBL regroupe 167 761 270 séquences qui ont été annotées de manière automatique mais qui n'ont pas encore été vérifiées. UniProt propose également d'autres bases de données comme la base UniRef50 qui regroupe toutes les séquences protéiques avec 50% de similarité sous la forme de clusters.

— La Protein Data Bank (PDB) [BHN03] est une base de données qui recense toutes les informations de structure des protéines. Les données sont généralement obtenues via des expériences de cristallographie aux rayons X, de spectroscopie RMN et de cryo-microscopie électronique. Dans la version actuelle, il y a 154 478 structures dans la base de données.

— La base Electron Microscopy Data Bank (EMDB) [LPB$^+$16] rassemble les expériences de microscopie électronique tridimensionnelle (3DEM) qui permettent de visualiser la structure tertiaire des protéines. Dans la mise à jour 2019_07, on retrouve 8 770 entrées.

3.2 Les domaines protéiques

Le terme domaine est utilisé pour désigner différentes entités protéiques. Les spécialistes de la structure des protéines définissent souvent le domaine comme étant une unité capable de se replier indépendamment du reste de la protéine. En

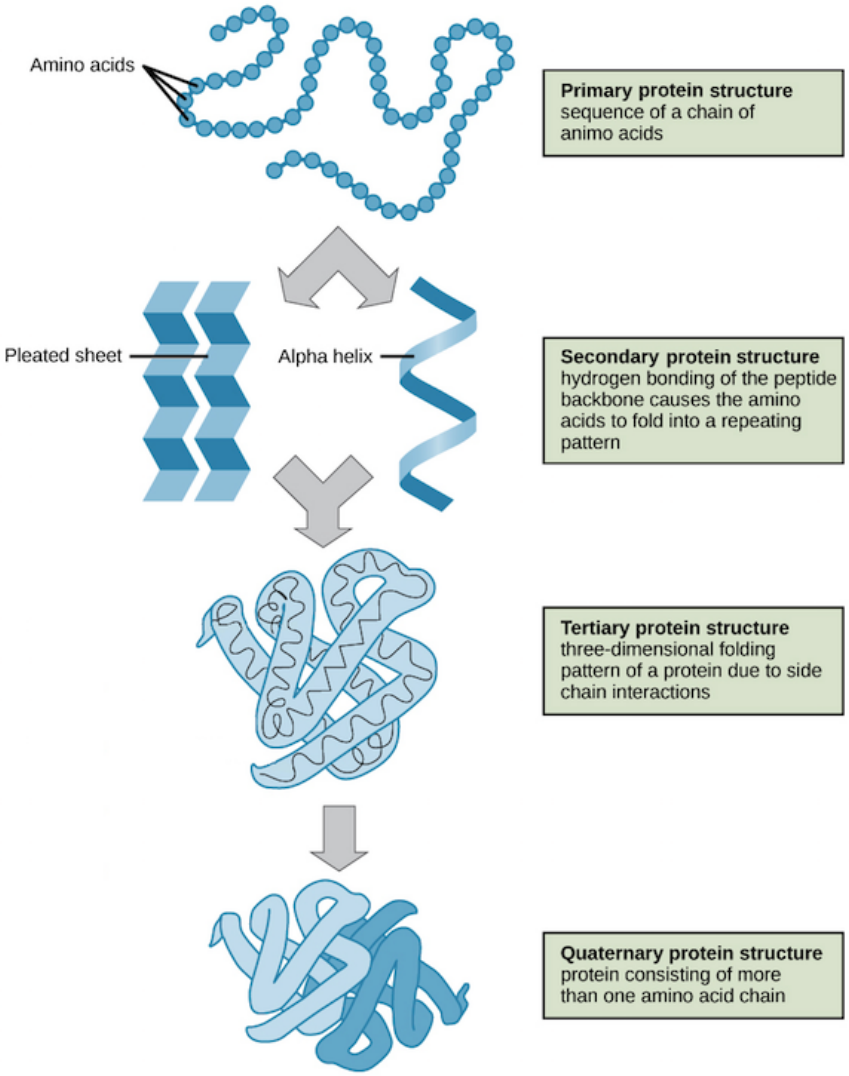


FIGURE 3.2 – Les quatre niveaux d'organisation structurelle d'une protéine
Source : [Ope18]

biochimie, les domaines sont généralement décrits comme des régions dont la fonction a été expérimentalement caractérisée indépendamment de sa structure. Enfin, une dernière définition, souvent utilisée en génomique comparative, est de considérer les domaines comme des sous-séquences homologues que l'on peut retrouver dans différentes protéines et dans différentes configurations [OT05]. Les domaines homologues sont définis comme des domaines issus d'un domaine ancestral commun. De même que pour les gènes, on pourra parler de domaines orthologues s'ils sont issus d'une spéciation et de domaines paralogues s'ils sont issus d'une duplication. Toutes ces définitions sont très souvent compatibles et s'accordent sur les régions identifiées. Cette dernière définition est proche de notre définition de domaine proposée Section 1.3. Le domaine constitue une unité indépendante pouvant constituer à lui seul une protéine mono-domaine ou pouvant s'associer avec d'autres domaines au sein d'une protéine multi-domaine [VTPL05].

3.2.1 Représentation des domaines

À l'heure actuelle, il n'existe pas de consensus strict pour représenter les domaines protéiques. Dans le cas de ProDom [BCC+05] et BLOCKS [HH94], les domaines sont représentés par une séquence consensus. Pour PROSITE [HBB+06], les domaines sont représentés sous la forme d'une expression régulière. Les bases Pfam [EGMB+19] et SCOP [AHB+04] représentent les domaines à l'aide d'un HMM profil.

3.2.2 Les bases de données de domaines protéiques

Il existe de nombreuses bases de données de domaines protéiques que l'on peut différencier suivant différent critères :

- la définition d'un domaine (structure, fonction ou séquence),
- la représentation (consensus, expression régulière, HMM profil, ...),
- le protocole de création de la base de données (automatique, manuelle, mixte),
- la couverture en nombre de séquences.

Parmi les principales bases de données de domaines protéiques, nous pouvons distinguer essentiellement deux catégories. D'un coté les bases ProDom et BLOCKS sont générées entièrement automatiquement. ProDom est construite sur la base d'un clustering automatique des sous-séquences homologues identifiées par une recherche PSI-BLAST [Alt97] contre Uniprot à l'aide de l'algorithme MKDOM2. La base BLOCKS regroupe des alignements multiples sans gap de petits motifs conservés appelés des « blocks ». Ces motifs sont identifiés à l'aide d'une implémentation du *Gibbs Sampler* [LAB+93]. D'un autre coté, les bases Pfam et PROSITE sont des approches plus hybrides avec dans le cas de Pfam, une phase de vérification

Espèce	%proteines	%résidus
P. falciparum	62	22
S. cerevisiae	82	45
D. melanogaster	80	36
M. musculus	78	46
A. thaliana	80	46
H. sapiens	71	45

TABLE 3.2 – Résumé des statistiques Pfam
Le %protéines correspond à la proportion des protéines de l'organisme couverte par au moins un domaine Pfam connu ; Le %résidus correpond à la proportion des résidus des protéines de l'organisme couverte par un domaine Pfam connu.

manuelle de l'alignement multiple qui servira de base pour l'entraînement du HMM profil, suivie d'une phase automatique pour rechercher toutes les occurrences de cette famille.

3.2.3 La base Pfam

La base Pfam est en réalité constituée de deux bases de données Pfam-A et Pfam-B. Pfam-A est la base vérifiée manuellement, tandis que la base Pfam-B est générée de manière entièrement automatique à l'aide de l'algorithme ADDA [HH03] de façon similaire à la base ProDom. Par défaut, quand on évoque les familles de Pfam on fait généralement référence à Pfam-A uniquement, les familles Pfam-B étant rarement annotées et de qualité inférieure. Par ailleurs, la base Pfam-B a cessé d'être mis à jour depuis la version 28 de Pfam.

Dans Pfam-A, chaque famille de domaines est définie par une sélection manuelle et un alignement de séquences protéiques, qui sont utilisées pour apprendre un HMM de cette famille [DREKJM98]. Pour identifier les domaines d'une nouvelle protéine, chaque HMM de la base de données est utilisé pour calculer un score qui mesure la similarité entre la séquence et la famille. Si le score est supérieur à un seuil prédéfini, propre à chaque famille, la présence du domaine dans la protéine peut être certifiée. On retrouve 17 929 familles de domaines dans la version 32 (Septembre 2018) de Pfam-A. Si on observe les statistiques de couverture de la base Pfam, on peut observer que les espèces eucaryotes modèles ont des taux de couverture relativement proches. Toutefois on observe également que ces taux sont relativement plus plus faibles pour *Plasmodium falciparum* (voir Table 3.2).

3.2.4 InterPro

Nous pouvons également citer la base InterPro [MAB+19] qui est en réalité une *méta*-base de données, c'est à dire que son objectif est de regrouper les informations de différentes bases de données, environ une dizaine, et de proposer différents outils aux utilisateurs afin de faciliter le parcours de toutes ces informations. Dans la version 76 (Septembre 2019) de InterPro, on retrouve 22 032 familles de domaines.

3.2.5 La co-occurrence des domaines protéiques

Différentes études ont montré qu'il existe un répertoire limité des combinaisons de domaines et que peu de domaines sont versatiles [AGT01, VBB+04, BBBK+05, WMBB08]. Au sein des protéines, les domaines n'apparaissent qu'avec un nombre restreint d'autres domaines auxquels ils sont liés par une forme de coopération structurelle ou fonctionnelle. Il existe donc un répertoire très limité de combinaisons de domaines dans la nature qui ne représente qu'une infime partie des combinaisons possibles. Les mécanismes aboutissant à la création de combinaisons de domaines sont probablement soumis à une forte pression de sélection [AHT03]. Nous exploiterons cette propriété forte des domaines dans les travaux présentés au chapitre suivant.

3.2.6 Le lien entre domaines et fonctions cellulaires

De nombreuses études ont remarqué une forte similarité des fonctions moléculaires et cellulaires entre protéines composées des mêmes domaines protéiques. Cela soutient donc l'hypothèse que ce sont les domaines protéiques qui confèrent leurs fonctions moléculaires aux protéines. Les auteurs de la référence [HG01] ont observé cela en comparant la similarité fonctionnelle des protéines partageant les mêmes domaines et ils ont montré que :

- les deux tiers des protéines mono-domaines qui partagent le même domaine, ont une fonction similaire,
- 35% des protéines multi-domaines qui partagent un domaine commun ont des fonctions semblables,
- ce nombre passe à 80% lorsqu'elles partagent deux domaines distincts,
- lorsque leur composition est strictement identique (en nombre et en ordre des domaines), 90% des protéines partagent les mêmes fonctions.

Toutes ces observations ont également été confirmées par les auteurs de la référence [Ye04] sur tous les domaines du vivant. Il est alors tout à fait logique de voir le domaine protéique comme l'unité fonctionnelle de la protéine.

Chapitre 4

Amélioration des outils de comparaison de paires de séquences, et découverte de nouvelles familles de domaines protéiques

4.1 Introduction

Comme on l'a vu au chapitre précédent, les protéines sont des macromolécules essentielles pour la structure et le fonctionnement des cellules vivantes. Les protéines possèdent différentes régions fonctionnelles conservées au cours de l'évolution [ZG11] et appelées « motifs fonctionnels » ou « domaines ». Les domaines peuvent être trouvés dans différentes protéines et différentes combinaisons [BBA13], et sont l'unité fonctionnelle des protéines juste au dessus du niveau des acides aminés. L'identification des domaines est donc une tâche essentielle en bioinformatique.

Différentes stratégies peuvent être utilisées pour identifier ces régions sur une protéine cible. L'analyse de profils, aussi appelée comparaison séquence-profil est une méthode efficace. Cette méthode *non ab initio* requiert une base de données de domaines protéines, telle que Pfam [FCE+16]. Dans Pfam, chaque famille de domaines est définie par une sélection manuelle et un alignement de séquences protéiques, qui sont utilisées pour apprendre un HMM de cette famille [DREKJM98]. Pour identifier les domaines d'une nouvelle protéine, chaque HMM de la base de données est utilisé pour calculer un score qui mesure la similarité entre la séquence et la famille. Si le score est supérieur à un seuil prédéfini, la présence du domaine dans la protéine peut être certifiée. Cependant, cette mé-

thode peut manquer de nombreux domaines lorsqu'on l'applique à un organisme phylogénétiquement éloigné des espèces utilisées pour entraîner le HMM. Cela peut arriver pour deux raisons. Tout d'abord, si la séquence protéique a fortement évolué, le HMM de la base de données peut mal correspondre aux spécificités de l'organisme distant. Dans ce cas, il existe plusieurs approches. Une première approche consiste à utiliser la propriété de co-occurrence des domaines protéiques [TGMB09, OLS11, GFG$^+$14, OS17, BVZC16, BZVC16]. Une autre approche consiste à modifier le HMM pour le faire correspondre aux spécificités d'une espèce cible. Les auteurs de Terrapon *et al.* [TGMB12] ont proposé plusieurs solutions à ce problème. Enfin, une autre approche consiste à analyser les motifs des acides aminés hydrophobes dans la séquence car les groupement hydrophobes sont plus conservés que la séquence elle-même. Ils peuvent ainsi être utilisés comme une signature pour comparer des séquences [CPM$^+$05, BFHBBC15]. Une autre possibilité qui pourrait expliquer les domaines manquants est que ces domaines appartiennent à des familles qui sont tout simplement absentes des base de données de domaines protéiques. Les bases de données comme Pfam ont été construites sur la base de séquences eucaryotes qui proviennent majoritairement des plantes, des champignons et des animaux, mais assez peu des autres groupes. Ainsi, la proportion des protéines couvertes par un domaine Pfam chez les plantes, les champignons et les animaux est près de deux fois supérieure à la proportion des protéines couvertes dans les autres groupes [GFG$^+$14]. Par exemple, chez *Plasmodium falciparum*, seulement 22% des résidus de ses protéines sont couverts par un domaine Pfam tandis que cette proportion est de 44% chez l'Homme et la levure (*Saccharomyces cerevisiae*).

Lorsque les domaines sont absents des bases de données, il existe une approche alternative pour leur identification qui consiste à lancer une recherche *ab initio* en utilisant un outils de comparaisons séquence-séquence tel que FASTA [PL88] ou BLAST [AGM$^+$90]. Ces outils recherchent des similarités locales entre une protéine requête et une base de données de séquences comme Uniprot [The15]. Étant donné que les domaines protéiques sont des sous-séquences conservées au cours de l'évolution, les similarités locales entre protéines correspondent généralement à ces régions. Comme on l'a vu au Chapitre 1, les outils comme BLAST utilisent une fonction de score spécifique pour évaluer la similarité et estiment la p-valeur d'obtenir cet alignement sous des hypothèses spécifiques de distribution des scores. Ces approches de comparaisons séquence-sequence n'incluent pas d'information provenant d'autres séquences homologues et sont donc plus enclins à produire des faux positifs/négatifs que les approches séquence-profil. Par conséquent, elles sont généralement utilisées avec des seuils de scores relativement stricts pour limiter le nombre de faux positifs et peuvent donc elles aussi manquer certaines homologies. Il existe différentes versions de BLAST qui ont été développées pour améliorer la

sensibilité de cette approche. Par exemple, PSI-BLAST [Alt97], qui construit une matrice PSSM (*Position-Specific Score Matrix*), équivalente à un PWM, pour effectuer des recherches progressives, PHI-BLAST [ZSM+98] qui utilise des motifs pour initialiser un alignement, ou encore DELTA-BLAST [BSA+12], qui commence par rechercher dans une base de données de PSSM pré-construits avant de rechercher dans la base de données de séquences afin de mieux détecter les homologies.

Étonnament, la co-occurrence de domaines n'a pas été utilisée jusqu'ici pour améliorer la sensibilité des approches de comparaisons séquence-séquence dans les protéines. Comme on l'a vu dans le chapitre précédent, la co-occurrence de domaines est une propriété très forte des protéines, basée sur le fait que la majorité des domaines protéiques tendent à apparaître avec un nombre limité d'autres domaines sur une même protéine [BBA13]. Une exemple bien connu sont les domaines PAZ et PIWI, que l'on retrouve fréquemment ensemble : lorsque l'on observe les protéines qui portent le domaine PAZ, le domaine PIWI est souvent présent également. L'information de co-occurrence a déjà utilisée pour améliorer la sensibilité des approches séquence-profil [TGMB09, OS17, BVZC16, BZVC16]. Cependant, cette information pourrait également être très utile pour la détection d'homologies dans les approches de comparaisons séquence-séquence. Par exemple, la Figure 4.1 montre les résultats BLAST d'une protéine de *Plasmodium falciparum* contre trois protéines provenant de la base Uniprot. La majorité de ces hits ont une e-valeur modérée et, pris individuellement, ne pourraient pas être considérés avec une grande confiance dans une analyse classique. Cependant, chacun de ces hits apparait en co-occurrence avec un ou deux autres hits sur la même protéine et ces co-occurrences sont présents dans les trois protéines. Considérés de manière collective, ils apportent donc une preuve solide des homologies identifiées.

Nous avons donc proposé une nouvelle méthode pour prendre en compte l'information de co-occurrence dans une analyse BLAST classique et pour construire de nouvelles familles de domaines sur la base de ces résultats. Dans ce chapitre nous commencerons donc par présenter le principe de notre méthode. Nous verrons ensuite en détails les résultats obtenus sur l'analyse du protéome de *Plasmodium falciparum*. Puis nous terminerons sur une discussion des résultats et des améliorations possibles de la méthode.

4.2 Méthode

Notre objectif est d'améliorer la sensibilité des outils de comparaison de paires de séquences tels que BLAST en utilisant l'information de co-occurrence. Le cœur de notre apparoche est une nouvelle fonction de score qui permet de prendre en

```
# query id, subject id,% identity, alignment length, mismatches, gap opens, q.start, q.end, s.start, s.end, evalue, bit score
[...]
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_H2YSE2 24.51   102     65      4       2689    2788    639     730     4e-05   38.5 *
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_H2YSE2 20.00   175     136     3       3635    3805    1143    1317    2e-06   42.7 -
[...]
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_Q7RG07 33.96   106     58      4       2689    2788    976     1075    4e-05   38.9 *
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_Q7RG07 21.95   369     218     11      3496    3853    1601    1910    2e-06   43.5 -
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_Q7RG07 26.07   234     164     6       1457    1681    301     534     2e-07   47.0 +
[...]
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_J9JQM2 32.11   109     66      4       2689    2795    535     637     8e-05   37.7 *
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_J9JQM2 22.73   352     184     14      3499    3827    1002    1288    1e-05   40.4 -
[...]
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_E4X0B2 28.12   96      64      3       2689    2782    1502    1594    2e-04   37.0 *
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_E4X0B2 18.12   276     171     6       3534    3803    1903    2129    4e-06   42.4 -
[...]
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_J7R331 40.43   47      27      1       2737    2782    2030    2076    3e-04   36.6 *
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_J7R331 19.57   184     139     4       3633    3810    2498    2678    4e-05   39.3 -
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_J7R331 20.58   277     203     11      1425    1686    1294    1568    2e-06   43.5 +
[...]
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_W5N8P1 29.51   61      39      2       2744    2803    1039    1096    4e-04   35.4 *
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_W5N8P1 18.66   343     213     13      3491    3810    1419    1718    3e-05   39.3 -
[...]
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_Q6CXU0 26.79   112     73      4       2692    2801    1972    2076    6e-04   35.0 *
tr|Q8IKH9|Q8IKH9_PLAF7  UniRef50_Q6CXU0 18.63   204     154     6       3633    3829    2484    2682    2e-05   40.4 -
[...]
```

FIGURE 4.1 – Extrait des résultats de BLAST de la protéine Q8IKH9_PLAF7 contre UniRef50
Certains hits ont été masqué pour plus de lisibilité. On voit ici les hits identifiés entre la protéine Q8IKH9_PLAF7 et 7 protéines différentes de la base UniRef50. Suivant les cas, les résultats BLAST révèlent la co-occurrence de deux ou trois sous-séquences indépendantes (chaque sous-séquence est surlignée d'une couleur différent).

compte l'information de co-occurrence pour évaluer des hits BLAST. Cette nouvelle fonction nous permet d'identifier des hits significatifs qui ne seraient habituellement pas considérés sur la seule base des résultats de BLAST à cause de leur e-valeur trop haute. La Figure 4.2 représente un diagramme des principales étapes de la procédure.

4.2.0.1 Découverte des domaines à partir de BLAST

La première étape de notre méthode consiste à effectuer une recherche d'homologie locale avec BLAST d'une protéine de l'organisme d'intérêt contre les séquences d'une base de donnée de protéines. Pour les analyses suivantes, nous avons utilisé la base de donnée UniRef50 fournie par UniProt (Section 3.1.3) et l'implémentation BLASTP (BLAST spécialisé pour l'analyse de séquences protéiques, version 2.2.28) avec les paramètres par défaut et un seuil de e-valeur maximum prédéfini (par exemple 10^{-3}). Cette recherche nous fourni une liste de toutes les similarités locales identifiées avec la protéine d'intérêt. Chaque paire de sous-séquences similaires est appelée un *hit*. Tous les hits dont la longueur est inférieur à 30 résidus sont supprimés et dans le cas où plusieurs hits se chevauchent sur la protéine d'intérêt, seul le hit avec la e-valeur la plus faible est conservé. Il est également à

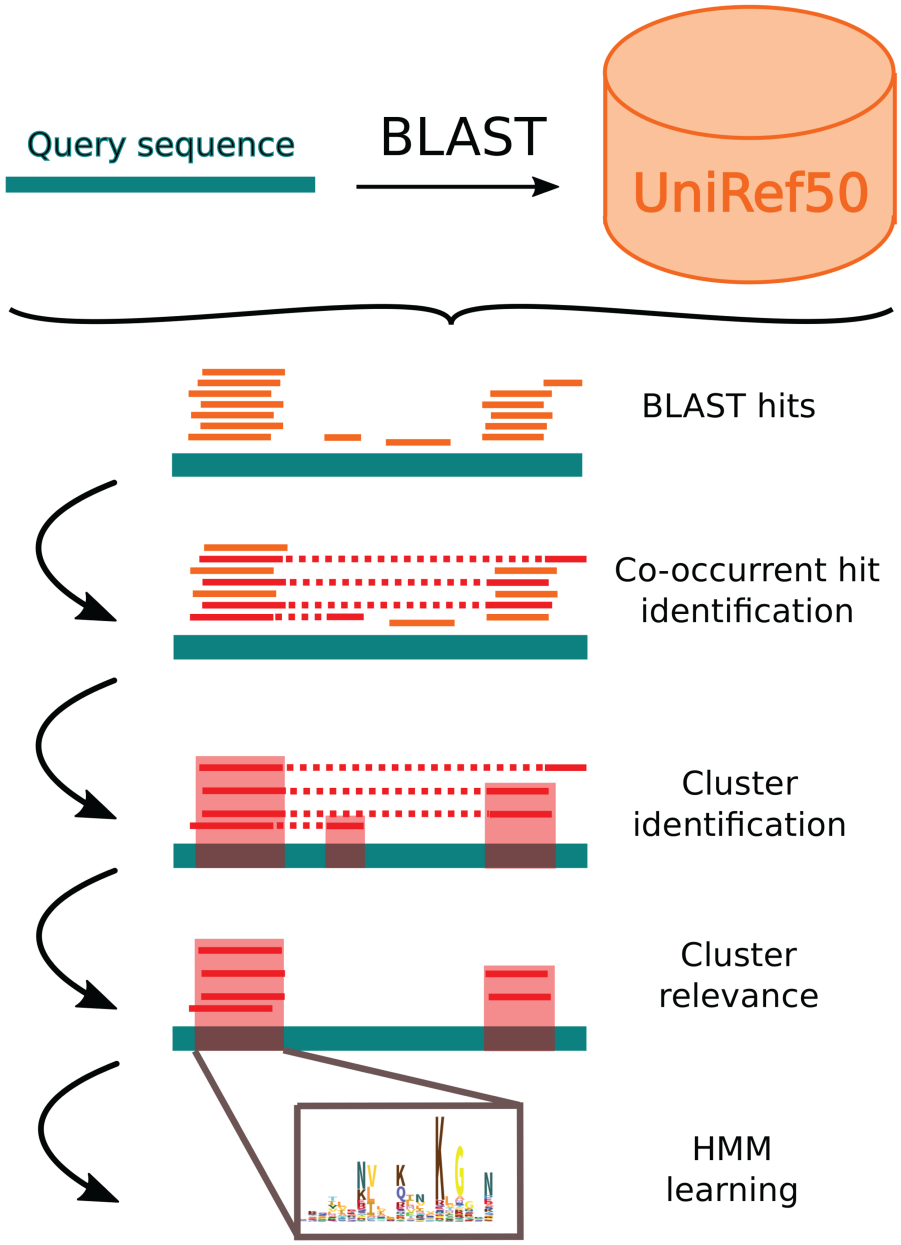


FIGURE 4.2 – Étapes principales de notre procédure
La première étape consiste à réaliser un alignement de séquence avec BLAST d'une protéine cible (*query*) contre une base de séquences. La deuxième étape consiste à identifier les hits co-occurrents ainsi que les clusters de hits co-occurrents. La troisième étape consiste à évaluer la pertinence des différents clusters identifiés. Enfin, la dernière étape consiste à construire une nouvelle famille de domaines à partir des hits sélectionnés.

noter que, par défaut, BLAST contrôle la complexité des séquences en supprimant automatiquement des résultats les alignements de faible complexité.

Ensuite, nous utilisons les résultats de BLAST pour découvrir les domaines potentiels. Notre hypothèse est que les domaines sont les sous-séquences les plus conservées dans les protéines. Sous cette hypothèse, les domaines devraient correspondre aux régions avec le plus grand nombre de hits BLAST. Par conséquent, nous utilisons la densité de hits par résidu pour identifier les domaines potentiels de la protéine étudiée. L'inconvénient principal de cette approche est qu'elle peut potentiellement produire de nombreux faux-positifs. Tout d'abord parce que les pics de hits peuvent être dûs à des régions de faible complexité qui n'ont pas été correctement identifiées par BLAST. Ensuite, même si le pic correspond effectivement à la présence d'un domaine, tous les hits qui composent le pic ne sont pas nécessairement de vraies occurrences du domaine, et la proportion des contaminants peut être très élevée dans certains cas. Pour limiter au maximum le nombre de faux-positifs, notre solution est de nous intéresser uniquement aux hits co-occurrents. On dit qu'un hit est co-occurrent à un autre hit s'il fait intervenir les mêmes protéines requêtes et cibles et que ces deux hits ne se chevauchent sur aucune des deux protéines de la paire. Partant de là, plutôt que de travailler sur la densité de tous les hits (courbe bleu de la Figure 4.3), nous travaillons sur la densité des hits co-occurrents, c'est à dire la densité des hits pour lesquels il existe au moins un autre hit co-occurrent sur la même protéine (courbe rouge de la Figure 4.3).

L'objectif est donc d'identifier des pics de hits homologues qui pourraient représenter des domaines protéiques. Une manière de voir le problème serait de le voir comme un problème de segmentation des courbes de densité. C'est cependant un problème de segmentation particulier dont la résolution nécessite de prendre en compte des informations qui vont au delà de la simple courbe de densité. En effet, le problème est rendu difficile par le fait que tous les hits d'un pic n'ont pas la même longueur et ne sont pas parfaitement alignés. Plus encore, deux domaines adjacents sur la protéine étudiée peuvent aussi être adjacents sur les protéines cibles de la base de données. Dans ce cas, les deux domaines peuvent apparaître sous la forme d'un seul et unique hit relativement long et cela peut créer des ambiguïtés.

Nous avons donc développé une heuristique itérative pour résoudre ce problème. La méthode se focalise uniquement sur la densité des hits co-occurrents et commence par identifier la position sur la séquence avec la densité la plus forte. Tous les hits qui couvrent ce résidu spécifique sont sélectionnés et utilisés pour définir les bornes du domaine. Cette méthode sélectionne ensuite de manière itérative un sous ensemble de ces hits en supprimant au fur et à mesure les hits dont les bornes sont trop éloignées des autres hits sélectionnés (voir Algorithme 3). À la fin de cette procédure nous avons identifié un cluster de hits similaires : similaires

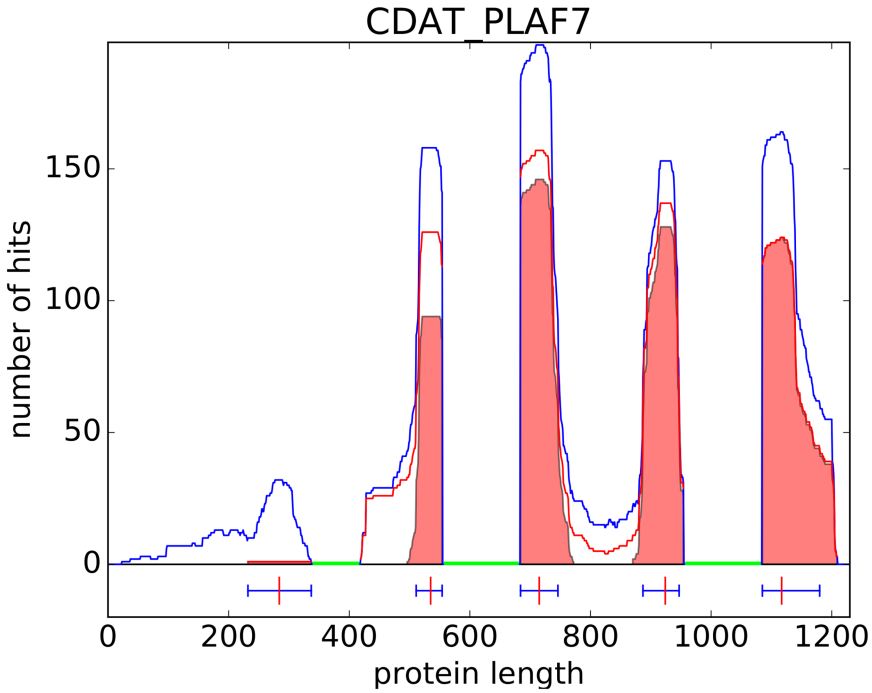


FIGURE 4.3 – Densité des hits BLAST par résidu sur la protéine CDAT_PLAF7 La courbe bleu représente la densité de tous les hits BLAST identifiés dans UniRef50. La ligne rouge représente la densité obtenue en ne considérant que les hits co-occurrents. Les régions remplies en rouge représentent les clusters de hits identifiés par notre algorithme. Les lignes sous l'axe des abscisses représentent la position des clusters (les bornes des domaines) identifiés par notre procédure. Dans cet exemple, les régions déjà couvertes par un domaine Pfam ont été masquées (les régions vertes) de façon à ignorer les hits BLAST qui pourraient les chevaucher (voir Section 4.2.0.5.

de part le contexte, car ils partagent la propriété de co-occurrence, et similaires en séquence parce qu'ils sont localisés sur la même région de la protéine étudiée. Le cluster défini une famille de domaines potentielle, et les différents hits sont différentes occurrences de ce domaine dans différentes protéines. La région couverte par ce cluster est ensuite masquée et toute la procédure est appliquée de nouveau jusqu'à ce qu'on ne puisse plus identifier de nouveau domaine sur cette protéine.

4.2.0.2 Évaluation de la pertinence des clusters

En sélectionnant uniquement les hits qui possèdent un autre hit co-occurrent sur la même protéine, nous devrions limiter fortement le nombre de faux positifs dans nos clusters. Cependant, il est aussi envisageable que la co-occurrence de deux hits soit fortuite. Pour contrôler cela, pour chaque cluster identifié (famille de domaines) par notre procédure, nous allons comparer le nombre de hits co-occurrents (courbe rouge) au nombre total de hits qui chevauchent cette région (courbe bleue), et nous allons estimer la probabilité d'avoir observé autant de hits co-occurrents par hasard. Pour cela nous utilisons un test binomial, et la p-valeur est calculée comme ci :

$$\text{p-valeur} = \sum_{k=m}^{n} \binom{n}{k} p^k (1-p)^{n-k}, \tag{4.1}$$

avec m le nombre de protéines cibles avec un hit chevauchant le domaine et un autre hit co-occurrent en dehors du domaine (courbe rouge, Figure 4.3). n est le nombre total de protéines avec un hit chevauchant le domaine (courbe bleue, Figure 4.3). p est la probabilité a priori qu'un hit chevauchant le domaine soit co-occurrent avec un autre hit en dehors du domaine étant donné le nombre de hits obtenus hors du domaine. Cette probabilité dépend de la protéine étudiée et du domaine identifié. Par exemple, certaines protéines peuvent contenir plusieurs régions de faible complexité avec de très nombreux hits. Dans ce cas, il est alors plus facile pour un hit d'identifier un autre hit co-occurrent hors du domaine. La probabilité p est alors estimée par le nombre total de protéines avec un hit sur la protéine étudiée hors du domaine, divisé par le nombre total de protéines dans la base de données. On peut alors calculer p de la manière suivante :

$$p = \frac{N - n + m}{U},$$

avec N le nombre total de protéines avec un hit sur la protéine étudiée et U le nombre total de protéines dans la base de données (UniRef50 dans ce cas). Le nombre de protéines qui chevauchent le domaine est calculé à la position avec la plus forte densité. Si plusieurs résidus ont la même densité, nous choisissons la position centrale. Les domaines avec une p-valeur supérieure à un seuil prédéfini (par exemple 10^{-3}) sont rejetés.

Algorithme 3 Identification des domaines potentiels

Entrée: $\mathscr{H}$: un ensemble de hits co-occurrents sur une protéine de l'organisme étudié

Sortie: $\mathscr{C}$: un ensemble de clusters de hits. Chaque cluster $C \in \mathscr{C}$ est défini par un ensemble de hits et une position de début C_s et de fin C_e sur la protéine.

$\mathscr{C} \leftarrow \emptyset$
faire
  $\mathscr{H}^* \leftarrow \mathscr{H}$ privé des hits qui chevauchent un cluster $C \in \mathscr{C}$
  Calculer la densité des hits avec $\mathscr{H}^*$
  $P \leftarrow$ la position avec la plus forte densité
  $C \leftarrow$ l'ensemble des hits qui couvrent P
  $C_s \leftarrow$ la plus petite position de début dans C
  $C_e \leftarrow$ la plus grande position de fin dans C
  faire
    Instable $\leftarrow$ Faux
    $N \leftarrow$ nombre de hits dans C
    Calculer N_s et N_e, le nombre de hits dans C qui couvrent C_s et C_e respectivement
    si $N_s < 1/3 \times N$ **alors**
      $C_s \leftarrow C_s + 1$
      Instable $\leftarrow$ Vrai
    fin si
    si $Ne < 1/3 \times N$ **alors**
      $C_e \leftarrow C_e - 1$
      Instable $\leftarrow$ Vrai
    fin si
    Supprimer les hits de C si plus de 30% de leurs résidus sont hors de la région $[C_s \ldots C_e]$
  tant que Instable
  si $(C_e - C_s) > 30$ **alors**
    Ajouter C dans $\mathscr{C}$
  fin si
tant que au moins un nouveau cluster ajouté dans $\mathscr{C}$

4.2.0.3 Estimation du nombre de faux positifs

La procédure décrite ci-dessus a été prévue pour être appliquée à l'ensemble du protéome d'un organisme particulier. Pour chaque domaine découvert, la p-valeur calculée nous permet de vérifier que le nombre de hits co-occurrents obtenus n'est pas simplement dû au hasard. Cela nous permet de vérifier que les hits qui composent un cluster appartiennent vraisemblablement à la même famille de domaines. Cependant, cela ne garantit pas que la protéine requête soit effectivement homologue à cette famille. En effet, il est envisageable qu'une protéine puisse contenir deux régions qui ressemblent par hasard à deux domaines co-occurrents. Dans ce cas, nous observerions de très nombreux hits co-occurrents mais les régions de la protéine ne seraient pas homologues aux familles de domaines identifiées. Bien que cela ne devrait se produire que très rarement, il est important d'estimer le nombre de faux positifs détectés par notre procédure lorsqu'on l'applique à un protéome entier. Nous avons utilisé deux procédures pour estimer cette proportion. Dans la première procédure, chaque séquence du protéome étudié est mélangée en permutant aléatoirement les blocs de 4-mers qui la composent. Dans la seconde procédure, chaque séquence est simplement renversée. Ensuite, toute notre procédure est appliquée sur ces fausses séquences et on compte le nombre de domaines identifiés avec une p-valeur inférieur au seuil de référence. En comparant le nombre de domaines «identifiés» dans les fausses données et le nombre identifié dans les vraies données, nous pouvons estimer la proportion de faux positifs (notée FDR pour *False Discoveries Rate*) produits par notre procédure :

$$\text{FDR} = \frac{\text{nombre de domaines identifiés dans les fausses données}}{\text{nombre de domaines identifiés dans les vraies données}}. \tag{4.2}$$

Par la suite, nous utiliserons donc deux estimations du FDR, une pour chaque procédure de génération des fausses séquences.

4.2.0.4 Apprentissage des nouveaux modèles

Une fois que nous avons identifié une nouvelle famille de domaines avec la procédure précédente, il y a une étape additionnelle et optionnelle qui consiste à entraîner un HMM pour chaque cluster identifié. Pour cela, nous ne considérerons que les clusters avec au moins 5 séquences pour assurer un minimum de diversité dans nos familles. Nous commençons par réaligner les séquences de chaque cluster en utilisant le programme MUSCLE [Edg04b, Edg04a], avec les paramètres par défaut. Les positions flanquantes avec moins de 75% de résidus sont supprimées. Ensuite, chaque alignement multiple de séquences (MSA) est utilisé pour entraîner un nouveau HMM qui représente la famille correspondante en utilisant le logiciel

HMMER3 (version 3.1b2) [Edd98] [1].

4.2.0.5 Intégration des domaines connus dans la procédure

Afin de concentrer la recherche sur les régions qui ne sont pas déjà couvertes par un domaine connu, une amélioration intéressante est d'inclure l'information des domaines connus dans la procédure. Cela se fait en deux étapes. Premièrement, avant d'effectuer la recherche BLAST, les régions déjà couvertes par un domaine Pfam sont masquées en remplaçant les résidus couverts par le résidu inconnu (X) qui est automatiquement ignoré par BLAST. Ensuite, lorsque l'on recherche les hits co-occurrents dans les résultats de BLAST, les domaines Pfam sont considérés comme des hits potentiels. Plus précisément, cela signifie que si on considère une protéine requête A avec un hit BLAST sur la protéine cible B, et si A et B portent le même domaine Pfam D (et que D ne chevauche pas le hit identifié sur A et B), alors le hit BLAST est considéré comme ayant un hit co-occurrent. L'intégration des domaines connus dans la recherche présente deux avantages. Tout d'abord, cela permet d'accélérer la recherche BLAST en masquant une partie du protéome. Ensuite, les hits Pfam représentent souvent une information d'homologie de meilleure qualité qu'un simple hit BLAST. Leur intégration permet donc également de minimiser les chances de détecter des co-occurrences fortuites et on s'attend donc à ce que cela améliore la qualité des résultats.

4.3 Analyse du protéome de *Plasmodium falciparum*

Nous avons appliqué notre procédure sur les protéines de *Plasmodium falciparum*. Chaque séquence protéique (PF3D7 version du 21 Février 2016 sur Uniprot ; 5 365 protéines au total) a été utilisée pour réaliser une recherche BLAST contre la base de données UniRef50 (version Octobre 2015) [SWH+15] restreinte aux séquences eucaryotes, qui regroupe 2 784 993 séquences provenant de 2 753 organismes de références avec un maximum de 50% de similarité entre chaque paire de séquences. Nous avons également utilisé la version 28 de Pfam pour ces expériences.

Nous avons tout d'abord lancé notre approche sans intégrer les domaines Pfam connus (voir Table 4.1). Nous avons utilisé un seuil de e-valeur de 10^{-3} pour la recherche BLAST. Avec ce seuil, nous avons identifié un total de 10 474 clusters

1. Les HMM sont entraînés à l'aide de la commande suivante : `hmmbuild -n <hmm_name> -amino -fast <hmmfile_out> <msafile>;` avec `<hmm_name>` le nom du HMM entraîné, `-amino` permet de spécifier que l'alignement en entrée est une séquence protéique, `-fast` assigne les colonnes avec $\geq 50\%$ (par défaut) de résidus aux états matchs du HMM, `<hmmfile_out>` le nom du fichier de sortie, `<msafile>` l'alignement en entrée.

Expériences	sans Pfam		avec Pfam	
	tous les clusters	≥ 5 hits	tous les clusters	≥ 5 hits
nombre de clusters	10 474	3 095	7 680	2 279
avec p-valeur $\leq 0.1\%$	6 489	3 033	6 467	2 240
protéines différentes	2 792	1 355	3 214	1 293
couverture des résidus	32.85%	12.90%	24.61%	6.71%
FDR estimé	3.76%	3.85%	6.22%	1.83%
#domaines chevauchant un Pfam	2 830	2 221	0	0

TABLE 4.1 – Résumé du nombre des nouvelles occurrences de domaines identifiées par notre approche
Dans la colonne «sans Pfam» les analyses ont été réalisées sur le protéome complet, sans masquer la présence des domaines Pfam connus aussi bien dans la recherche BLAST que dans l'analyse des hits co-occurrents. À l'inverse, dans la colonne «avec Pfam» les domaines Pfam ont été masqués dans la recherche BLAST et la présence des domaines Pfam a été utilisée pour l'analyse des hits co-occurrents.

(3 905 avec au moins 5 hits). Parmi ces 10 474 clusters, 6 489 avaient une p-valeur inférieur à 0.1% (3 033 avec au moins 5 hits) sur 2 792 protéines différentes (1 355 pour les clusters avec au moins 5 hits). Ces clusters couvrent 32.85% des résidus du protéome de *Plasmodium falciparum*. Les clusters avec au moins 5 hits couvrent 12.90% des résidus. Le FDR de la procédure est estimé à 3.76% avec les séquences renversées (voir plus bas pour la comparaison des FDR estimés avec les deux procédures). Parmi les 3 033 clusters avec au moins 5 hits, 2 221 chevauchent un domaine Pfam déjà connu. On considère qu'il y a un chevauchement si au moins un tiers des résidus du plus petit domaine est inclus dans l'intersection des deux.

Nous avons ensuite évalué dans quelle mesure notre procédure automatique est capable de retrouver les domaines présents dans la base Pfam. Pour répondre à cette question, nous avons généré un nouveau HMM pour chacun des 3 033 clusters avec plus de 5 hits et nous avons comparé ces HMM avec ceux des domaines Pfam qu'ils chevauchent. Pour cela nous avons utilisé le logiciel HHsearch (version 2.0.16) [SBL05] [2]. Cet outils permet de réaliser un alignement local de deux HMM et de calculer la p-valeur associée à cet alignement. À partir de cet alignement, nous avons ensuite calculé la proportion de chevauchement en prenant la taille de l'alignement local et la taille du HMM le plus grand, voir Figure 4.4. Dans la majorité des cas (87%), l'alignement HMM-HMM a obtenu une p-valeur inférieur à 10^{-10}, indiquant que nos HMM ressemblent (au moins localement) aux HMM

2. Les comparaisons HMM-HMM sont réalisées à l'aide de la commande suivante : `hhsearch -i <hmmfile_in> -d <hmm_db> -o <resfile_out> -loc;` avec `<hmmfile_in>` un HMM en entrée, `<hmm_db>` la base de données de HMM à quoi le comparer, `<resfile_out>` le fichier de sortie qui contiendra les résultats de l'analyse, `-loc` pour rechercher le meilleur alignement local possible.

des domaines Pfam. Plus encore, dans 54% des cas, les HMM obtenus à partir de nos clusters avaient un taux de chevauchement supérieur à 80% avec les HMM des domaines Pfam.

Ensuite nous avons testé notre approche pour identifier de nouveaux domaines qui n'étaient couverts par un domaine Pfam. Comme expliqué précédemment, nous avons inclus tous les domaines Pfam connus dans notre procédure, c'est à dire que toutes les régions des protéines déjà couvertes par un domaine Pfam ont été masquées avant la recherche BLAST et nous avons utilisé la présence d'un domaine Pfam comme un hit potentiellement co-occurrent lors de l'analyse des hits co-occurrents. Nous avons ensuite lancé notre procédure avec différents seuils de e-valeur pour BLAST et pour la p-valeur pour estimer la pertinence des clusters. La Figure 4.5 présente le nombre de clusters et le FDR estimé avec ces différents seuils, pour les deux procédures d'estimation du FDR. Comme nous pouvons le constater sur cette figure, la procédure basée sur la permutation des 4-mers fournie un FDR qui converge rapidement vers 0%, ce qui semble très optimiste. Par la suite nous ne considérons donc que le FDR estimé avec les séquences renversées en utilisant un seuil de 10^{-3} pour la e-valeur de BLAST et la p-valeur de la pertinence des clusters.

Avec ces seuils, notre méthode permet d'identifier un total de 7 680 clusters (2 279 clusters avec au moins 5 hits, voir Table 4.1). Parmi les 7 680 clusters, 6 467 ont obtenu une p-valeur inférieur à 0.1% (2 240 avec au moins 5 hits) sur 3 214 protéines différentes (1 293 avec les clusters avec au moins 5 hits). Ces clusters (qui ne chevauchent aucun domaine Pfam connu) couvrent 24.61% des résidus du protéome de *Plasmodium falciparum*. Les clusters avec au moins 5 hits couvrent 6.71% des résidus. Pour comparaisons, les domaines Pfam couvrent 22% de son protéome. Le FDR de la procédure est estimé à 6.22% (1.83% pour les clusters avec au moins 5 hits). Les clusters avec une p-valeur inférieure à 0.1% rassemblent un total de 121 486 hits. Parmi ceux-ci, 39 617 ont obtenu une e-valeur modérée ($> 10^{-6}$) et n'auraient sûrement pas été considérés dans une analyse BLAST classique du protéome complet. Pour comparaison, le nombre total de hits BLAST avec une e-valeur inférieure à 10^{-6} est de 288 351. Par conséquent, avec un seuil de e-valeur à 10^{-6}, la propriété de co-occurrence nous a donc permis d'augmenter le nombre de hits considérés de 14%.

Nous nous sommes ensuite intéressés à l'origine des hits sélectionnés par notre procédure, comparée à tous les hits BLAST et à la base de données UniRef50. Chaque hit a été classé en fonction de l'espèce cible en 5 super-groupes couvrant le domaine des eucaryotes : *Chromalveolata*, *Excavata*, *Rhizaria*, *Unikont* et *Viridiplantae* [KBD+05]. La Figure 4.6 (a) présente les distributions des groupes dans l'ensemble des hits BLAST, l'ensemble des hits sélectionnés par notre procédure et l'ensemble des séquences protéiques présentes dans UniRef50. Nous observons un

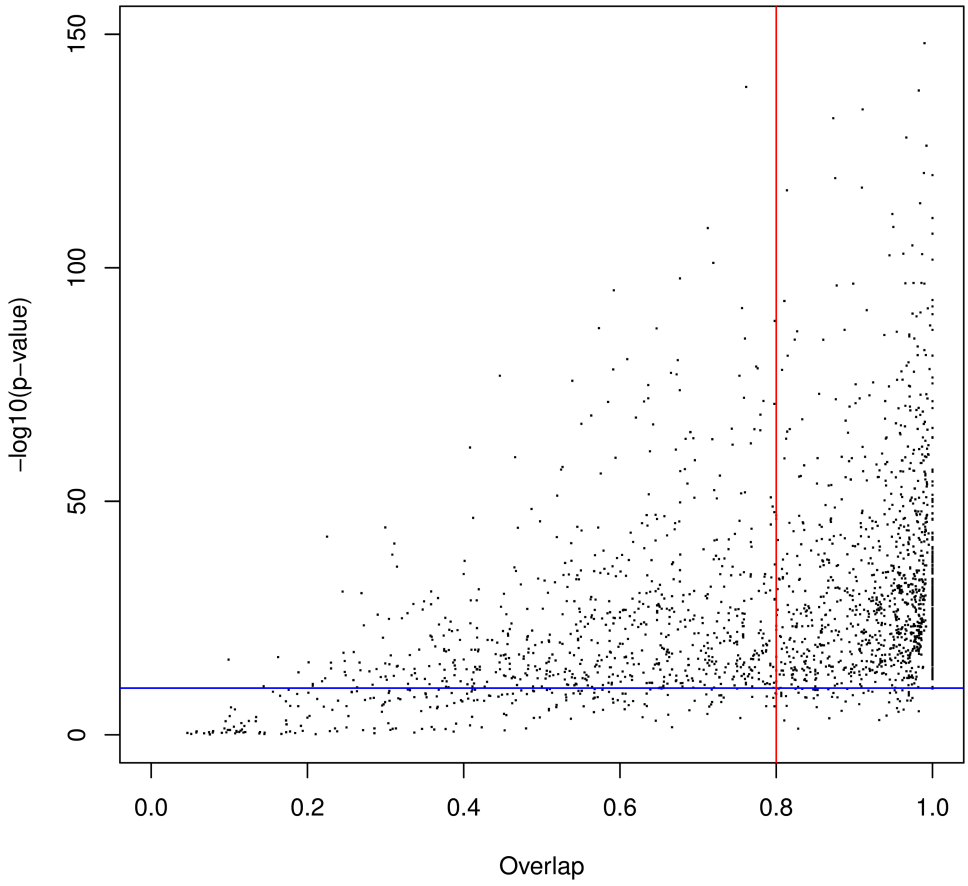
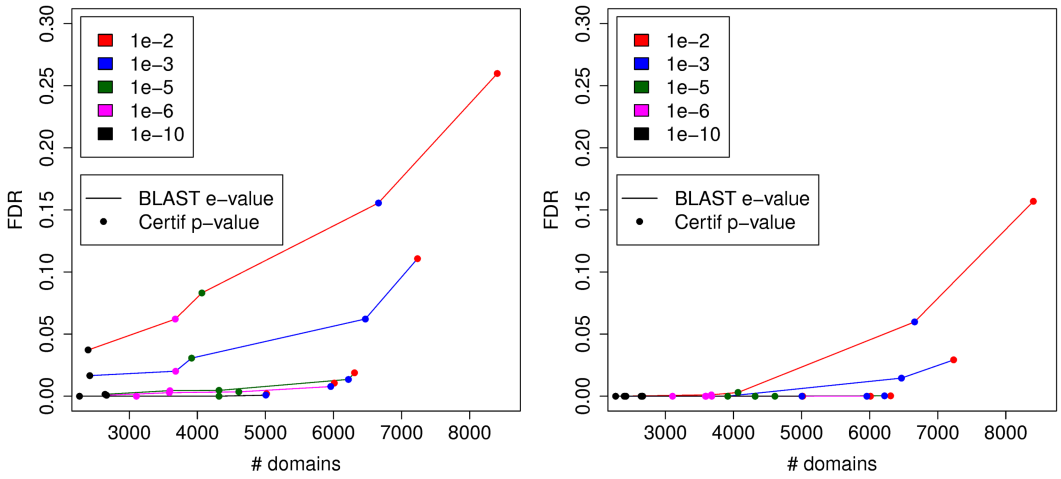


FIGURE 4.4 – Comparaison HMM-HMM des domaines identifiés par notre approche qui chevauchent un domaine Pfam connu
L'axe des abscisses montre la proportion de chevauchement de l'alignement local ; l'axe des ordonnés montre l'opposé du log10 de la p-valeur ; la ligne bleue montre une p-valeur de 10^{-10} ; la ligne rouge montre un chevauchement de 80%.



(a) (b)

FIGURE 4.5 – Nombre de domaines et FDR obtenus avec différents seuils de e-valeur et p-valeur
Cette figure présente le FDR estimé à l'aide des séquences renversées (a) et à l'aide des séquences mélangées en permutant les 4-mers (b).

léger enrichissement du groupe *Chromalveolata* dans les hits BLAST comparé à la base UniRef50, ainsi qu'un enrichissement substantiel de ce groupe dans les hits sélectionnés par notre procédure. Cela est quelque peu attendu si l'on considère que la proportion de faux positifs est sûrement plus faible dans les hits qui proviennent du même super-groupe que *Plasmodium falciparum* que dans les autres super-groupes. De plus, cela peut indiquer que certains des nouveaux domaines identifiés sont spécifiques au super-groupe *Chromalveolata*. Pour vérifier cette hypothèse, nous avons calculé la proportion de hits provenant des espèces de ce groupe dans chacun des 2 240 nouveaux domaines, voir Figure 4.6 (b). On observe que 14% des domaines ont une forte majorité ($> 90\%$) de hits provenant d'espèces du super-groupe *Chromalveolata* et que par conséquent ces domaines pourraient être considérés comme spécifiques à ce super-groupe.

4.3.1 Évaluation de la qualité des domaines

Nous nous sommes ensuite intéressés à la qualité des nouveaux domaines découverts. Dans la suite de cette section, nous utiliserons les domaines identifiés avec notre procédure en masquant les domaines Pfam déjà connus. Nous avons utilisé différentes mesures de qualité et nous avons comparé nos résultats avec ces mêmes mesures appliquées aux familles de domaines Pfam. Afin de comparer des

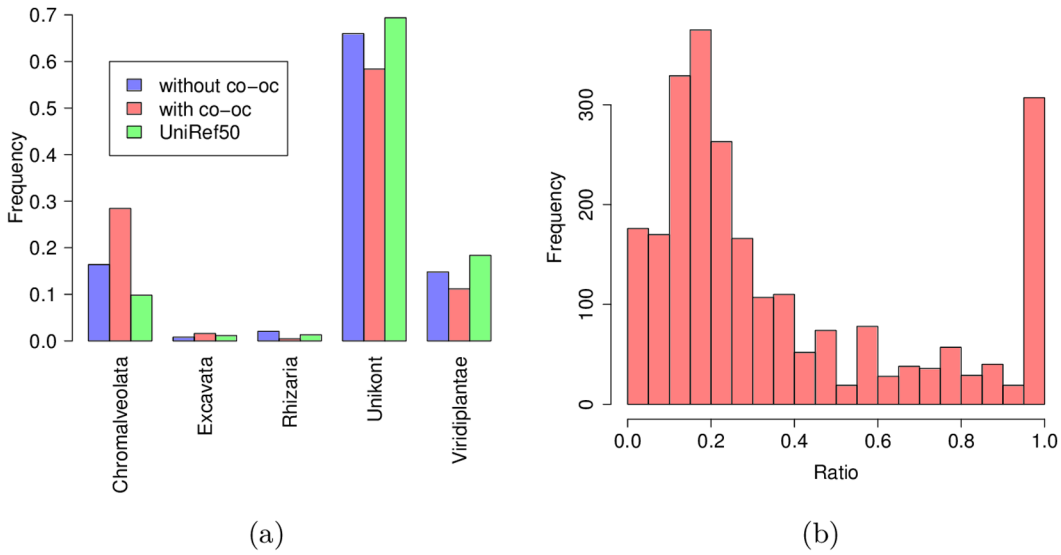


FIGURE 4.6 – Distribution des hits
(a) Distribution des hits parmi les 5 super-groupes des eucaryotes définis par Keeling *et al.* [KBD+05]. En vert, la distribution des protéines UniRef50 (restreint aux séquences eucaryotes) ; en bleu distribution de tous les hits BLAST ; en rouge distribution des hits sélectionnés par co-occurrence. (b) Distribution de la proportion de hits provenant du groupe *Chromalveolata* dans les nouvelles familles.

familles construites sur des séquences similaires, nous avons restreint les alignements de Pfam strictement aux séquences présentes dans UniRef50. De plus, pour éviter tout biais provenant de l'étape d'alignement des séquences, les familles de domaines Pfam ont été réalignées en utilisant le même outil et les mêmes paramètres que nous avons utilisé pour produire nos familles. Ensuite, afin d'évaluer les bénéfices de l'utilisation de l'information de co-occurrence, nous avons aussi réalisé ces expériences sur les domaines prédis en utilisant tous les hits et pas seulement les hits co-occurrents. Pour cela nous avons utilisé la même procédure itérative pour identifier les clusters de hits mais en utilisant cette fois tous les hits disponibles (courbe bleu sur la Figure 4.3).

Évaluer la qualité d'un alignement multiple de séquences n'est pas simple en l'absence d'autres informations. Pour cela, nous proposons d'utiliser quatre mesures : l'homogénéité, l'entropie, l'hydrophobicité de l'alignement, et la complexité des séquences. La première mesure est l'*homogénéité* de l'alignement. Un bon alignement contiendra habituellement une minorité d'indels (insertion ou suppression) à chaque position. Nous pouvons mesurer cela en observant la proportion de résidus à chaque position de l'alignement. Cette proportion devrait être soit très faible (quand peu de séquences ont une insertion à cette position) soit très forte (quand peu de séquences ont une suppression à cette position). Nous mesurons l'homogénéité par :

$$\text{Homogénéité} = \frac{1}{p \times s} \sum_{i=1}^{p} \max(r(i); \overline{r}(i)), \tag{4.3}$$

avec s et p qui représentent respectivement le nombre de séquence et la longueur du MSA, tandis que $r(i)$ et $\overline{r}(i)$ représentent respectivement le nombre de résidus et d'indels à la position i. Avec cette formule, un bon alignement devrait avoir un score qui tend vers 1.

La deuxième mesure est l'*entropie* des positions conservées (match) dans le MSA. La mesure est basée sur les classes d'acide aminés. Brièvement, les acides aminés peut être classés en fonction de leurs propriétés physico-chimique. Nous avons utilisé la définition des classes proposée dans le logiciel Seaview [GGG10]. Nous mesurons l'entropie de la manière suivante :

$$\text{Entropie} = \frac{1}{m} \sum_{i \in \text{Match}} \sum_{c \in \text{Classes}} -p_c(i) \log_2(p_c(i)), \tag{4.4}$$

avec m représentant le nombre de positions match dans le MSA. On considère une position match si le nombre de résidus à cette position est supérieur au nombre d'indels. $p_c(i)$ représente la proportion de résidus appartenant à la classe c à la position i. Dans un bon alignement, tous les résidus à une position match tendent

à appartenir à la même classe d'acide aminé et donc l'entropie tend vers 0. Dans un mauvais alignement, la distribution des classes d'acide aminés est équiprobable est donc l'entropie tend vers ≈ 3.

La troisième mesure est le score d'*hydrophobicité* :

$$\text{Hydrophobicité} = \frac{\text{Nombre de match hydrophobes}}{\text{Nombre de match}}. \tag{4.5}$$

On considère une position comme un match hydrophobe si la majorité des résidus à cette position sont considérés hydrophobes (les résidus L, A, F, W, V, M, I, P, C et G). La proportion de résidus hydrophobes est souvent utilisée comme une mesure de globularité car les domaines globulaires ont une quantité stable d'acides aminés hydrophobes (environ un tiers de la séquence) [Dil85]. Il est à noter que contrairement aux mesures d'homogénéité et d'entropie, il est difficile d'établir quel serait un «bon» score d'hydrophobicité. On utilisera cette mesure essentiellement pour comparer nos résultats et les domaines Pfam.

La dernière mesure est la *complexité* des séquences du MSA :

$$\text{Complexité} = \sum_{r \in \text{acides aminés}} -p_r \log_2(p_r), \tag{4.6}$$

avec p_r représentant la fréquence relative de l'acide aminé r dans une séquence. Contrairement aux trois mesures précédentes qui étaient basées sur les colonnes (positions) des alignements, cette mesure est appliquée à chaque séquence indépendamment. Mesurer la complexité des séquences présentes est un bon moyen d'identifier les séquences répétées qui sont caractéristiques de régions non globulaires [Woo94].

Comme nous pouvons le voir sur la Figure 4.7, les familles de domaines identifiées avec la co-occurrence obtiennent des scores de qualité relativement proches des scores obtenus par les familles de domaines Pfam. À l'inverse, les résultats obtenus sans la co-occurrence sont beaucoup plus éloignés des scores des familles de domaines Pfam. Cela illustre l'intérêt de notre procédure et montre que la co-occurrence fournie une information précieuse pour filtrer les résultats de BLAST à l'échelle d'un protéome.

4.3.2 Comparaison avec les familles de domaines connues

Nous avons ensuite cherché à évaluer si certaines des nouvelles familles de domaines ne seraient pas similaires à des familles de domaines de Pfam qui n'auraient pas encore été identifiée sur les protéines de *Plasmodium falciparum* (pour rappel les occurrences connues étaient masquées au préalable dans cette expérience). Pour cela, nous avons effectué des comparaisons HMM-HMM en utilisant

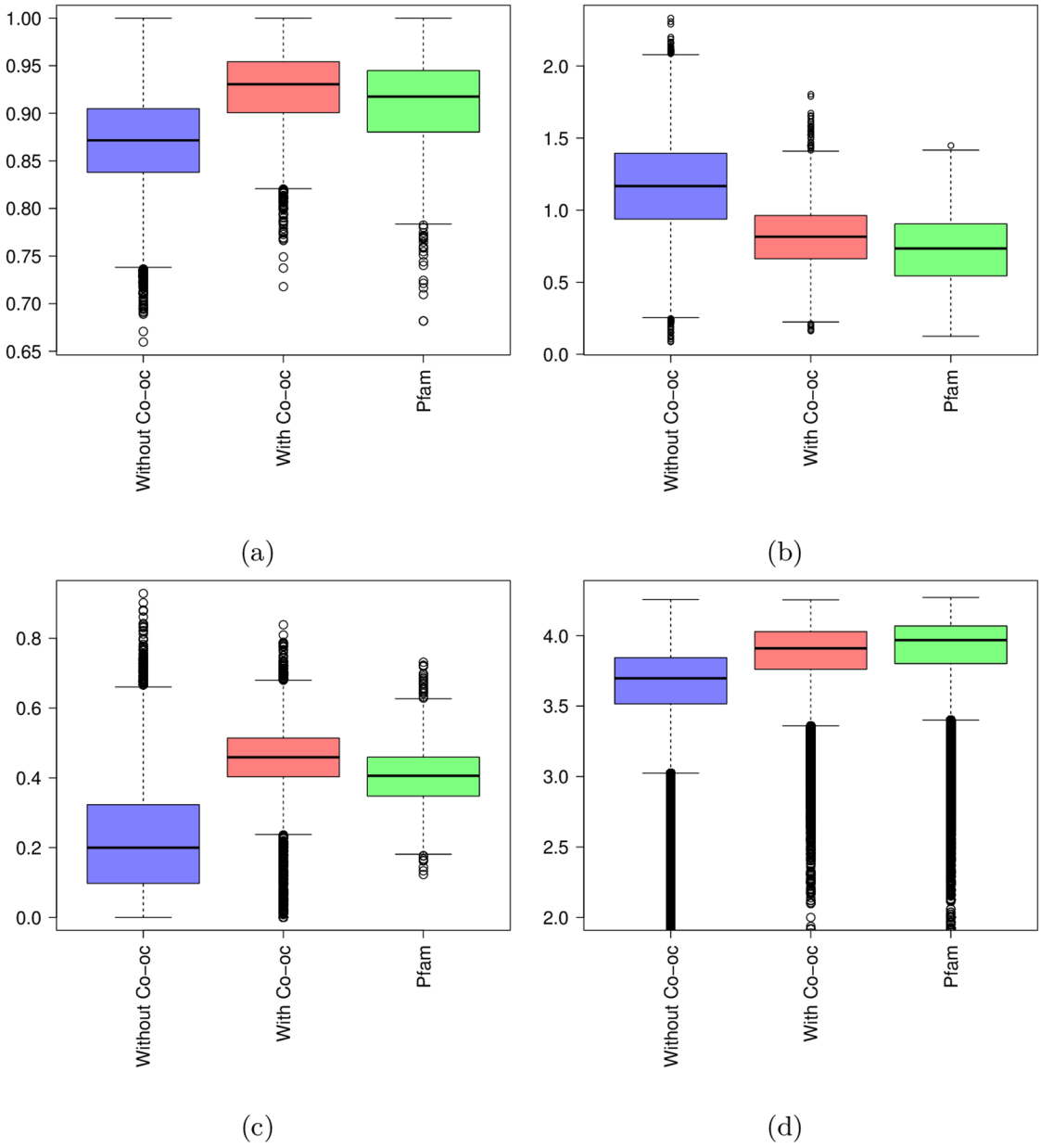


FIGURE 4.7 – Scores de qualité mesurés
Sur les familles de domaines obtenues sans la co-occurrence (en bleu), les familles de domaines obtenues avec la co-occurrence (en rouge) et les familles de domaines Pfam (en vert). (a) Homogénéité ; (b) Entropie ; (c) Hydrophobicité ; (d) Complexité des séquences.

le logiciel HHsearch [SBL05]. Comme précédemment, nous avons calculé, pour chaque alignement de HMM, une p-valeur et un taux de chevauchement entre deux HMM. Tout d'abord, nous avons lancé une comparaison de tous les HMM de Pfam contre eux mêmes. Nous avons identifié, pour chaque HMM de Pfam, quel autre HMM différent de celui ci, lui ressemblait le plus. La Figure 4.8 (a) montre les résultats obtenus pour cette analyse. Comme nous pouvons le constater, la majorité des paires de HMM ont une p-valeur supérieure à 10^{-10} et/ou un chevauchement inférieur à 80%. Ainsi, nous avons choisi d'utiliser ces deux seuils comme critères approximatifs pour décider si deux HMM sont homologues. La Figure 4.8 (b) présente les résultats obtenus pour la comparaison de nos HMM contre les HMM de Pfam. Avec les critères choisis précédemment, on observe que, sur les 2 240 nouvelles familles identifiées, environ 7% sont strictement similaires à un modèle Pfam et correspondent donc vraisemblablement à une occurrence non identifiée. On note également qu'une grande partie de nos HMM ont obtenus une p-valeur inférieur à 10^{-10}, ce qui indique qu'ils ressemblent au moins localement à une famille Pfam. Bien que ces ressemblances locales sont relativement fréquentes entre les familles Pfam (voir les nombreux points dans le coin en haut à gauche sur la Figure 4.8 (a)), il est également possible qu'une partie de ces nouveaux domaines correspondent à une occurrence partielle d'une famille Pfam. Même si les vrais occurrences partielles semblent relativement rares [PB15], il est connu que les artefacts provenant des alignements et des annotations peuvent entraîner l'observation d'une occurrence partielle d'un domaine [TP15]. Pour tester cette hypothèse, nous avons comparé la longueur des nouveaux domaines et celle du domaine Pfam qui leur ressemble le plus. Étonnamment, pour les domaines qui ont une bonne p-valeur ($< 10^{-10}$) mais un faible chevauchement ($< 80\%$) avec une famille Pfam, le nouveau domaine est plus long que le domaine Pfam dans 89% des cas. De plus, la majorité du temps, le domaine Pfam est très court (voir Figure 4.8 (c)). Nous avons également observé le même type de domaines courts dans les comparaisons de domaines Pfam-Pfam en restreignant aux paires avec une bonne p-valeur et un mauvais chevauchement (voir Figure 4.8 (c)). Ainsi, ces nouveaux domaines peuvent être interprétés comme des extensions de domaines Pfam plus courts plutôt que des domaines partiels, une observation relativement commune au sein de la base Pfam.

Pour compléter cette analyse, nous avons aussi comparé nos prédictions aux occurrences de domaines identifiées à l'aide de méthodes d'identification (et non de découverte *ab initio* comme nous) utilisant le principe de co-occurrence pour enrichir les scores de HMM de Pfam [TGMB09, OLS11, GFG+14, BVZC16, OS17]. Pour cela, nous avons utilisé les logiciels dPUC2 [OS17] et DAMA [BVZC16] sur les protéines de *Plasmodium falciparum* en utilisant les paramètres par défaut. Nous avons ensuite retiré des résultats les occurrences de domaines déjà connues,

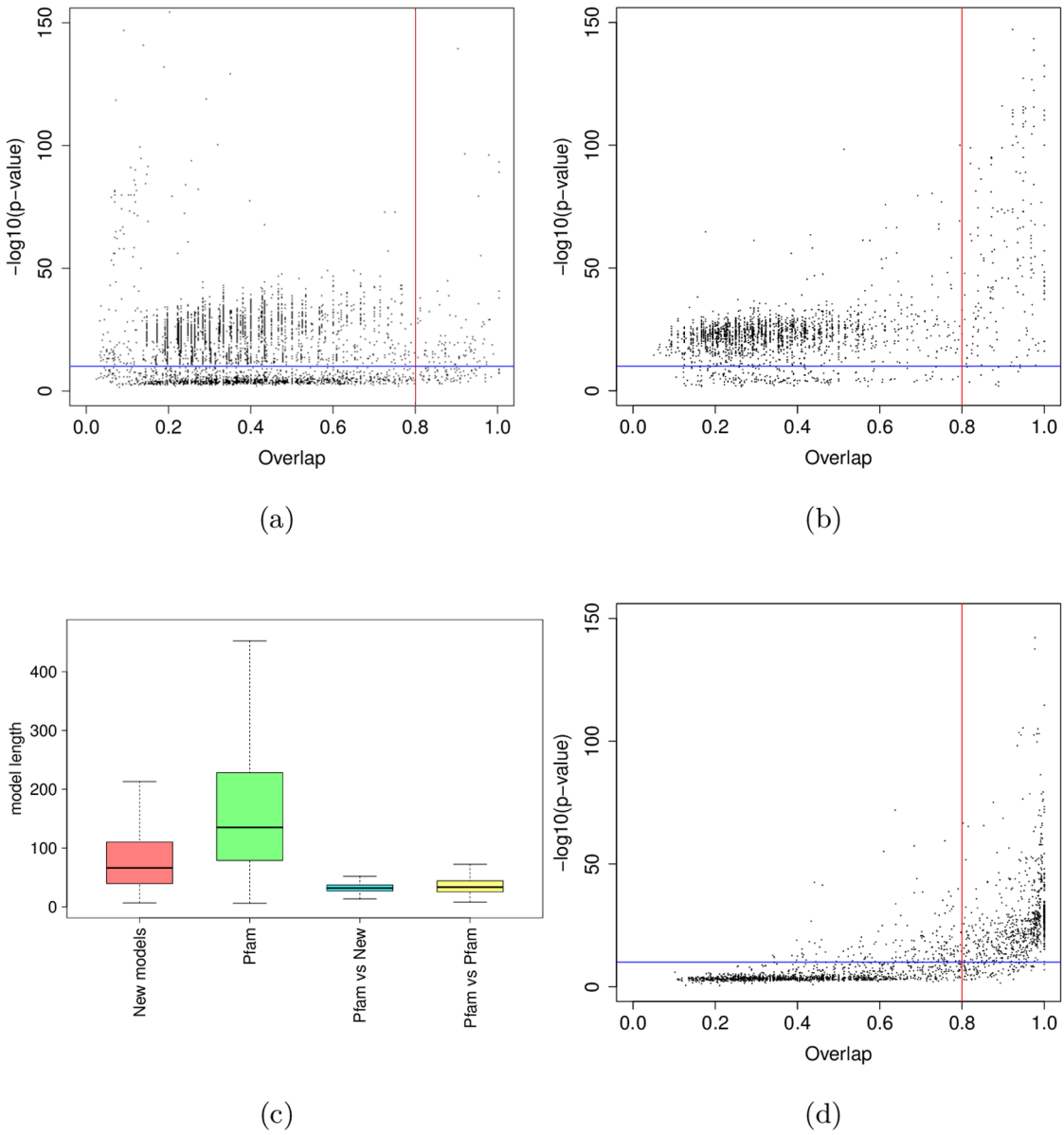


FIGURE 4.8 – Comparaisons HMM-HMM des nouvelles familles de domaines et des familles de domaines Pfam
Dans les figures (a), (b) et (d), chaque point est associé à un HMM particulier et correspond au meilleur alignement trouvé entre ce HMM et tous les autres. L'axe des abscisses montre le taux de chevauchement de l'alignement local entre les deux HMM ; l'axe des ordonnés montre l'opposé du log10 de la p-valeur de l'alignement ; la ligne bleu correspond à une p-valeur de 10^{-10} ; la ligne correspond à 80% de chevauchement. (a) comparaisons Pfam contre Pfam ; (b) comparaisons nouvelles familles contre Pfam. La figure (c) montre la longueur des domaines obtenus par notre approche (rouge), de tous les domaines Pfam (vert), la longueur des domaines Pfam associés aux points dans le cadre en haut à gauche de la figure (b) (bleu), la longueur du plus petit modèle Pfam associé aux points dans le cadre en haut à gauche de la figure (a) (jaune) ; (d) comparaisons des nouvelles familles contre elles-mêmes.

et nous avons calculé la couverture et le FDR associé aux nouvelles occurrences de domaines identifiées par ces approches. DAMA et dPUC2 permettent de prédire 1 376 et 1 039 nouveaux domaines qui ne chevauchent pas de domaine déjà connu (*i.e.* moins de 30 résidus ou 50% de chevauchement), respectivement. Les FDR estimés sont de 4% et 2% respectivement. Sur les 2 240 nouveaux domaines avec plus de 5 hits identifiés par notre procédure, 618 (27%) et 482 (21%) chevauchent d'au moins 30 résidus (ou 50%) un domaine de DAMA ou dPUC2, respectivement. Cependant, il est à noter que ces chevauchement sont souvent partiels et ne peuvent donc pas être considérés comme des domaines strictement similaires, ce qui explique la différence avec les 7% estimés au dessus.

Nous nous sommes ensuite posé la question de savoir si les nouveaux domaines identifiés par notre procédure pouvaient être simplement des prolongations des occurrences Pfam connus et non de vrais domaines à part entière. Pour répondre à cette question, nous avons calculé pour chaque domaine, la distance au domaine Pfam le plus proche sur les protéines de *Plasmodium falciparum*. Nous avons également calculé la distance entre le hit BLAST et le même domaine Pfam sur les protéines cibles d'UniRef50. Pour comparaison, nous avons aussi calculé les distances entre deux domaines Pfam sur un protéome bien annoté (*Saccharomyces cerevisiae*). Les résultats sont représentés sur la Figure 4.9. Comme nous pouvons le constater, les hits qui composent les nouvelles familles de domaines ne sont pas particulièrement proches des domaines Pfam connus sur les protéines (la médiane est d'environ 50 résidus), et sont plutôt supérieures aux distances mesurées chez la levure.

4.3.3 Redondance des nouvelles familles

Nous avons ensuite voulu estimer la proportion de familles redondantes parmi nos prédictions, c'est à dire que nous avons vérifié si certaines de nos nouvelles familles étaient similaires à une autre famille prédite (la même famille peut être identifiée plusieurs fois dans un protéome donné). Pour tester cette hypothèse, nous avons comparé nos modèles deux à deux à l'aide du logiciel HHsearch (voir Figure 4.8 (d)). 1 454 (65%) des nouveaux modèles ne semblent pas avoir de fortes similarités avec les autres modèles, tandis que 786 (35%) modèles semblent similaires à au moins un autre modèle. Cette proportion de modèles redondants est plus élevée que dans la base Pfam mais est tout à fait attendu au regard du fait que les familles de domaines ont souvent plusieurs occurrences dans un protéome [VTPL05]. Nous avons ensuite essayé de regrouper les familles similaires entre elles. Pour cela, nous avons construit un graph où chaque famille est représentée par un nœud et la similarité entre deux familles est représentée par une arrête. La procédure de clustering revient alors à isoler chaque composante connexe du graph. De cette manière, nous obtenons 1 645 familles différentes. Cette estimation

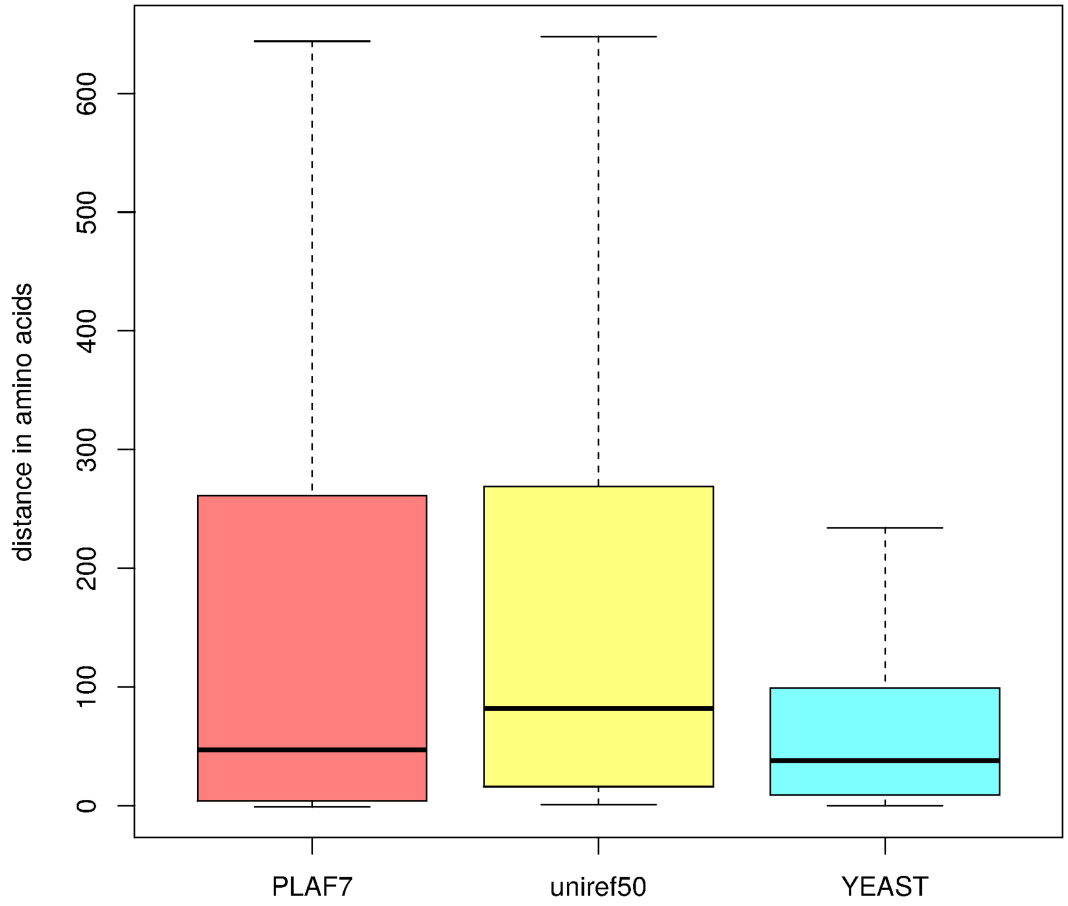


FIGURE 4.9 – Distances (en nombre de résidus) entre les domaines adjacents
En rouge : les distances entre les hits BLAST des nouvelles familles de domaines identifiées et le domaine Pfam le plus proche dans les protéines de *Plasmodium falciparum* (seules les protéines annotées par un domaine Pfam ont été considérées) ; en jaune : les distances entre les hits BLAST et le même domaine Pfam dans les protéines de UniRef50 associées ; en bleu : les distances entre les domaines Pfam adjacents dans le protéome de *Saccharomyces cerevisiae*.

est certainement sous-estimée car cette procédure de clustering a pu regrouper des familles qui n'avaient qu'une similarité indirecte au travers d'une autre famille.

4.3.4 Comparaison avec les autres bases de domaines générées automatiquement

Nous avons comparé les résultats obtenus avec notre approche aux bases Pfam-B et ProDom [Ser02], deux bases de données de domaines générées automatiquement. Comme mentionné au chapitre précédent, Pfam-B faisait parti de la base Pfam jusqu'à la version 26. Jusqu'à cette version, la base Pfam contenait deux types de familles de domaines : les familles de grande qualité et vérifiées manuellement Pfam-A (et qui correspondent aux familles simplement appelées Pfam dans les sections précédentes), et les familles Pfam-B générées automatiquement à l'aide de l'algorithme ADDA [HH03] sur la base de toutes les séquences Uniprot disponibles non déjà couvertes par un domaine Pfam-A. ProDom est une base de données de familles de domaines construite automatiquement sur la base d'un clustering de segments homologues à l'aide de l'algorithme MKDOM2, lui-même basé sur une recherche récursive PSI-BLAST contre la base Uniprot. Ainsi, les familles de Pfam-B et ProDom sont constituées de l'alignement de sous-séquences provenant de Uniprot. Parmi les 460 125 familles contenues dans Pfam-B (version 26), nous avons trouvé 651 familles avec au moins 5 protéines différentes de la base UniRef50. Ces 651 familles couvrent environ 9.70% du protéome de *Plasmodium falciparum*. La base ProDom (version décembre 2015) contient de nombreuses petites familles (moins de 30 résidus de long). Par soucis de cohérence avec nos expériences et la base Pfam-B, nous avons choisi d'ignorer ces petites familles. Parmi les 3 739 157 familles contenues dans ProDom, nous avons trouvé 1 453 familles de plus de 30 résidus, avec au moins 5 protéines de la base UniRef50, une protéine de *Plasmodium falciparum*, et qui ne chevauchent pas un domaine Pfam-A. Ces familles couvrent environ 3.99% du protéome de *Plasmodium falciparum*. Pour comparaison, les clusters avec au moins 5 hits identifiés par notre procédure couvrent 6.71% des résidus de *Plasmodium falciparum*. Ainsi, en terme de couverture, notre approche (6.71%) se situe entre ProDom (3.99%) et Pfam-B (9.70%).

Les 651 familles de Pfam-B et les 1 453 familles de ProDom ont été restreintes aux séquences provenant de *Plasmodium falciparum* et UniRef50, et elles ont été réalignées en utilisant la même procédure que nous avons utilisée pour générer nos modèles. Ensuite, nous avons calculé les différentes mesures de qualité pour chaque procédure (voir Figure 4.10)). Comme nous pouvons l'observer, les familles de ProDom, et surtout de Pfam-B, obtiennent des scores d'homogénéité assez faibles comparés aux scores obtenus par les familles de Pfam-A. Cela illustre le fait que les familles de ces bases de données incluent parfois des séquences très

différentes qui ne peuvent pas être alignées et qui devraient donc être retirées de ces familles. À l'inverse, ils ont de bons scores d'entropie, meilleurs même que ceux de Pfam-A. Cependant ces bons scores sont vraisemblablement une conséquence directe du faible nombre de séquences qui composent ces familles (voir Figure 4.10 (e)). Pour Pfam-B par exemple, 80% des familles considérées ont moins de 15 séquences dans UniRef50. Ainsi, il semblerait que beaucoup de familles contiennent uniquement des séquences fortement similaires, ce qui permet d'obtenir un bon score d'entropie mais implique également un manque de diversité. À l'inverse, en utilisant l'information de co-occurrence, notre approche permet de sélectionner des séquences plus diverses tout en limitant l'introduction de faux positifs. En terme d'hydrophobicité, ProDom obtient des scores comparables à notre approche et Pfam-A, tandis que Pfam-B obtient des scores plus faibles. Enfin, ProDom et Pfam-B ont des séquences de complexité comparable mais légèrement inférieure à celles de notre approche et Pfam-A. En prenant en compte tout ceci, ces résultats suggèrent que les domaines construits en utilisant l'information de co-occurrence sont globalement de meilleur qualité que ceux trouvés dans Pfam-B et ProDom. Il est à noter cependant, que l'objectif de ces bases de données est différent et plus large que le notre. En effet, leur objectif est de construire une base qui rassemble toutes les familles de domaines de toutes les espèces, alors que notre objectif est de se concentrer sur la découverte de nouvelles occurrences de domaines pour une espèce donnée (ici *Plasmodium falciparum*).

Finalement, nous avons comparé directement nos nouveaux domaines aux familles de Pfam-B. Comme précédemment, nous avons utilisé HHsearch pour comparer les HMM appris sur les familles avec plus de 5 hits aux HMM fournis par Pfam-B. Nous pouvons observer les résultats Figure 4.11. Comme nous pouvons le constater, la majorité (95%) ont une faible similarité avec les familles de Pfam-B (p-valeur $> 10^{-10}$ et chevauchement < 0.8).

4.3.5 Test contre les versions précédentes de Pfam

Comme dernier test, nous avons lancé notre analyse en utilisant une ancienne version de Pfam (version 26) et nous avons calculé le nombre de nouveaux domaines provenant de la version 28 que nous pouvons retrouver. Sur les 92 nouvelles familles de domaines de Pfam 28 avec au moins une occurrence sur *Plasmodium falciparum*, nous avons identifié un domaine sur la même région pour 54 d'entre elles. En inspectant les familles identifiées par notre approche, nous trouvons souvent une forte similarité avec les nouvelles familles de Pfam 28. Par exemple, le domaine identifié sur la protéine O77317_PLAF7 montre une forte similarité avec la famille PF16876 de la version 28 (p-valeur $= 4 \times 10^{-31}$ et chevauchement 0.99).

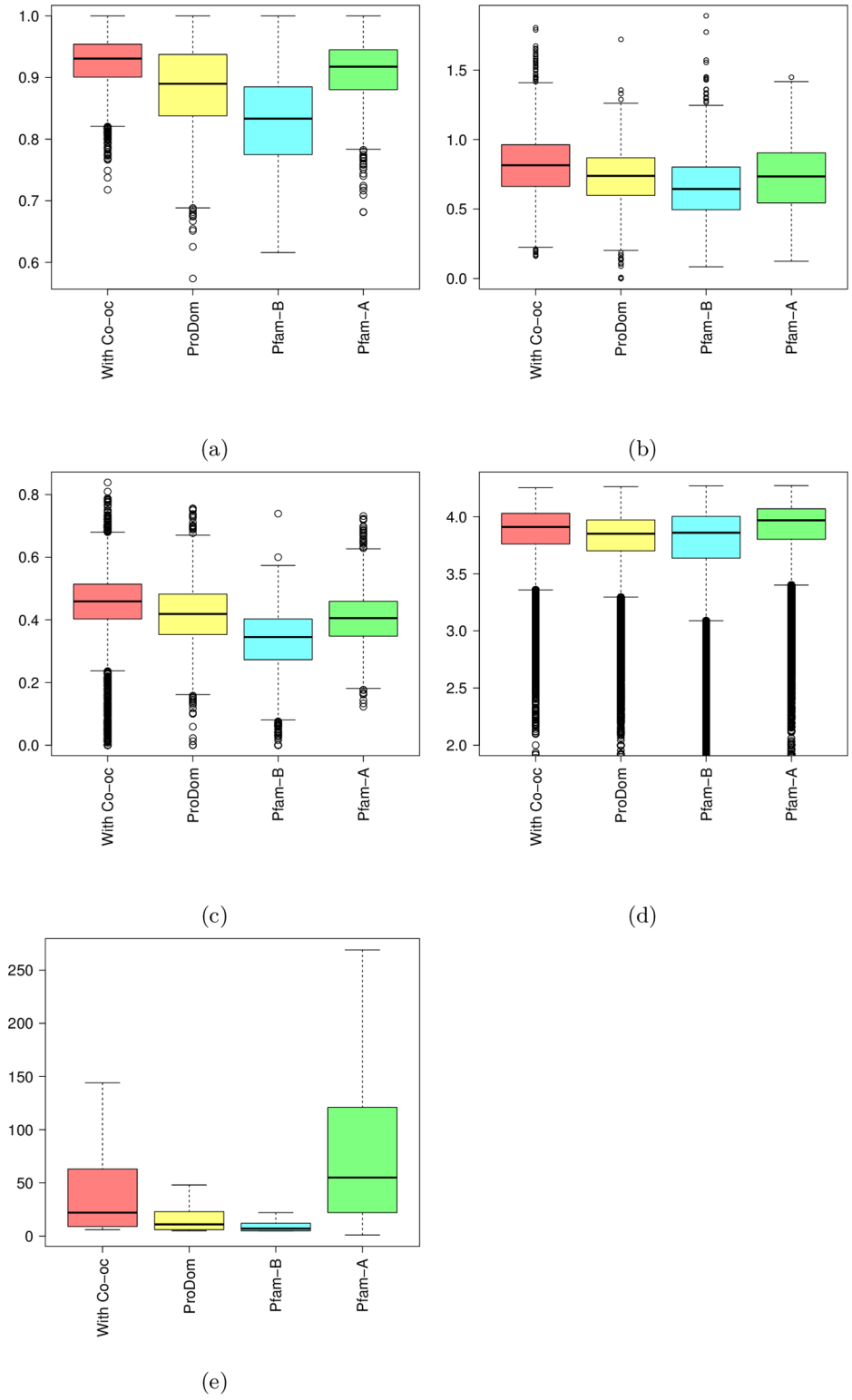


FIGURE 4.10 – Scores de qualités contre d'autres bases de données générées automatiquement
Scores de qualité (a-d) mesurés sur les familles obtenues par notre approche (rouge), les familles de ProDom (jaune), les familles Pfam-B (cyan) et les familles de Pfam-A (vert). (a) Homogénéité ; (b) Entropie ; (c) Hydrophobicité ; (d) Complexité des séquences ; La figure (e) montre le nombre de séquences par famille dans les différentes bases de données.

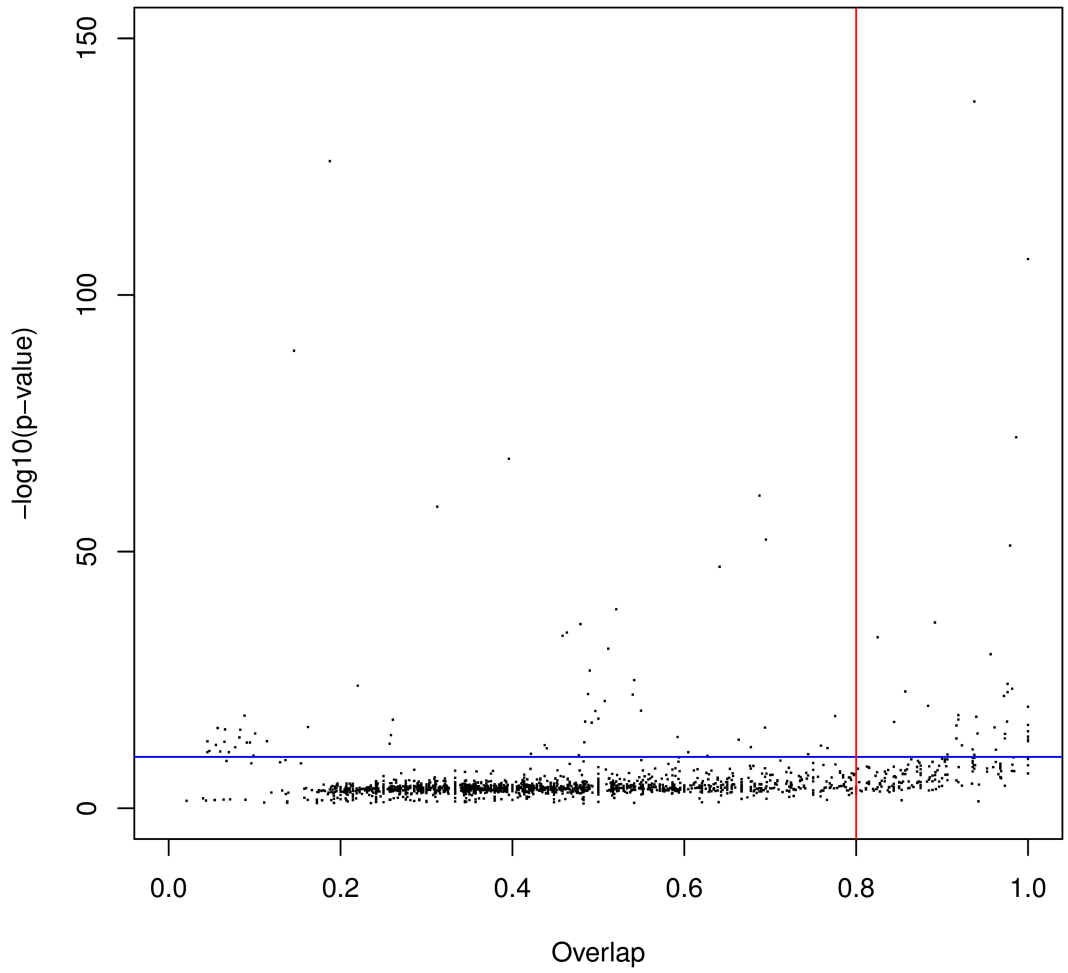


FIGURE 4.11 – Comparaisons HMM-HMM des nouvelles familles de domaines contre les familles de domaine de Pfam-B
Chaque point représente un des nouveaux HMM et correspond au meilleur alignement local obtenu contre tous les HMM de Pfam-B. L'axe des abscisses montre le taux de chevauchement ; l'axe des ordonnés montre l'opposé du log de la p-valeur de l'alignement ; la ligne bleue correspond à $y = -\log 10^{-10}$; la ligne rouge correspond à $x = 0.8$.

4.3.6 Annotations fonctionnelles

Le Consortium Gene Ontology (GO) [ABB+00] fourni un vocabulaire structuré décrivant la fonction des gènes suivant trois points de vue (processus biologique, fonction moléculaire, composant cellulaire). Chaque ontologie est organisé sous forme d'un graph acyclique orienté où chaque nœud est associé à un terme et chaque arrête décrit la relation de généralisation ou de spécialisation entre les termes. Le Consortium GO fourni également la liste des annotations des protéines de la majorité des espèces séquencées. Avec cette information, nous avons essayé d'associer une annotation à chacune de nos nouvelles familles de domaines avec au moins 5 hits. Pour chaque famille, nous avons utilisé le HMM appris avec celle ci pour scanner toutes les protéines de UniRef50 à la recherche de nouvelles occurrences. Toutes les protéines avec une p-valeur inférieur à 10^{-10} sont ajoutées à l'ensemble des séquences présentes dans la famille. Nous avons ensuite récupéré toutes les annotations associées à chacune de ces protéines (à l'exception de la protéine de *Plasmodium falciparum*). Nous avons ensuite parcouru la GO, et si plus de 95% des protéines annotées partagent un même terme GO, nous pouvons annoter la famille avec ce terme. Nous avons choisi de fixer une limite d'au moins 5 protéines annotées par famille pour éviter les annotations non pertinentes. Les termes GO qui descendent directement de la racine du graph ne sont pas considérés comme des annotations pertinentes et sont donc ignorés. Parmi les 2 240 nouvelles familles de domaines, nous pouvons proposer une nouvelle annotation pour 1 366 d'entre elles. Parmi celles ci, pour 1 195 familles, l'annotation proposée est en accord avec les annotations déjà connues de la protéine de *Plasmodium falciparum* où nous avons identifié le domaine. Pour 171 familles, l'annotation proposée permet d'étendre les annotations connues.

Pour finir, nous avons vérifié si les nouvelles familles redondantes (*i.e.* les 786 nouvelles familles qui ressemblent à une autre des nouvelles familles) obtenaient des annotations GO similaires. Pour répondre à cela, nous avons comparé les annotations obtenues sur des pairs de familles similaires et les annotations obtenues sur des pairs de familles différentes. Pour cela, nous avons utilisé l'indice de Jaccard pour mesurer la similarité entre deux ensemble d'annotations GO. Nous pouvons observer les résultats sur la Figure 4.12. Comme nous pouvons le constater les familles similaires obtiennent un score très proche de 1 tandis qu'en prenant des familles différentes on obtient beaucoup plus de diversité dans les annotations.

4.3.7 Complexité algorithmique et temps de calcul

Si n est le nombre de hits retournés par BLAST pour une protéine de l'organisme étudié, l'identification des hits co-occurrents et le calcul de la densité des hits co-occurrents (la courbe rouge) s'effectuent en $\mathcal{O}(n \log n)$ opérations. Ensuite,

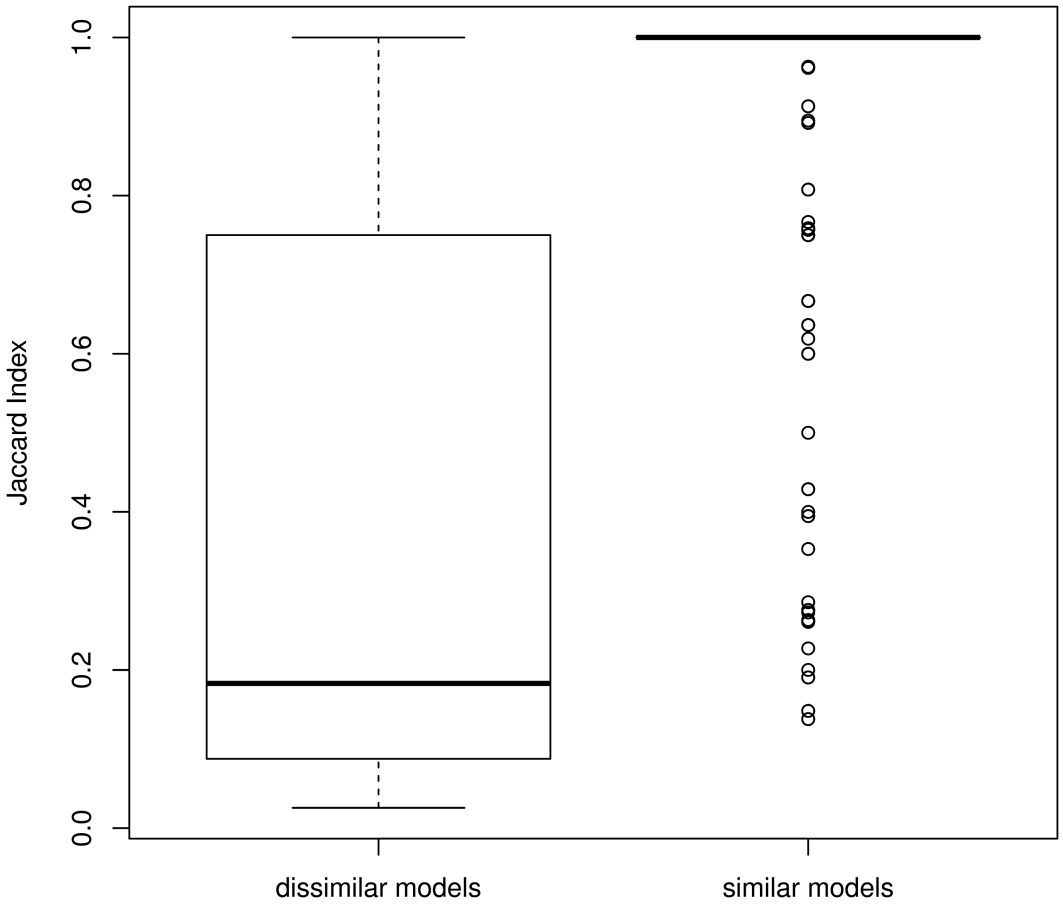


FIGURE 4.12 – Similarité entre les annotations GO de pairs de familles
À gauche : la similarité d'annotation pour des familles différentes ; à droite : la similarité d'annotation entre des familles de domaines identifiées comme similaires dans la Figure 4.8 (d).

si n' est le nombre de hits co-occurrents sur la protéine, l'Algorithme 3 s'effectue en $\mathcal{O}(n' \times q)$ opérations, avec q le nombre de clusters sur la protéine. q est supposé relativement faible (moins d'une dizaine) et $n' \ll n$, donc la complexité en temps de toute la procédure est de $\mathcal{O}(n \log n)$. Pour les expériences sur le protéome de *Plasmodium falciparum* et avec un ordinateur portable standard (core i7, 8Go RAM), avec un seuil de e-valeur de 10^{-3} pour BLAST, l'identification des hits co-occurrents pour toutes les protéines prend environ 7 minutes et l'identification des clusters prend 29 secondes. Les calculs de p-valeur prennent 33 secondes. Si l'on prend également en compte le temps nécessaire aux opérations de lecture et écriture des fichiers, l'analyse des résultats de BLAST pour l'ensemble du protéome de *Plasmodium falciparum* prend environ 17 minutes.

4.4 Discussion

Nous avons proposé une nouvelle méthode pour prendre en compte l'information de co-occurrence des domaines dans une analyse BLAST classique et ensuite construire de nouvelles familles de domaines sur la base de ces résultats. Notre méthode est basé sur l'analyse de la densité des hits co-occurrents le long d'une protéine cible. Nous avons conçu une procédure de clustering pour identifier les clusters de hits similaires qui indiquent la présence d'un domaine conservé ainsi qu'un test statistique pour évaluer la pertinence des clusters identifiés. Nous avons également présenté une procédure pour estimer la proportion de faux positifs parmi un ensemble de clusters.

Nous avons choisi l'organisme *Plasmodium falciparum* comme étude de cas afin d'évaluer notre procédure. Nos expériences ont montré une augmentation de 14% du nombre de hits significatifs ainsi qu'une augmentation de 25% du taux de couverture du protéome. Nous avons identifié 6 467 clusters significatifs. Le taux de faux positifs est estimé aux alentours de 6% (3% pour les clusters avec au moins 5 hits). Nous avons utilisé ces clusters pour construire de nouvelles familles de domaines qui pourront enrichir les bases de données des domaines connus. Ces modèles ont montré une qualité proche des modèles Pfam et relativement peu de redondance avec les familles Pfam connues, ce qui semble indiquer qu'il s'agit vraisemblablement de nouvelles familles de domaines qui ne sont pas présentes dans la base Pfam. A l'inverse, nous avons identifié plus de redondances parmi les nouveaux modèles, ce qui semble indiquer la présence de plusieurs occurrences provenant d'une même famille de domaines dans le protéome de *Plasmodium falciparum*.

Notre approche pourrait être améliorer de plusieurs façons. Tout d'abord, l'étape de clustering des hits est une procédure *ad-hoc* qui implique l'utilisation de deux paramètres déterminés empiriquement. Une amélioration intéressante se-

rait d'intégrer une procédure automatique capable de trouver les valeurs optimales de ces paramètres compte tenu de l'organisme étudié. Ensuite, la qualité des HMMs générés dépend fortement des paramètres des logiciels utilisés (ici BLAST et MUSCLE). Nous avons utilisé les paramètres par défaut dans cette étude, cependant d'autres paramètres pourraient certainement aider à construire de meilleurs modèles. C'est particulièrement vrai pour les espèces comme *Plasmodium falciparum* qui ont une distribution des acides aminés relativement éloignée de la distribution classique observée chez les autres espèces. Pour finir, un inconvénient majeur de notre approche est qu'elle ne peut pas annoter toutes les protéines d'une espèce cible. Dans le cas de *Plasmodium falciparum*, notre procédure n'est pas capable d'identifier un seul cluster significatif pour 2 151 protéines. La raison principale est qu'il existe des protéines mono-domaine pour lesquelles la co-occurrence n'est d'aucune aide. Dans ce cas, il serait intéressant d'identifier d'autres composantes qui pourraient remplacer les domaines dans l'analyse de co-occurrence. Nous pouvons citer par exemple la présence de séquences répétées en tandem ou la présence de régions de faible complexité qui constitueraient de nouvelles classes de séquences conservées et qui pourraient être utile pour prédire la présence d'un domaine.

Troisième partie

Découverte des domaines de régulation

Chapitre 5

Méthodes de prédiction de l'expression à partir de l'ADN

Comme on l'a vu au Chapitre 2, l'expression des gènes chez les eucaryotes fait intervenir différents mécanismes, associés à diverses régions régulatrices, qui permettent d'assurer une grande variété de types et de fonctions cellulaires. Un défi de la biologie moderne consiste à identifier quelles régions régulatrices sont actives, quelles sont leurs caractéristiques et comment elles fonctionnent ensemble et individuellement. Plusieurs approches de bioinformatique ont abordé ce problème en modélisant l'expression des gènes sur la base des marques épigénétiques. L'objectif de ces travaux est d'étudier les liens existant entre différentes marques épigénétiques et le niveau d'expression des gènes. Pour cela, une fonction de prédiction est apprise sur la base d'un sous-ensemble de gènes d'apprentissage, et la précision est ensuite évaluée sur un ensemble de test indépendant. Une analyse *a posteriori* des variables utilisées pour la prédiction permet alors d'identifier les régions qui semblent le plus étroitement liées à l'expression. Parmi ces travaux on peut citer [PRT02, SYK03, CLN$^+$16, DCJ$^+$17, LLZ14, SGG$^+$17]. Cependant, ces modèles sont fondés sur des données expérimentales qui se limitent à des échantillons spécifiques (souvent à quelques lignées cellulaires) qui peuvent être difficile à obtenir, suivant les espèces et les conditions. De plus, ces méthodes ne sont généralement pas conçues pour capturer les instructions de régulation présentes au niveau de la séquence, avant même la fixation des facteurs de transcription ou l'ouverture de la chromatine.

La recherche des instructions de régulation au niveau de la séquence ADN est un champ de recherche relativement ancien mais qui connaît un nouvel engouement du fait de la disponibilité des données expérimentales ainsi que du développement de méthodes d'analyse basées sur l'apprentissage statistique. Dans ce chapitre, nous allons présenter quelques méthodes qui ont été développées spécifiquement pour ce problème et les résultats obtenus pour prédire l'expression des

gènes dans différentes espèces. Nous présenterons ensuite la seule étude publiée (à notre connaissance) de prédiction de l'expression des gènes chez *Plasmodium falciparum*.

5.1 Méthodes basées sur de la régression linéaire

Dans cette section nous allons présenter quelques méthodes de prédiction de l'expression des gènes à l'aide d'un modèle simple de régression linéaire tel qu'on l'a présentée au Chapitre 1 :

$$y(g) = a + \sum_i b_i x_{i,g} + e(g), \tag{5.1}$$

où $y(g)$ est l'expression du gène g, $x_{i,g}$ est une variable i associée à g, $e(g)$ est l'erreur résiduelle associée à g, a est l'ordonnée à l'origine, et b_i est le coefficient de régression associé à la variable i. Nous avons recensé deux méthodes de prédiction de l'expression basées sur ce modèle.

5.1.1 Approche REDUCE

Les auteurs de Bussemaker *et al.* [BLS01] présentent un algorithme (nommé REDUCE) qui permet d'identifier des motifs qui semblent liés à l'expression des gènes. Pour cela, leur algorithme énumère un certains nombre de k-mers (k $\in$ [2 : 7]) et mesure la relation entre la fréquence d'un k-mer et l'expression à l'aide d'une corrélation. Ensuite vient une procédure itérative de raffinement des k-mers qui permet de générer de nouveaux motifs. Le principe de cette procédure est de sélectionner les k-mers avec la plus forte corrélation et de les modifier de toutes les manières possible en utilisant les symboles IUPAC (voir Table 5.1) pour générer de nouveaux motifs. Ensuite, ils réalisent un alignement des occurrences de ces nouveaux motifs et produisent une PWM (voir Section 1.3.2) qu'ils corrigent en ajoutant un pseudo-compte de 1. Pour finir, ils construisent un modèle linéaire utilisant comme variables prédictives la fréquence des occurrences de chaque PWM dans les 600bp avant le TSS pour prédire l'expression des gènes. Ils ont évalué leur procédure sur différentes conditions de *Saccharomyces cerevisiae*. Leur procédure permet la création de 192 motifs, et le prédicteur appris sur la base de ces variables a une corrélation entre les valeurs d'expression prédites et observées d'environ 30%.

5.1.2 Régression linéaire avec une pénalisation LASSO

Les auteurs de Bessière *et al.* [BTP+18] ont développé un modèle de régression global pour prédire l'expression des gènes en utilisant la composition dinucléotidique de différentes régions. Les auteurs ont ainsi distingués les régions 5' UTR et

code	signification
A	adénine
C	cytosine
G	guanine
T	thymine
R	A ou G
Y	C ou T
S	G ou C
W	A ou T
K	G ou T
M	A ou C
B	C ou G ou T
D	A ou G ou T
H	A ou C ou T
V	A ou C ou G
N	A ou C ou G ou T

TABLE 5.1 – Liste des symboles IUPAC

3' UTR, les introns, les exons, ainsi que 3 sous-régions différentes des promoteurs : une région core-promoteur située entre -500bp et +500bp autours du TSS, une région upstream (entre -2000 et -500bp) et une région downstream (entre +500 et +2000bp) (voir Figure 5.1). Ces régions ont été définies sur la base de connaissances préalables et d'expériences de prédictions préliminaires. Cette méthode a été testée sur plusieurs séries de données chez l'Homme provenant des consortiums *The Cancer Genome Atlas* (`https://www.cancer.gov/tcga`) et ENCODE [DHS+18]. Pour chaque série de données, un modèle de régression linéaire avec pénalisation LASSO est appris. *A posteriori* l'analyse des coefficients non nuls du modèle permet alors d'identifier les variables importantes pour chaque série de données. Lorsqu'il est limité aux variables issues des 3 régions promotrices, leur modèle présente une précision similaire à celle de deux méthodes indépendantes basées sur des données expérimentales [LLZ14, SGG+17]. Cela a ainsi confirmé l'importance de la composition en nucléotides dans la prédiction de l'expression des gènes. De plus, la performance de l'approche augmente lorsqu'on ajoute aux variables des promoteurs les variables calculées sur les UTR, les exons et les introns. Un des résultats les plus surprenant de ce travail est que le corps du gène (introns, CDS et UTR), semble avoir la contribution la plus significative dans le modèle devant celle des variables issues des régions promotrices. Une explication avancée par les auteurs est que la composition nucléotidique du corps du gène

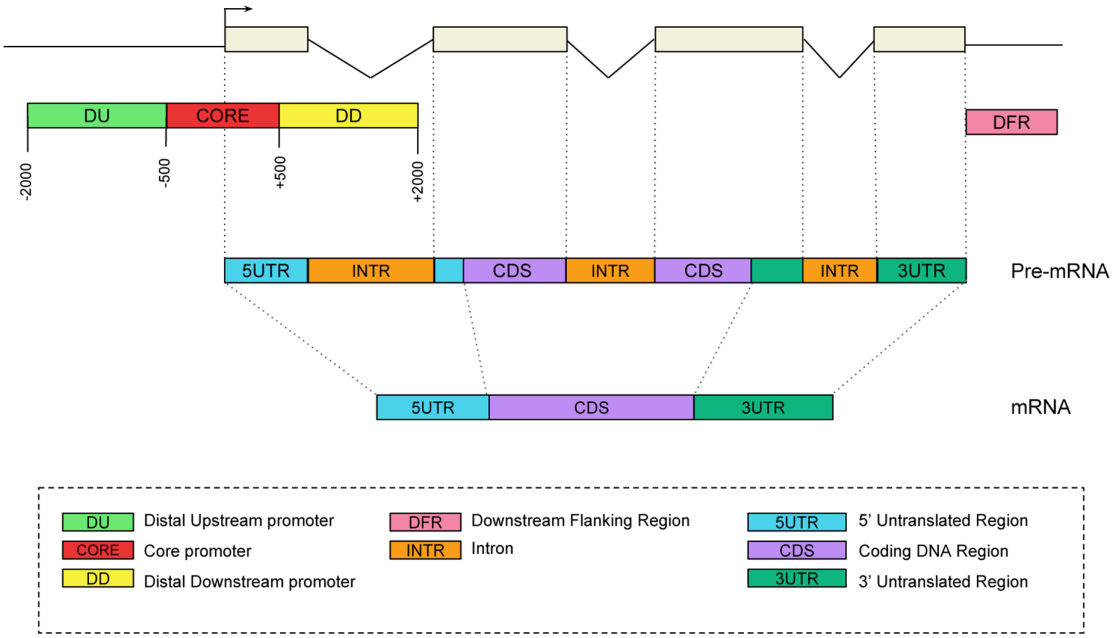


FIGURE 5.1 – Architecture des promoteurs humains
Segmentation proposée par les auteurs de Bessière *et al.*. Source : [BTP+18]

semble liée aux TAD (*Topologically Associated Domains*) qui gouvernent la structure tridimensionnelle de la chromatine et est connue pour avoir également un lien fort avec l'expression chez les vertébrés.

5.2 Méthodes basées sur les réseaux de neurones

Dans cette section nous allons présenter les méthodes récentes à base de réseaux de neurones pour prédire l'expression des gènes. Un réseau de neurones est une méthode statistique d'apprentissage automatique qui s'inspire directement de l'architecture des neurones biologiques. Chaque neurone est un objet mathématiques qui prend en entrée une série de données pour calculer une valeur de sortie. Les neurones sont généralement connectés entre eux sous forme de couche pour former un réseau (Figure 5.2). Le choix de l'architecture et les techniques et méthodes d'apprentissage des paramètres du réseau est un domaine en plein essor que nous ne détaillerons pas ici. Mais il faut savoir que les réseaux de neurones sont souvent vus comme des boites noires capables d'apprendre des relations complexes. On peut voir ces modèles comme une généralisation des modèles de régression linéaire, chaque fonction implémentée dans un neurone étant une fonction linéaire suivie d'une fonction de seuillage destinée à s'affranchir des contraintes de linéarité. Les

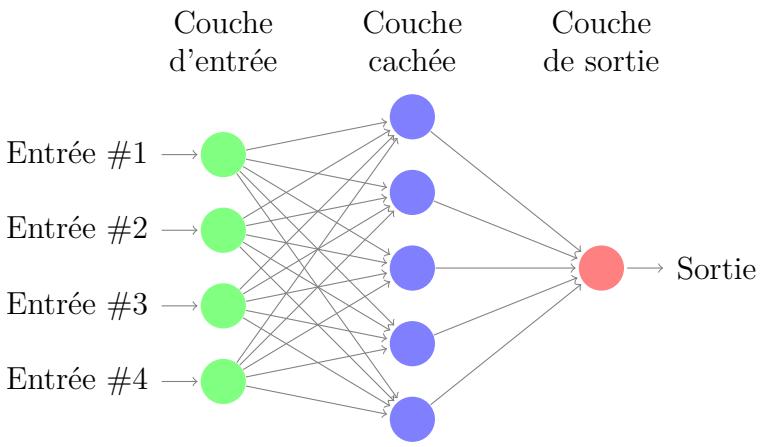


FIGURE 5.2 – Schéma de principe de l'architecture d'un réseau de neurones Chaque cercle correspond à un neurone qui prend un certain nombre d'entrée pour fournir une sortie. Les flèches indiquent les connexions entre les neurones.

réseaux de neurones sont, comme les fonctions de régression, inférées sur la base d'algorithmes de maximisation de la vraisemblance [GBC16]. Ils sont en théorie supposés pouvoir approximer n'importe qu'elle fonction et constituent donc une alternative très intéressante aux modèles linéaires. Enfin, ils sont également connus pour fonctionner sur le principe de «boite noire» difficile à interpréter et donc limités dans un but d'explications biologiques. Cependant, tout dernièrement quelques travaux ont proposé des pistes intéressantes pour pallier ce problème (DeepBlueR [ALBL17], DeepLift [JHP+18], et BPnet [AWS+19]). Pour plus de détails sur les réseaux de neurones et leur utilisation en bioinformatique, nous invitons le lecteur à consulter la revue suivante [MLY17].

Dans cette section, nous allons présenter quatre méthodes récentes pour prédire l'expression des gènes en utilisant un réseau de neurones.

5.2.1 ExPecto

Les auteurs de Zhou *et al.* [ZTY+18] on développé un modèle à «deux étages», appelé ExPecto, pour prédire l'expression des gènes en utilisant la séquence ADN. Le premier étage correspond à une extension de leur précédent modèle, DeepSEA [ZT15] pour prédire les variants épigénétiques à partir de la séquence ADN en utilisant un réseau de neurones convolutif. Le second étage est une régression linéaire avec une pénalisation Ridge [HK70] pour prédire le niveau d'expression à partir des prédictions du premier étage.

Leur modèle a été testé sur les données de 218 types cellulaires humains prove-

nant de GTEx (`https://gtexportal.org`), Roadmap epigenomics [RKM+15] et ENCODE [DHS+18]. Ils ont choisi d'utiliser 990 gènes provenant du chromosome 8 et des régions de -20kb à +20k centrées sur les TSS définis par FANTOM5. Les auteurs ont montré une corrélation de 80% entre les prédictions de ExPecto et les données observées.

5.2.2 Xpresso

Agarwal & Shendure [AS18] ont proposé un modèle pour prédire l'expression des gènes en se basant uniquement sur la séquence ADN et les caractéristiques liées à la demi-vie de l'ARNm (stabilité et dégradation) à l'aide d'un réseau de neurones convolutif. Pour cela, ils ont sélectionné des séquences de -7000 à +3500bp autour des TSS des gènes codants et fournissent également au modèle le taux de %CG et les longueurs des régions fonctionnelles (5' UTR, ORF, introns et 3'UTR).

Leur modèle a été testé sur les données d'expression de 56 types cellulaires humains provenant de Roadmap epigenomics [RKM+15]. Leurs résultats montrent une corrélation de 76% entre les prédictions de leur modèle et les données observées.

5.2.3 Basenji

Les auteurs de Kelley *et al.* [KRB+18] ont développé un réseau de neurones convolutif pour prédire les profils épigénétiques et les profils de transcriptions en utilisant uniquement de grandes séquences ADN. Leur méthode, nommée Basenji, est une version modifiée de leur modèle précédent Basset [KSR16] qu'ils ont présenté pour prédire les régions accessibles de la chromatine (problème de classification).

Leur modèle a été entraîné et évalué sur 973 expériences de CAGE chez l'Homme provenant de FANTOM5 [LHS+15]. La corrélation médiane entre les prédictions et les données observées est de 86%. Toutefois, il est à noter que leur étude n'est pas limitée aux promoteurs. En effet, ils ont extrait des séquences de 131kbp sur les différents chromosomes. Plusieurs autres aspects biologiques sont également abordés dans cet article, en particulier l'influence des régions distantes sur l'expression des gènes.

5.3 Prédiction de l'expression chez *Plasmodium falciparum*

À l'heure actuelle, à notre connaissance, il n'existe qu'une seule publication traitant de la prédiction de l'expression des gènes chez *Plasmodium falciparum*.

Cet article a été publié le 11 septembre 2019 dans PlosCB [RCL+19]. Les auteurs de cette étude, Read *et al.*, ont rassemblé une riche collection de données. Cette collection comprend des caractéristiques des promoteurs comme le taux de %GC et les scores de 25 PWM des TF de la famille AP2 (voir Section 2.3.4.4). Dans cette collection, on trouve également des expériences de ChIP-seq des modifications d'histones, des données de MNase-seq sur le positionnement des nucléosomes, des données de Hi-C qui permettent d'estimer la distance spatiale d'un gène aux télomères, centromère. Contrairement aux méthodes présentées dans la section précédente, les prédictions ne sont donc pas réalisées exclusivement à partir de données de séquences mais incluent également de nombreuses données épigénétiques. La problématique dans cet article est un problème de classification. Leur objectif est de prédire si un gène fait parti des gènes les plus exprimés ou bien s'il fait parti des gènes les moins exprimés. Pour cela, ils ont trié les gènes par leur expression et ils les ont ensuite séparé en 3 groupes égaux dans chaque condition. Le groupe des gènes «moyennement» exprimés est ensuite retiré de l'analyse. En d'autres mots, l'objectif est de prédire si les gènes sont «actifs» ou «inactifs».

Les données utilisées en entrée proviennent de la collection de données expérimentales citée ci-dessus. Chaque mesure est décomposée en deux variables correspondant à 2 régions : les mesures observées dans le promoteur (1 000bp avant le codon *start*), et les mesures observées dans la région codante (500bp après le codon *start*). Les auteurs ont choisi d'utiliser le codon *start* comme point de référence plus que le TSS habituellement utilisé car ils ont observé une meilleure précision dans les prédictions avec cette manière de faire. Il y a en tout 73 variables en entrée (50 scores de PWM, 14 modifications d'histones, 7 variables sur la structure locale et globale de la chromatine, et 2 variables mesurant le %GC dans chaque région).

Pour résoudre le problème de classification, les auteurs ont utilisé trois approches dont ils comparent les prédictions. Le premier modèle est une régression logistique avec une pénalisation *elastic net* en utilisant l'implémentation python fournie par `sklearn`. Le deuxième modèle est un arbre de classification appris avec l'algorithme *gradient boosting* en utilisant l'implémentation `XGboost` de python. Le troisième modèle est un réseau de neurones multicouche avec deux couches cachées, chacune contenant autant de nœuds que la couche d'entrée. Ce réseau est construit en utilisant l'implémentation `DeepPINK` [LFLN18] conçue pour obtenir une sélection de variables robuste avec un contrôle du taux d'erreur.

La performance de chaque modèle est évaluée à l'aide de l'aire sous la courbe ROC (*Receiver Operating Characteristic*). La courbe ROC représente le taux de vrais positifs (la sensibilité, en ordonnée) en fonction du taux de faux positifs (la spécificité, en abscisse). L'aire sous cette courbe quantifie donc la capacité du modèle à classer les données en fonction de ses prédictions. Une valeur d'AUC (aire sous la courbe ROC) proche de 1 correspond au modèle parfait tandis qu'une

valeur proche de 0.5 correspond aux performances d'un modèle aléatoire ou celle d'un modèle prédisant toujours la même classe.

Ils ont ensuite testé leurs différents modèles sur différentes étapes du cycle de vie de *Plasmodium falciparum*. Ils ont observé des résultats similaires entre les prédictions des différents modèle. L'AUC est comprise globalement entre 0.79 et 0.88. Il est à noter que ces valeurs sont quelque peu inférieures aux valeurs d'AUC observées dans d'autres études similaires chez l'Homme (0.94 [CYY$^+$11]) ou chez la souris (0.95 [DGK$^+$12]). Il y a plusieurs explications à cela mais l'une des principales retenues par les auteurs et que *Plasmodium falciparum* possède relativement peu de gènes comparé à ces espèces et donc l'apprentissage des modèles est d'autant plus difficile du fait du peu d'exemples disponibles. Une autre explication viendrait du fait de l'importance de la régulation post-transcriptionnelle chez *Plasmodium falciparum*, comparée à la régulation transcriptionnelle [CHO04].

Chapitre 6

Découverte de longues séquences régulatrices

Comme on l'a vu au chapitre précédent, les auteurs de Bessière *et al.* [BTP+18] ont montré que la composition nucléotidique de la région promotrice chez l'Homme est porteuse d'information pour la prédiction de l'expression du gène associé. Plus précisément, dans leur étude, les auteurs montrent que de découper la région promotrice en 3 sous-régions (proche du TSS, en amont du TSS et en aval du TSS) et de calculer les compositions nucléotidiques de chacune de ces sous-régions séparément permet d'améliorer les résultats de prédiction. Dans cet article, la segmentation de la région promotrice est réalisée sur la base de connaissances *a priori* des promoteurs humains. Si elle semble donner de bons résultats chez l'Homme, il n'est pas certain que cette segmentation soit optimale pour d'autres espèces, d'autant plus si l'on s'intéresse à une espèce phylogénétiquement éloignée de l'Homme comme *Plasmodium falciparum*. De plus, les auteurs se sont restreint pour l'étude des compositions au calcul des fréquences de dinucléotides dans ces régions, mais il serait évidemment intéressant d'étudier l'impact de compositions nucléotidiques plus complexes (k-mers, avec k$>$ 2).

La problématique qui nous intéresse dans ce chapitre est le développement d'une méthode permettant d'identifier de manière automatique des sous-régions spécifiques, dont la composition en un k-mer particulier soit informative de l'expression des gènes. On nomme ces sous-régions potentiellement longues (plusieurs dizaines ou centaines de bp) des « domaines de régulation ». Pour un k-mer donné, on se trouve donc en présence d'un problème de segmentation, qui vise à trouver la ou les régions dont la fréquence du k-mer est la plus informative de l'expression. Notons cependant qu'il ne s'agit pas d'un problème de segmentation classique tel que présenté au Chapitre 1. En effet, ce qui nous intéresse ici n'est pas d'identifier des régions de composition homogène pour ce k-mer, mais des régions dont la fréquence du k-mer est informative de l'expression. Par analogie au distinguo

utilisé pour la classification supervisée contre non-supervisée, on pourrait dire que le problème de segmentation classique est non-supervisé, alors que le problème qui nous intéresse ici est de réaliser une segmentation supervisée.

Nous présentons dans ce chapitre une nouvelle méthode dédiée à ce problème nommée DExTER. DExTER permet d'identifier des paires (région/k-mer) dans lesquelles la fréquence du k-mer dans la région est corrélée à l'expression des gènes. Pour cela, la méthode repose sur une procédure itérative d'exploration de l'espace des domaines de régulation (région/k-mer) possibles. Les paires identifiées forment un ensemble de variables prédictives qui sont ensuite utilisées pour prédire l'expression des gènes. Nous utilisons un modèle linéaire avec une pénalisation LASSO (voir Section 6.1.3) comme prédicteur afin de sélectionner les variables les plus importantes parmi toutes les variables proposées. Nous présentons ensuite dans la Section 6.2 les expériences réalisées sur deux espèces de *Plasmodium* et sur d'autres espèces eucaryotes. Suivant les espèces, la méthode identifie différentes régions relativement longues (centaine de bp) dont l'enrichissement est corrélé à l'expression des gènes.

6.1 La méthode DExTER

La méthode DExTER utilise un ensemble de séquences (une par gène) alignées sur un point d'intérêt, et un vecteur d'expression des gènes. Dans les expériences suivantes, nous utilisons des séquences de 4 001bp centrées généralement sur le TSS mais il est tout à fait possible de considérer d'autres points d'alignement (début/fin du gène, bornes introns/exons, ...).

Il y a deux grandes étapes dans la procédure. La première étape consiste à identifier les paires des (k-mer, région) dont la fréquence du k-mer dans cette région est corrélée à l'expression des gènes. Pour cela, nous avons développé une procédure d'exploration itérative qui vise à optimiser le critère de corrélation (voir ci-dessous). La seconde étape consiste à sélectionner les meilleurs paires identifiées pour apprendre un prédicteur de l'expression sur la seule base de ces variables. Pour cela, nous utilisons un modèle de régression linéaire :

$$y(g) = a + \sum_i b_i x_{i,g} + e(g), \tag{6.1}$$

où $y(g)$ est l'expression du gène g, $x_{i,g}$ est la variable i calculée sur le gène g, $e(g)$ est l'erreur résiduelle associée à g, a est l'ordonnée à l'origine et b_i est le coefficient de régression associé à la variable i. Étant donné que le nombre de variables identifiées peut-être relativement important, nous utilisons un modèle linéaire avec une pénalisation LASSO pour sélectionner les variables les plus importantes.

6.1.1 Critère d'optimisation

L'objectif de notre méthode est d'identifier un ensemble de domaines de régulation c-à-d de paires (région/k-mer) qui puissent être utilisées comme variables prédictives de l'expression. Nous utilisons une mesure de corrélation comme critère d'optimisation. Le coefficient de corrélation ρ de Pearson est un critère d'optimisation souvent utilisé pour mesurer les performances d'un filtre [BCH08, ZDXF16]. Dans notre cas, il est calculé de la manière suivante :

$$\rho(D_{k,r}, Y) = \frac{\mathrm{Cov}(D_{k,r}, Y)}{\sigma_{D_{k,r}} \sigma_Y}, \tag{6.2}$$

avec Cov la fonction de covariance, $D_{k,r}$ le vecteur de fréquences du k-mer k dans la région r pour tous les gènes, Y le vecteur du niveau d'expression des gènes, et σ la fonction d'écart type. Le coefficient de Pearson permet de détecter des relations linéaires entre deux séries de variables. La valeur de ce coefficient est définie entre -1 et 1 et permet d'identifier des corrélations positives lorsque l'expression augmente avec la fréquence du k-mer, ou négatives si l'expression diminue lorsque la fréquence augmente. Si la fréquence du k-mer varie indépendamment de l'expression du gène, la corrélation obtenue est proche de 0. Pour des relations non linéaires, une solution est d'utiliser le coefficient de corrélation de Spearman qui se calcule comme le coefficient de Pearson mais en utilisant cette fois le rang des valeurs de $D_{k,r}$ et Y [HK11]. Lorsque la relation est linéaire, le coefficient de Spearman donne souvent des résultats similaires au coefficient de Pearson. En revanche, la valeur de corrélation peut être plus élevé si la relation est monotone mais non linéaire.

6.1.2 Procédure d'exploration

Notre objectif est donc d'explorer l'espace des k-mers ainsi que l'espace de toutes les régions possibles afin de maximiser notre critère d'optimisation. Une procédure naïve consisterait à explorer tous les k-mers possibles (de longueur maximale K) et toutes les régions possibles sur une séquence. L'exploration de tous les k-mers différents pour $k \in [2 : K]$ nous donne

$$\sum_{k=2}^{K} 4^k = \frac{4^{K+1} - 16}{3} \tag{6.3}$$

k-mers différents. Si l'on prend $K = 6$, cela nous donne 5 456 k-mers à explorer. L'exploration de toutes les régions possibles sur une séquence de taille L nous donne

$$L + (L-1) + \ldots + 1 = \frac{L(L+1)}{2} \tag{6.4}$$

sous-séquences possibles. Si l'on prend une séquence de taille $L = 4\,000$, cela nous donne $8\,002\,000$ sous-séquences différentes. Pour un génome de N gènes, si on imagine une procédure naïve qui nécessite de reparcourir la séquence entière pour calculer la fréquence du k-mer dans chaque sous-séquence, cela nous donne donc une complexité globale de $\mathcal{O}(NL^3 4^K)$ pour l'exploration naïve. En prenant par exemple le génome de *Plasmodium falciparum*, contenant $5\,231$ gènes, $L = 4\,000$ et $K = 6$, cela nous donne environ 10^{18} opérations nécessaires. Cette opération est évidemment impossible et il est donc nécessaire de recourir à une heuristique pour explorer l'espace des k-mers et des régions possibles.

Pour résoudre ce problème, nous avons développé une heuristique qui permet de réduire à la fois le nombre de sous-séquences considérées et le nombre de k-mers à explorer. Tout d'abord, l'espace des sous-régions possibles est réduit en procédant au découpage en n sous-régions unitaires (de longueur fixe ou variable, voir plus loin). On considère ensuite une structure en treillis où chaque nœud correspond à une sous-région, voir Figure 6.1. En mathématiques, un treillis est un ensemble partiellement ordonné dans lequel chaque paire d'éléments admet une borne supérieure et inférieure. Ici les éléments sont des régions contiguës, et la relation d'ordre considérée est l'inclusion. Formellement, on a donc un demi-treillis, car chaque paire de région A et B est associée à une borne supérieure correspondant à la région minimale incluant A et B. La base du demi-treillis correspond aux n sous-séquences unitaires définies. Plus on monte dans le demi-treillis, plus la région considérée est grande. Le sommet correspond à la séquence entière. De cette manière, avec une séquence de longueur $L = 4000$ découpée en 10 sous-régions unitaires de longueur 400bp, il ne reste que 55 régions contiguës possibles à explorer. Un intérêt de cette structure est qu'elle nous fournit une structure naturelle pour conduire les différents calculs dont nous avons besoin, et qu'elle permet en outre, comme nous le verrons par la suite, de visualiser facilement l'importance de la région pour un k-mer donné (voir l'exemple de la Figure 6.4). Un autre intérêt de cet structure est de faciliter les calculs de fréquences des k-mers. En effet, pour calculer la corrélation entre l'expression et la fréquence d'un k-mer dans une région donnée, il est nécessaire de compter pour chaque séquence, le nombre d'occurrence du k-mer dans la région, voir Figure 6.3. Étant donné un k-mer, si on recalcule sa fréquence dans toutes les régions de manière naïve, on doit parcourir la séquence $\mathcal{O}(n^2)$ fois, avec n le nombre de régions unitaires, ce qui peut être long, même si on utilise un index. Avec le treillis, chaque séquence n'est parcourue qu'une fois pour calculer les fréquences des éléments de la base, et les fréquences des autres nœuds du treillis sont calculées de manière itérative avec des opérations en temps constant (additions/soustractions). En effet, il est possible de calculer la fréquence d'un k-mer dans une région comme la somme des fréquences des deux régions sous-jacentes moins la fréquence dans l'intersection de ces régions

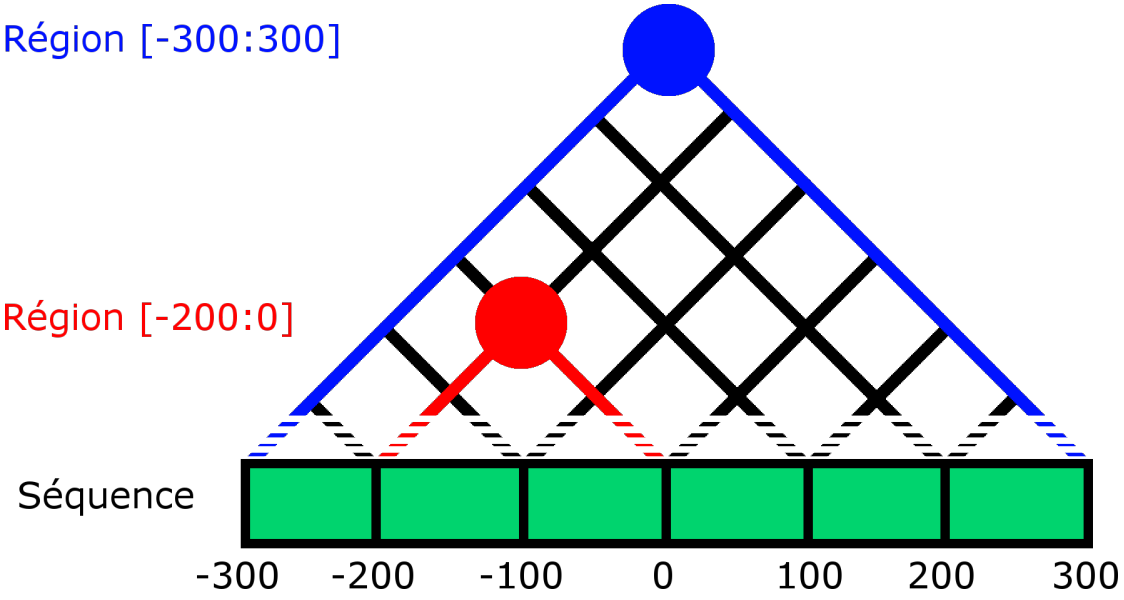


FIGURE 6.1 – Exemple de demi-treillis
La séquence est découpée en sous-régions unitaires et on définie une structure de demi-treillis représentant les différentes régions considérées par DExTER. Chaque nœud du treillis correspond à une sous-région ou l'union de plusieurs sous-régions. Le sommet correspond à la séquence entière.

(voir Figure 6.2).

Le nombre de sous-régions unitaires n est laissé à l'appréciation de l'utilisateur et est à adapter aux séquences étudiées. Un plus grand nombre de sous-régions permet de gagner en précision au détriment du temps de calcul. Nous proposons deux solutions de segmentation avec des régions de tailles uniformes ou des régions de tailles variables. Les régions de tailles uniformes sont de longueur $\frac{L}{n}$ (arrondie au supérieur). Elles sont utilisées lorsque l'on ne dispose d'aucun *a priori* sur les régions d'intérêts possibles dans les séquences. Les régions de tailles variables sont construites progressivement, en utilisant un polynôme générateur. Dans l'implémentation actuelle nous utilisons le polynôme $(x+a)^p$. p est défini par l'utilisateur (par défaut 3) et le a est déterminé automatiquement (en commençant à 0 puis par pas de 1) afin de s'approcher au mieux du nombre de sous-régions n demandé par l'utilisateur. De cette manière, les régions proches du point d'alignement sont courtes, tandis que les régions éloignées sont plus longues. Les régions de tailles variables permettent d'augmenter la précision autour du point d'alignement (proche du TSS par exemple) et de diminuer progressivement la précision lorsqu'on s'en éloigne pour limiter le nombre de régions total.

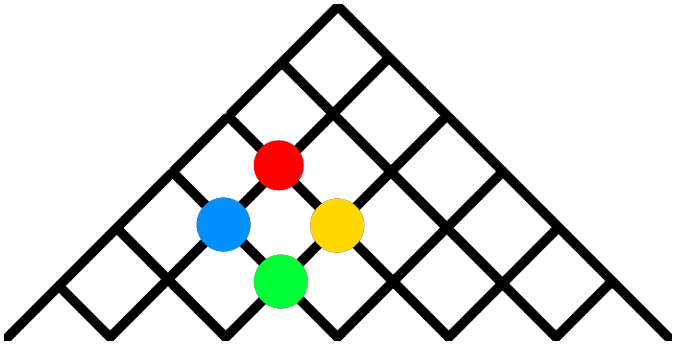


FIGURE 6.2 – Remplissage du treillis de manière itérative

Une fois les fréquences calculés dans les sous-régions unitaires (nœuds à la base du treillis) les fréquences associées aux autres nœuds sont calculés de manière itérative. Pour cela il faut utiliser le nombre d'occurrences du k-mer dans le nœud (fréquence multipliée par la taille de la sous-région). On obtient alors que le nombre d'occurrences du nœud rouge est égal à la sommes des nombres d'occurrences des nœuds bleu et jaune moins le nombre d'occurrences du nœud vert (qui est l'intersection des régions bleu et jaune).

Une fois les séquences découpées en sous-régions unitaires, la procédure d'exploration consiste alors en l'alternance de deux phases : une phase de segmentation, et une phase d'expansion. En partant de l'ensemble des dinucléotides, on cherche les régions d'intérêt de chaque k-mer (segmentation). Ensuite, sur chaque région d'intérêt, on étend le k-mer considéré dans les 8 (k+1)-mers possibles et on évalue leurs performances (expansion). Les (k+1)-mers qui montrent de meilleures performances que les k-mers dont ils sont issus sont sélectionnés pour continuer l'exploration, et les deux phases (segmentation, expansion) sont réitérées sur ces (k+1)-mers. Cela permet de concentrer l'exploration sur les k-mers qui semblent porter de l'information et d'interrompre l'exploration des k-mers peu informatifs.

Le principe de cette heuristique se résume alors en l'alternance de deux phase :

— la segmentation : on recherche une région r' qui améliore la corrélation par rapport à la région initiale r : $\rho(D_{k,r'}, Y) > \rho(D_{k,r}, Y)$,

— l'expansion : on recherche un (k+1)-mer k' qui améliore la corrélation de k sur la même région : $\rho(D_{k',r}, Y) > \rho(D_{k,r}, Y)$.

Tous les domaines qui ont permis d'améliorer la corrélation à un moment dans l'exploration sont retenus et ajoutés dans un ensemble qui sera renvoyé lorsqu'il ne restera plus de domaine à explorer. Un domaine A est désigné comme «parent» d'un domaine B s'il a permis de donner «naissance» à B lors d'une phase de segmentation ou d'expansion. Lors de l'exploration, on vérifie qu'on observe à chaque étape un gain de corrélation avec le domaine parent mais également avec tous les

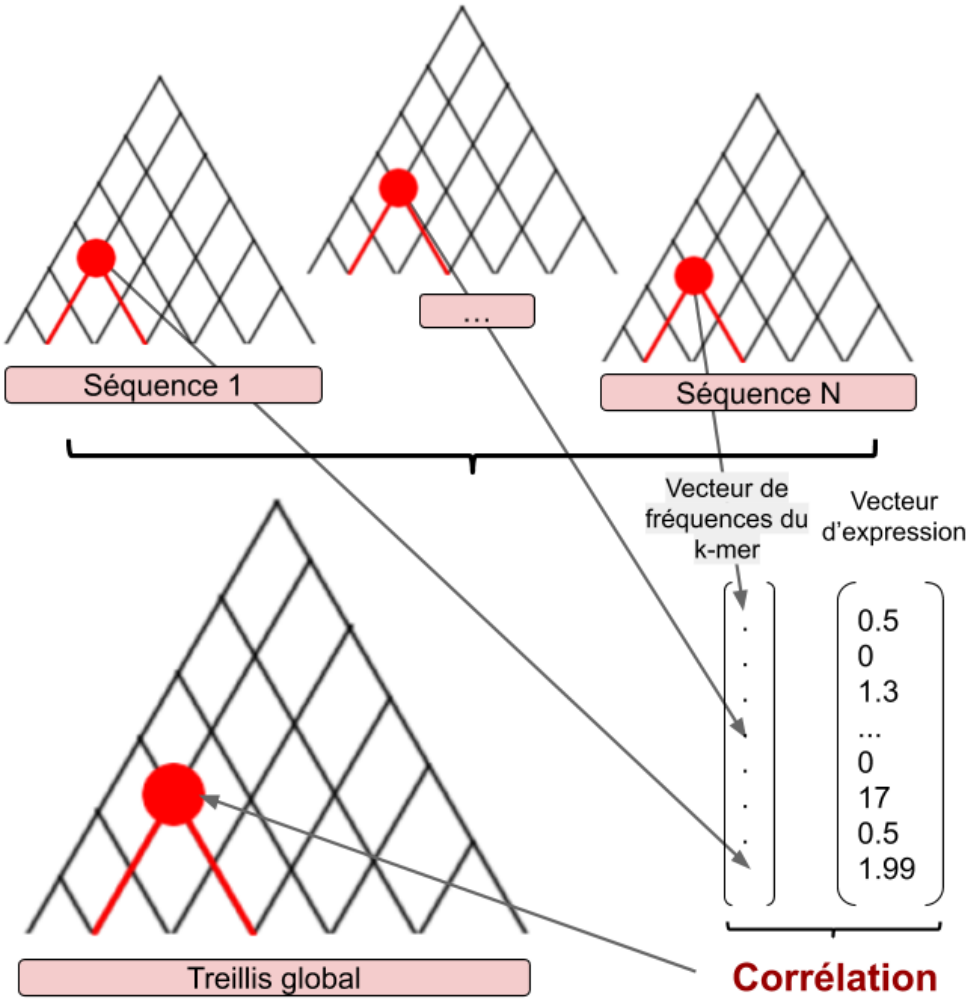


FIGURE 6.3 – Construction du treillis de corrélation
DExTER commence par construire un treillis par séquence et les rempli avec la fréquence d'un k-mer donné dans chaque région considérée. Ensuite, il construit un nouveau treillis contenant les valeurs de la corrélation entre la fréquence du k-mer et le vecteur d'expression dans chaque région. Voir un exemple Figure 6.4.

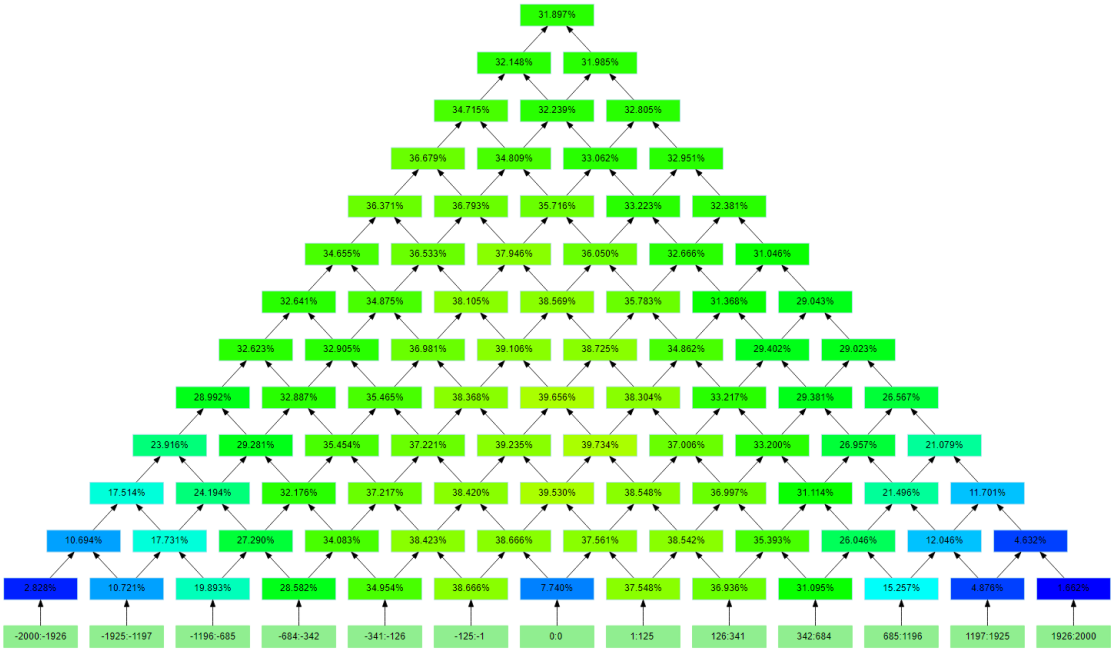


FIGURE 6.4 – Treillis de corrélation

Treillis de corrélation pour le dinucléotide CG obtenu lors de l'exploration de DExTER sur des séquences de ± 2kbp autour du TSS. Plus la corrélation est forte, plus les couleurs tendent vers des couleurs chaudes. La dernière ligne en bas correspond aux sous-séquences définies lors de l'exploration.

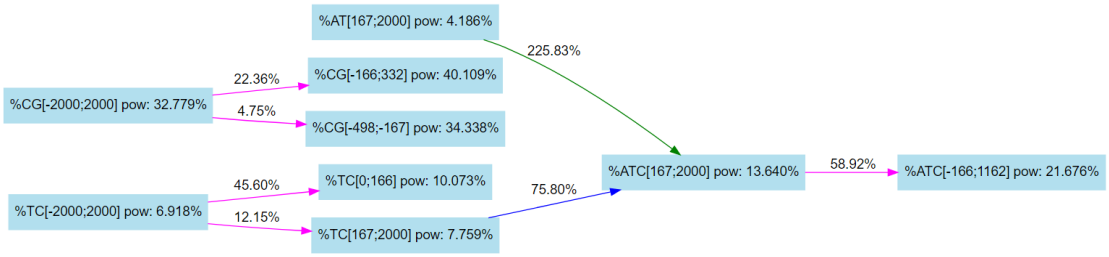


FIGURE 6.5 – Extrait du graph d'exploration
Extrait du graphe d'exploration de DExTER montrant la liste des domaines retenus. Les flèches roses indiquent les étapes de segmentation, les bleus les étapes d'expansion, les vertes indiquent les contrôles effectués pour s'assurer qu'on ne capture pas le signal d'un autre k-mer lors de l'exploration.

domaines k-1 qui auraient pu donner naissance au domaine considéré. Par exemple, si l'on observe un gain de corrélation entre le domaine $D_{CGC,r}$ et le domaine $D_{CG,r}$, on vérifiera également que l'on observe un gain par rapport au domaine $D_{GC,r}$ afin de s'assurer que l'on ne capture pas le signal provenant d'un autre k-mer en passant au (k+1)-mer. Une fois la procédure terminée, il est possible de dessiner un graphe d'exploration des domaines (voir Figure 6.5). L'Algorithme 4 décrit le pseudo-code complet de la procédure. En terme de temps de calcul, la complexité dans le pire des cas de DExTER est $\mathcal{O}(Nn^2 4^K)$, avec N le nombre de séquences, n le nombre de sous-régions unitaires (base du treillis), et K la longueur maximale des k-mers considérés. Il est important de noter que la stratégie d'exploration permet en pratique de n'explorer qu'une très faible proportion des 4^K k-mers possibles. De plus, l'utilisateur a également la possibilité de définir un seuil d'augmentation minimale de la corrélation afin de concentrer l'exploration sur les branches où le gain est le plus important. Nous discuterons des différents paramètres et leur influence sur les résultats et le temps de calcul dans la Section 6.2.1.

Il est intéressant de noter que la procédure d'exploration de l'Algorithme 4 peut s'apparenter à un parcours de graphe en largeur. Cette stratégie nous permet en pratique de paralléliser le traitement de tous les k-mers pour un k donné et de vérifier à chaque itération si différentes branches de l'exploration se rejoignent afin de ne pas traiter inutilement plusieurs fois le même domaine. Cela permet également de ne garder en mémoire que les treillis de corrélation des k et (k-1)-mers puisqu'il n'y a pas de retour en arrière. Il est possible d'envisager la même exploration en parcourant le graphe en longueur mais cela nécessiterait des mécanismes de synchronisation plus important pour limiter les traitements redondants et il serait nécessaire de conserver tous les treillis de corrélation en mémoire étant donné que les processus traiteraient des k-mers de tailles différentes.

Algorithme 4 Procédure d'exploration de DExTER

Entrée: Y : vecteur d'expression des gènes
Sortie: $\mathcal{R}$: liste des domaines retenus

$L \leftarrow$ initialisation de la liste des domaines à explorer (dinucléotides sur toute la séquence)
$\mathcal{R} \leftarrow \emptyset$
tant que $L \neq \emptyset$ **faire**
  $D_{k,r} \leftarrow$ un domaine de L
  si $\rho(D_{k,r}, Y) > \rho(\text{Parent}(D_{k,r}), Y)$ **et** $\rho(D_{k,r}, Y) > \rho(D_{k-1,r}, Y)$ **alors**
    $\mathcal{R} \leftarrow \mathcal{R} + D_{k,r}$
    si Le k-mer de $D_{k,r}$ est égal au k-mer de Parent($D_{k,r}$) **alors**
      {*Le domaine $D_{k,r}$ est issue d'une phase de segmentation*}
      $E \leftarrow 8$ nouveaux domaines $D_{k+1,r}$ enfants de $D_{k,r}$
      $L \leftarrow L + E$
    sinon
      {*Le domaine $D_{k,r}$ est issue d'une phase d'expansion*}
      $R \leftarrow$ liste des régions triées par ordre décroissant de corrélation pour le k-mer k
      E $\leftarrow \emptyset$
      pour $r' \in R$ **faire**
        si r' n'intersecte pas une région de E et $\rho(D_{k,r'}, Y) \geq \rho(D_{k,r}, Y)$ **alors**
          E $\leftarrow$ E + $D_{k,r'}$
        fin si
      fin pour
      $L \leftarrow L+$ E
    fin si
  fin si
fin tant que

	Y	%GT [-1925 : 126]	%GG [-1925 : -1197]	%GC [-1925 : 2000]	...
gene1	0.15	0.17	0.24	0.15	...
gene2	0.09	0.20	0.19	0.09	...
gene3	0.17	0.16	0.18	0.12	...
...	...	...	...	...	...

TABLE 6.1 – Matrice des fréquences observées dans chaque séquence pour chaque domaine identifié par DExTER
Une fois l'exploration terminée, DExTER génère une matrice comprenant pour chaque gène, l'expression mesurée ainsi que pour chaque domaine $D_{k,r}$ identifié, la fréquence du k-mer k dans la région r considérée. Cette matrice sera ensuite utilisée pour l'apprentissage du modèle linéaire.

Une étape clé en terme de temps de calcul est celle de l'identification de la position des occurrences des différents k-mers dans chaque séquence. Nous utilisons pour cela le logiciel MOTIF [MMR18]. Ce logiciel exploite une implémentation de la transformée de Burrows-Wheeler (BWT) [BW94] et du FM-Index [FM00] pour explorer efficacement des séquences biologiques.

6.1.3 Apprentissage et évaluation du prédicteur

Une fois l'exploration de DExTER terminée, celui-ci renvoie la liste des domaines retenus. Il est alors possible de construire une matrice de variables prédictives $N \times p$ contenant les valeurs de chaque domaine pour chaque gène, avec N le nombre de gènes et p le nombre de domaines identifiés (voir Table 6.1). Cette matrice est ensuite utilisée pour entraîner un modèle linéaire avec une pénalisation LASSO. Le LASSO permet de sélectionner un sous-ensemble des variables importantes en affectant un coefficient nul aux variables peu informatives ou redondantes. Comme on l'a vu Section 1.6, la fonction objective du LASSO est la suivante :

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} \frac{1}{2N}\sum_{i=1}^{N}(y_i - \beta_0 - x_i^T\beta)^2 + \lambda||\beta||_1, \tag{6.5}$$

avec $x_i \in \mathbb{R}^p$ les valeurs des p variables prédictives pour le gène i, $y_i \in \mathbb{R}$ l'expression du gène i, $i \in [1 : N]$, et λ un paramètre de complexité. Le choix du paramètre λ permet de contrôler l'importance de la pénalisation. L'algorithme réalise l'optimisation pour différentes valeurs de λ possibles. Par la suite nous retiendrons le modèle obtenu avec le λ qui minimise la somme des erreurs quadratiques, souvent appelé le λ_{min} (voir Figure 6.6).

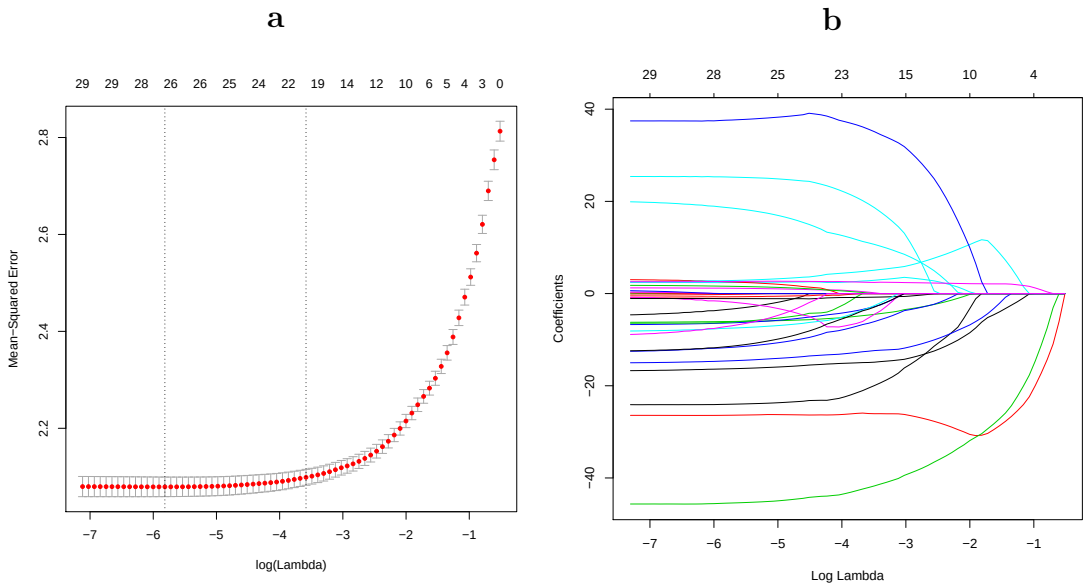


FIGURE 6.6 – Choix du λ et coefficients du LASSO
a. Erreur quadratique en fonction des différents λ testés par LASSO. Les lignes verticales en pointillés montrent à gauche λ_{min} : la pénalisation qui minimise l'erreur ; à droite λ_{1se} : la plus forte pénalisation où l'erreur se situe à un écart type de l'erreur avec λ_{min}. **b.** Valeurs des β du modèle en fonction des différents λ testés par LASSO.

Lorsque l'on dispose de plusieurs vecteurs d'expression Y d'une même série, il existe une variante multi-tâche du LASSO. Dans cette variante, on construit un ensemble de prédicteurs (un pour chaque vecteur d'expression Y) en faisant en sorte que les mêmes variables prédictives soient sélectionnées pour les différents prédicteurs. Plus précisément, les β à zéro sont les mêmes pour les différents prédicteurs, tandis que les β non nuls sont estimés indépendamment pour chaque Y. Le problème d'optimisation est le suivant :

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{(p+1)\times K}} \frac{1}{2N}\sum_{i=1}^{N}||y_i-\beta_0-\beta^T x_i||_F^2+\lambda\sum_{j=1}^{p}||\beta_j||_2, \tag{6.6}$$

avec β_j la j-ème ligne de la matrice de coefficient β. En pratique nous utilisons les fonctions d'apprentissage de la librairie R glmnet, avec l'option `gaussian` pour la régression avec pénalisation LASSO classique, et l'option `mgaussian` pour la variante multi-tâche.

Une fois les domaines identifiés et le modèle entrainé, les performances du modèle sont évaluées en mesurant la corrélation entre les prédictions du modèle et les données observées. Une corrélation élevée nous indiquera que le modèle est vraisemblablement capable de prédire l'expression des gènes sur la base des domaines identifiés par notre procédure. Notons que cette évaluation se fait sur un ensemble de séquences qui n'ont pas été utilisées par DExTER ni pour l'extraction de domaines, ni pour l'apprentissage du modèle. Généralement, nous appliquons DExTER sur deux tiers des séquences et nous évaluons le modèle sur le tiers restant. La sélection des séquences qui participent à l'apprentissage ou au test est aléatoire.

6.1.4 Quantification de l'importance des variables

Une fois que nous avons identifié un modèle capable de prédire l'expression des gènes, la question suivante est de savoir, parmi toutes les variables sélectionnées par le LASSO, quelles sont les variables les plus importantes. En effet, bien que la pénalisation LASSO permette de réduire le nombre de variables de manière effective, ce nombre peut rester relativement important lorsqu'on s'intéresse aux variables utilisées dans le modèle correspondant au $\lambda_{\min}$ (une vingtaine par exemple). De plus, au delà de la distinction sélectionné/non-sélectionné il est difficile de quantifier l'importance de chaque variable pour le modèle. Différentes approches ont été proposées dans la littérature pour faire cela. Suivant le type de prédicteur utilisé, il existe des méthodes propres à chacun qui permettent d'évaluer l'importance des variables. C'est par exemple le cas de la profondeur moyenne d'une variable dans une forêt d'arbres aléatoires [SGS]. Les auteurs de [GE03] fournissent une revue des méthodes standards d'évaluation de l'importance des variables pour différents

prédicteurs. Parmi les méthodes les plus simples, on peut citer par exemple l'utilisation de la corrélation de chaque variable au vecteur Y, ou encore l'information mutuelle des paires de variables comme critère de tri. Cependant, comme illustré par [ZL09], l'inconvénient majeur de ces méthodes simples est qu'elles peuvent sous-estimer l'importance de certaines variables qui ne semblent individuellement pas importantes mais le deviennent en coopération avec d'autres variables. Lorsque l'on cherche à évaluer l'importance de chaque variable en prenant en compte l'aspect collaboratif, il est intéressant de remarquer que ce problème peut être résolu à l'aide des Valeurs de Shapley, issus de la théorie des jeux, et qui définissent une répartition des gains méritocratique dans un jeu coopératif [Sha53, AK14]. Les valeurs de Shapley présentent l'avantage de pouvoir s'appliquer à n'importe quel modèle mais elles présentent l'inconvénient majeur de nécessiter un nombre exponentiel d'opérations. En effet, pour calculer ces valeurs il faudra estimer les performances de tous les arrangements de variables possibles. Pour limiter cela, les auteurs de [AK14] proposent d'estimer ces valeurs à l'aide des méthodes de Monte Carlo.

Nous avons choisi de quantifier l'importance de chaque variable en nous basant sur le paramètre λ qui contrôle la pénalisation LASSO. Comme on l'a vu, chaque valeur de λ est associée à un modèle (voir Figure 6.6). On s'intéresse aux modèles associés aux λ compris entre $\lambda_{\min}$ et $\lambda_{\max}$, le modèle associé à $\lambda_{\max}$ étant le modèle le plus contraint, c'est à dire celui avec le moins de paramètres. Notre approche de quantification de l'importance des variables consiste à prendre ces modèles dans l'ordre, en commençant par $\lambda_{\max}$. À chaque modèle m est associé un ensemble de variables qui sont sélectionnées dans le modèle m mais ne l'étaient pas dans les modèles précédents (les modèles les plus contraints). Pour chacune de ces variables X, on calcule la performance du modèle m et la performance du modèle m privé de X. La différence de performance des deux modèles est utilisée pour estimer l'importance de la variable X (voir le pseudo-code de l'Algorithme 5). Notons qu'une procédure alternative consisterait à utiliser le modèle $\lambda_{\min}$ et à calculer pour chaque variable X la perte de performance de ce modèle et du même modèle privé de X. Cependant, en pratique, avec cette solution les variables les plus importantes au sens du LASSO (celles qui sont sélectionnées lorsque la pénalisation est la plus importante) ne sont pas toujours celles dont l'importance estimée est la plus grande dans le modèle $\lambda_{\min}$, ce qui ne nous semble pas satisfaisant. Avec notre solution, l'importance d'une variable est calculée via les performances du plus petit modèle ayant sélectionné cette variable, et l'importance estimée est en pratique plus en adéquation avec l'ordre d'apparition dans le LASSO. Il s'agit cependant d'une solution purement *ad-hoc* qui mériterait d'être plus amplement testée et évaluée. Notons cependant que cette procédure présente l'avantage d'exploiter les différents modèles retournés par glmnet et a donc également l'avantage d'être peu coûteuse

en temps de calcul.

Algorithme 5 Calcul du score d'importance des variables dans le LASSO

Entrée: Λ : différentes valeurs de λ testées lors de l'apprentissage (par ordre décroissant), $\mathcal{B}$: matrices des valeurs de β pour chaque *lambda*, $\mathcal{D}$: liste des domaines sélectionnés par LASSO avec λ_{min}

Sortie: $\mathcal{S}$: vecteur de scores pour chaque domaine

pour tout $D_{(k,r)} \in \mathcal{D}$ **faire**
$\lambda' =$ premier λ de Λ où $\beta(D_{(k,r)}) \neq 0$ et au moins deux $\beta \neq 0$
$\rho_1 =$ corrélation entre l'expression observée et la prédiction du modèle $\mathcal{B}_{\lambda'}$
$\rho_2 =$ corrélation entre l'expression observée et la prédiction du modèle $\mathcal{B}_{\lambda'}$ et $\beta(D_{(k,r)}) = 0$
$\mathcal{S}(D_{(k,r)}) = \rho_1 - \rho_2$
fin pour

6.2 Expériences

Nous avons appliqué notre approche sur différents jeux de données ciblant des espèces eucaryotes unicellulaires et pluricellulaires dans différentes conditions. Pour les organismes unicellulaires, nous avons analysé le cycle érythrocytaire de *Plasmodium falciparum* [OWA+10], le cycle de vie de *Plasmodium berghei* [OBJ+14], le cycle de vie de *Toxoplasma gondii* [RMW+19], et le cycle cellulaire de *Saccharomyces cerevisiae* [HSH+18]. Pour les organismes pluricellulaires, nous avons analysé différents types cellulaires et le développement de *Drosophila melanogaster* [YJP+18, LAC+16], *Caenorhabditis elegans* [CPR+17, ZY17], l'Homme [GTE13, WXL+18], et *Arabidopsis thaliana* [LJX+12, SAE+16]. De manière plus détaillée, les données sont les suivantes :

- *P. falciparum* : 7 points dans le temps de 0 à 48h ;
- *P. berghei* : 2 conditions par étape du cycle de vie : gametocyte, ookinète, ring, schizont, trophozoïte ;
- *T. gondii* : avant infection puis 3, 5 et 7 jours après l'infection du chat, 12 conditions au total
- *S. cerevisiae* : 5 points dans le temps sur 1h du cycle cellulaire ;
- *D. melanogaster* : 10 conditions du développement embryonnaire et 8 types cellulaires (abdomen, système digestif, système génital, gonades, tête, appareil reproducteur, thorax, corps complet) ;
- *C. elegans* : 7 points dans le temps du développement embryonnaire et 7 types cellulaires (muscles de la paroi du corps, cellule gliale, gonades,

intestins, neurones, pharynx) ;
— Homme : 8 points dans les premiers temps du développement embryonnaire et 7 types cellulaires (fibroblaste, œsophage, poumon, glande salivaire, pancréas, glande pituitaire, sang) ;
— *A. thaliana* : 4 types cellulaires (fleur, fruit, feuille, racine) et 7 points dans le temps du développement embryonnaire.

Chaque série de données est composée de plusieurs conditions (type cellulaire, stade de développement ou point dans le temps). On apprend un modèle par condition, et on a donc plusieurs modèles pour une même série de données.

6.2.1 Analyses préliminaires : Influence des paramètres et des choix de la méthode

Avant de réaliser les expériences, nous devons commencer par déterminer différents paramètres liés à la segmentation (combien de régions ? taille uniforme ou variable ?) et à l'exploration de l'espace des k-mers (différence de corrélation minimum pour continuer l'exploration). Afin de déterminer ces paramètres, des études préalables ont été réalisées sur certains jeux de données présentés plus haut (Homme, *P. falciparum*, *D. melanogaster* et *S. cerevisiae*) et nous avons observé les effets produits sur la précision des prédictions et le temps de calcul.

La Figure 6.7 présente les résultats obtenus lorsqu'on fait varier le nombre de régions, avec des régions uniformes et un seuil de gain de corrélation de 5%. On observe ainsi qu'un plus grand nombre de régions tend à augmenter la précision des prédictions. Cependant, cela implique également des temps de calcul supplémentaires. La Figure 6.8 présente les résultats obtenus lorsqu'on compare les prédictions obtenues avec des régions de tailles uniformes et des régions de tailles variables pour un nombre de régions égal à 13. On observe que les résultats sont généralement meilleurs avec des régions de tailles variables. La Figure 6.9 présente les résultats obtenus avec 13 régions de tailles variables et différents seuils de gain de corrélation. On observe qu'un seuil trop élevé nuit à la précision des prédictions mais un seuil nul engendre des temps de calcul beaucoup plus long.

Compte tenu de ces résultats, nous avons choisi d'utiliser 13 régions de tailles variables et un seuil de 3% de gain de corrélation pour les expériences à venir. Ce choix nous permet de concilier les bonnes performances du prédicteur et un temps de calcul acceptable.

Nous avons également voulu estimer l'importance des différents choix qui nous ont amené à développer la méthode DExTER. Plus précisément, on a cherché à comparer les résultats obtenus lorsqu'on utilise uniquement la fréquence des dinucléotides calculés sur toute la séquence (sans segmentation donc), la fréquence des dinucléotides sur la meilleure région identifiée ou bien si on utilise tous les

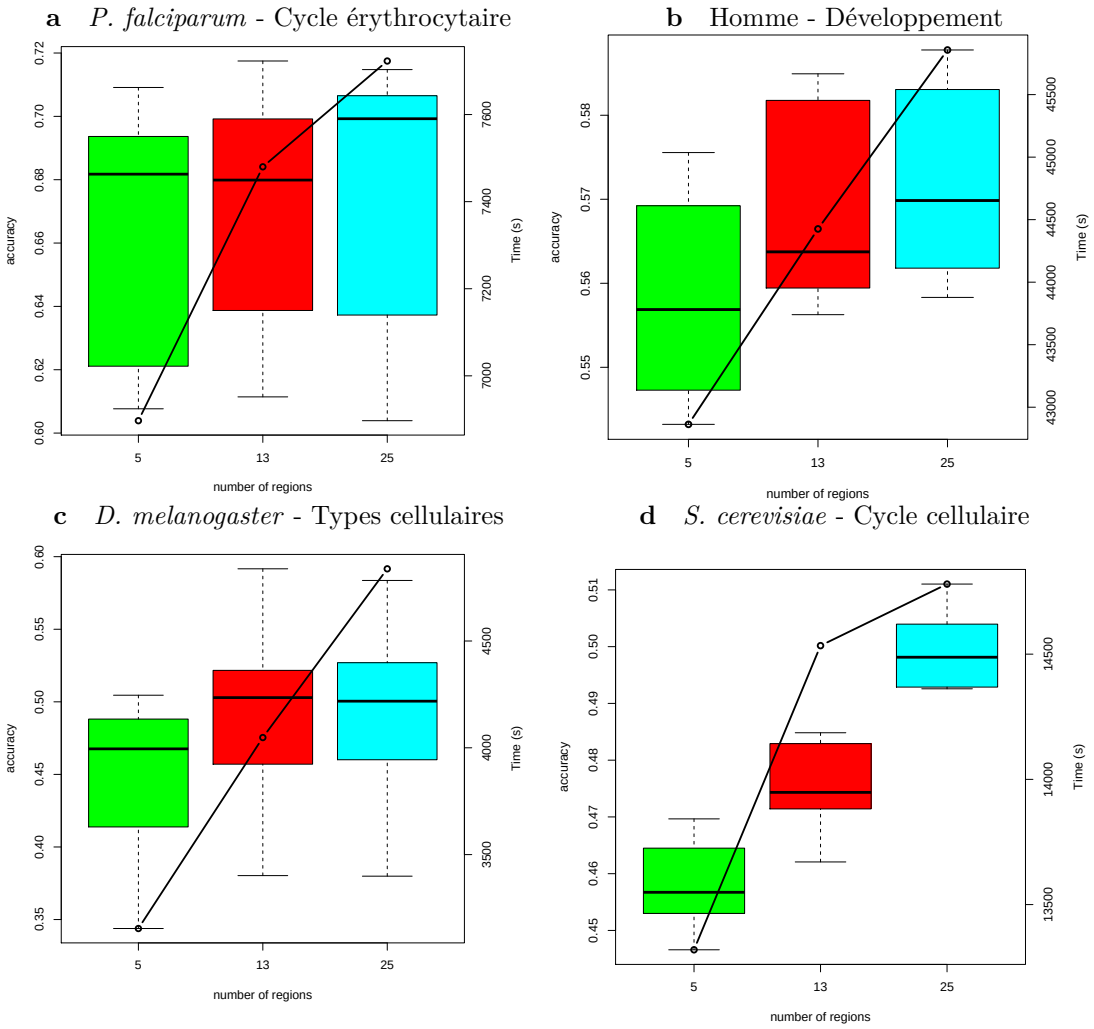


FIGURE 6.7 – Effet du nombre de régions
Précision de la prédiction obtenue avec 5 régions (en vert), 13 régions (en rouge) et 25 régions (en cyan), un seuil de gain de corrélation de 5%, et temps de calcul associé (courbe noire).

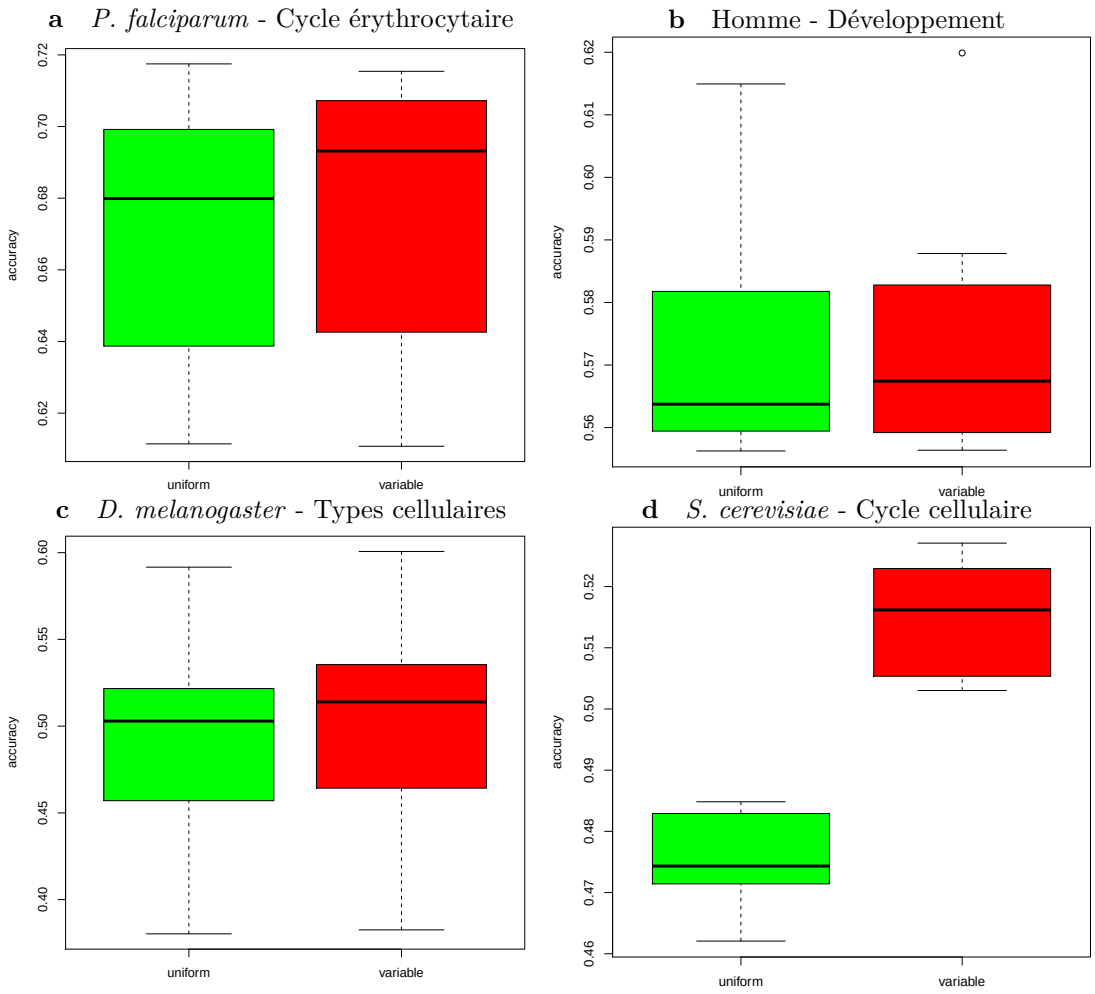


FIGURE 6.8 – Effet des régions uniformes ou variables
Précision de la prédiction obtenue avec des régions de taille uniforme (en vert) et des régions de taille variable (en rouge) pour 13 régions et un seuil de gain de corrélation de 5%.

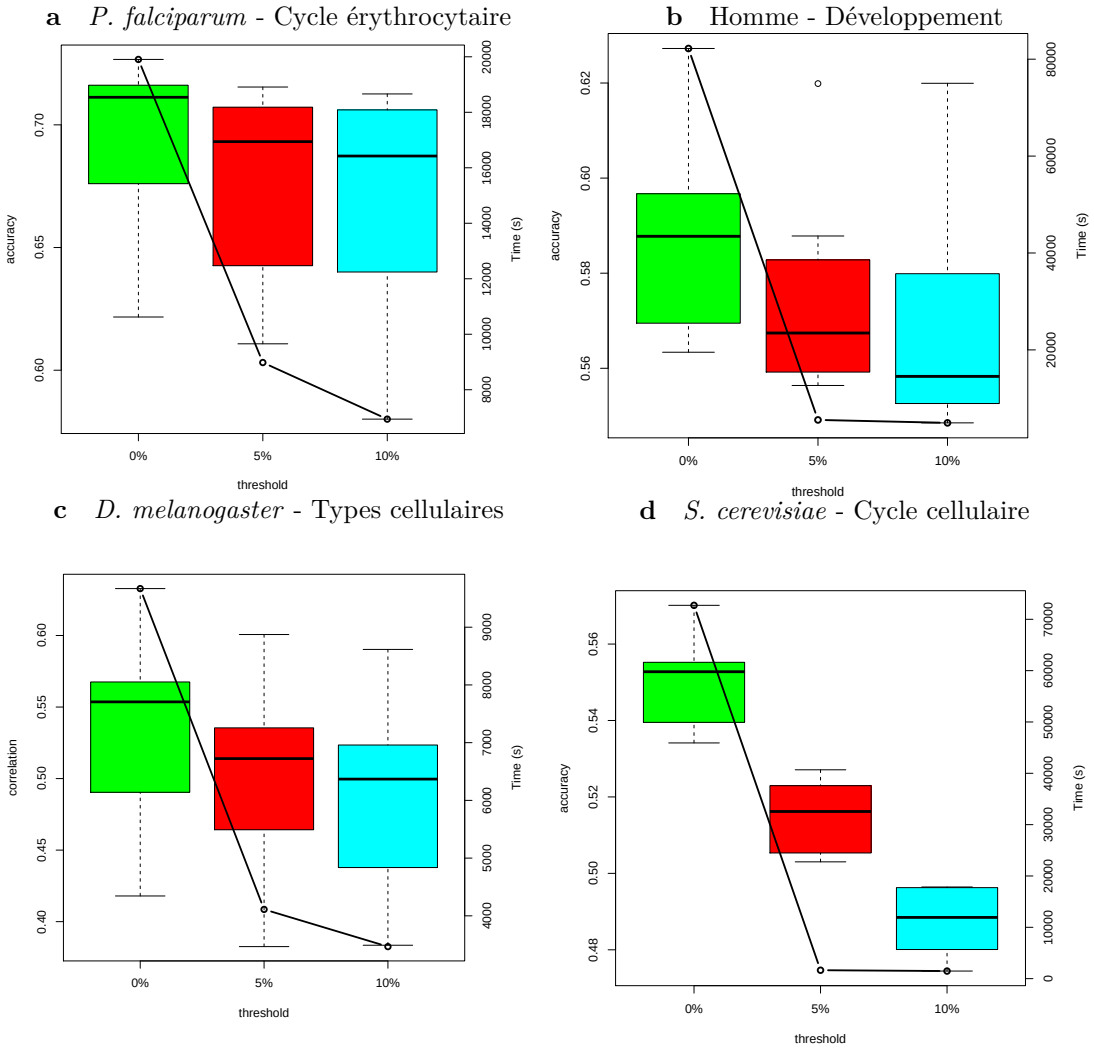


FIGURE 6.9 – Effet du seuil du gain de corrélation
Précision de la prédiction obtenue avec différents seuils de corrélation : 0% en vert, 5% en rouge et 10% en cyan, et 13 régions de taille variable. La courbe noire représente le temps de calcul associé pour les différents seuils.

domaines identifiés par la méthode (k-mers de longueurs variables et segmentation) (voir Figure 6.10). On observe un gain très net apporté par la procédure de segmentation par comparaison aux performances obtenues lorsqu'on utilise toute la région. Pour la procédure d'exploration des k-mers, le gain semble plus modeste pour certaines espèces, mais il peut également être très importants pour certaines autres.

Enfin, nous avons voulu évaluer le temps nécessaire aux différentes opérations de la méthode. Pour cela, nous avons mesuré le temps passé à rechercher les occurrences des k-mers, construire et explorer les treillis de corrélation et toutes les entrées/sorties du programme (voir Figure 6.11). On observe qu'une grande partie du temps est consacrée aux entrées/sorties du programme. Cela s'explique en partie par le fait que l'on a choisi de garder un journal de chaque opération de la méthode. Cela était nécessaire dans la phase de développement pour contrôler *a posteriori* l'exploration, mais cela sera laissé en option dans la version finale du logiciel. L'autre opération coûteuse en temps est la recherche des k-mers. Le logiciel MOTIF que nous utilisons pour rechercher les occurrences d'un k-mer est performant pour rechercher un k-mer particulier. Cependant, nous avons besoin de l'interroger pour chaque k-mer que l'on souhaite explorer et cela nécessite de recharger l'index à chaque opération dans son implémentation actuelle. De plus, il est nécessaire d'utiliser des fichiers temporaires pour récupérer les résultats car on ne dispose pas d'interface simple.

6.2.2 Premiers résultats : de longues séquences avec une composition spécifique permettent de prédire l'expression chez les différentes espèces

La Figure 6.12 présente les résultats de corrélation entre la prédiction des modèles et l'expression observée dans les différentes conditions présentées plus haut. Les résultats de la méthode fluctuent entre 50% et 60% pour la majorité des espèces ce qui semble généraliser les résultats observés chez l'Homme dans Bessière *et al.* [BTP+18], qui montrent qu'il est possible de prédire l'expression des gènes avec une étonnante précision en considérant uniquement les fréquences des dinucléotides dans des régions spécifiques mais relativement grandes. En particulier, on remarque que la précision de la méthode dépasse les 70% de corrélation chez *Plasmodium falciparum* à différents temps de son cycle érythrocytaire, ce qui est très intrigant pour un organisme dans lequel la plupart des tentatives d'identification des facteurs de transcription (TF) ont été infructueuses.

Nous avons ensuite vérifié si les grandes régions identifiées par notre méthode pouvaient correspondre à de multiples occurrences d'un site de fixation d'un TF classique comme présentés dans les bases de données TRANSFAC [WDKK96] ou

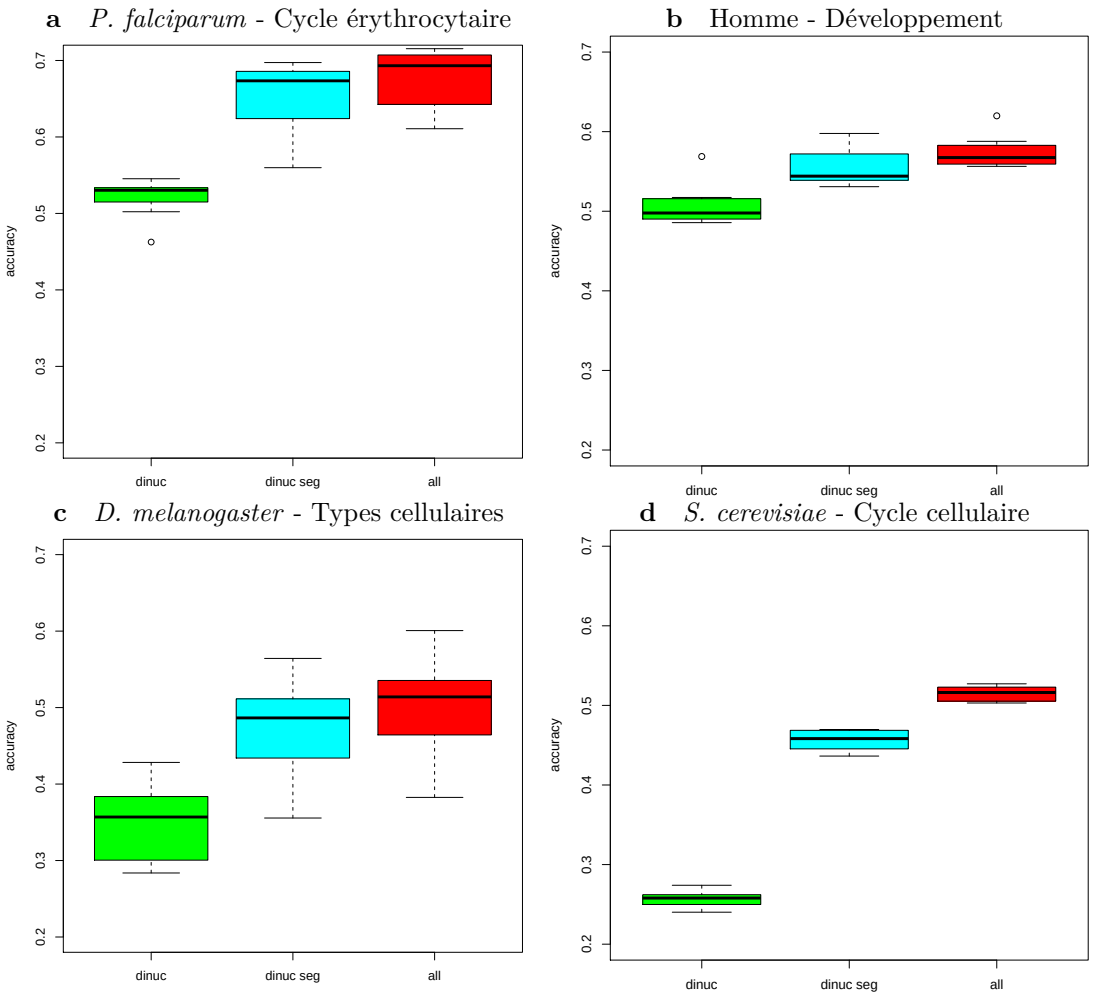


FIGURE 6.10 – Effet de la taille des k-mers et de la segmentation
Précision de la prédiction avec différentes variables. En vert, la fréquence des dinucléotides calculés sur toute la séquence ; en cyan, la fréquence des dinucléotides dans les meilleurs régions ; en rouge, la fréquence de tous les domaines identifiés par la méthode. Les boxplots sont calculés sur l'ensemble des conditions de chaque série (7 pour *P. falciparum*, 8 pour l'Homme, 8 pour *D. melanogaster*, et 5 pour *S. cerevisiae*.

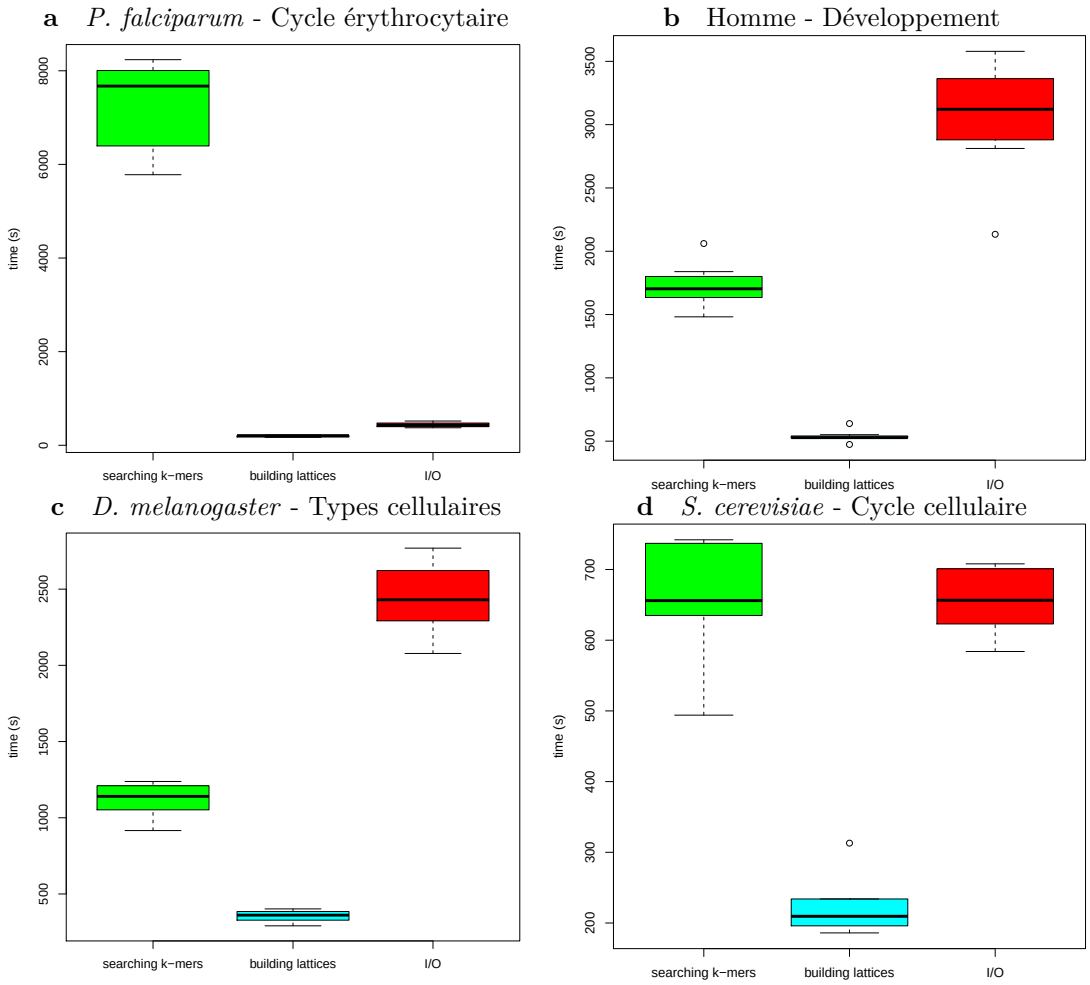


FIGURE 6.11 – Temps de calcul nécessaire par opération
En vert, le temps nécessaire pour identifier les occurrences des k-mers ; en cyan, le temps nécessaire à construire et explorer les treillis de corrélation ; en rouge, le temps nécessaire aux entrées/sorties du programme.

JASPAR [MFA+16]. Pour cela, nous nous sommes concentré sur les 10 variables les plus importantes identifiées par le prédicteur dans chaque condition. La Figure 6.13 présente la distribution de la taille des k-mers et des régions de ces variables, ainsi que le nombre médian d'occurrences du k-mer par séquence dans la région associée. Dans la majorité des cas, la longueur des régions (centaine de bp), la longueur des k-mers identifiés (majoritairement $k \leq 3$) et le nombre d'occurrences élevé (médiane > 20) semblent incompatibles avec la définition classique d'un site de fixation de TF, qui est normalement constitué d'une dizaine de bp et qui n'est pas connu pour se répéter un très grand nombre de fois sur une région aussi longue. Parmi les 154 variables que nous avons étudié, nous estimons que moins d'une dizaine pourrait s'apparenter à un motif de fixation. Il y a cependant un cas intéressant avec le k-mer AGACA identifié chez *P. berghei*, dont le nombre d'occurrences est strictement de 1 ou 0 pour la plupart des séquences. Un autre cas intéressant est la présence d'un petit motif TTA qui apparaît à la position exacte du TSS chez *C. elegans* et qui semble corrélé négativement à l'expression des gène.

6.2.3 La dynamique, la composition et la localisation diffèrent suivant les espèces

Nous avons ensuite cherché à vérifier si ces longues séquences étaient associées à des processus de régulation dynamiques ou statiques. Pour cela, chaque prédicteur appris dans une condition spécifique a été utilisé pour prédire l'expression dans les autres conditions de la même série et nous avons calculé la précision de ces nouvelles prédictions. Les courbes colorées de la Figure 6.12 résument ces expériences de permutation. Les comportements statiques et dynamiques semblent coexister, et dépendent fortement des espèces et des conditions. On observe que les prédicteurs appris sur les différentes types cellulaires chez l'Homme, *A. thaliana*, *D. melanogaster*, et *C. elegans* sont sensiblement les mêmes puisqu'ils sont globalement interchangeables. À l'inverse, on observe chez les deux *Plasmodium* que les prédicteurs appris aux différents stades ne sont pas interchangeables. Chez ces espèces, un prédicteur appris à un temps donné manquera de précision pour prédire les autres temps, même lorsque les permutations sont restreintes au cycle érythrocytaire. Pour *T. gondii*, le comportement est similaire, bien que beaucoup moins prononcé, alors que pour *S. cerevisiae*, le mécanisme semble complètement statique. Il est intéressant de noter qu'un comportement dynamique est également observé sur les séries du développement de *C. elegans* et *D. melanogaster* bien qu'aucune différence ne puisse être observée lors de la permutation des modèles appris sur différents tissus de ces organismes.

Nous avons ensuite cherché à comparer la composition et la localisation des séquences régulatrices identifiées dans les différentes conditions. Pour cela, nous

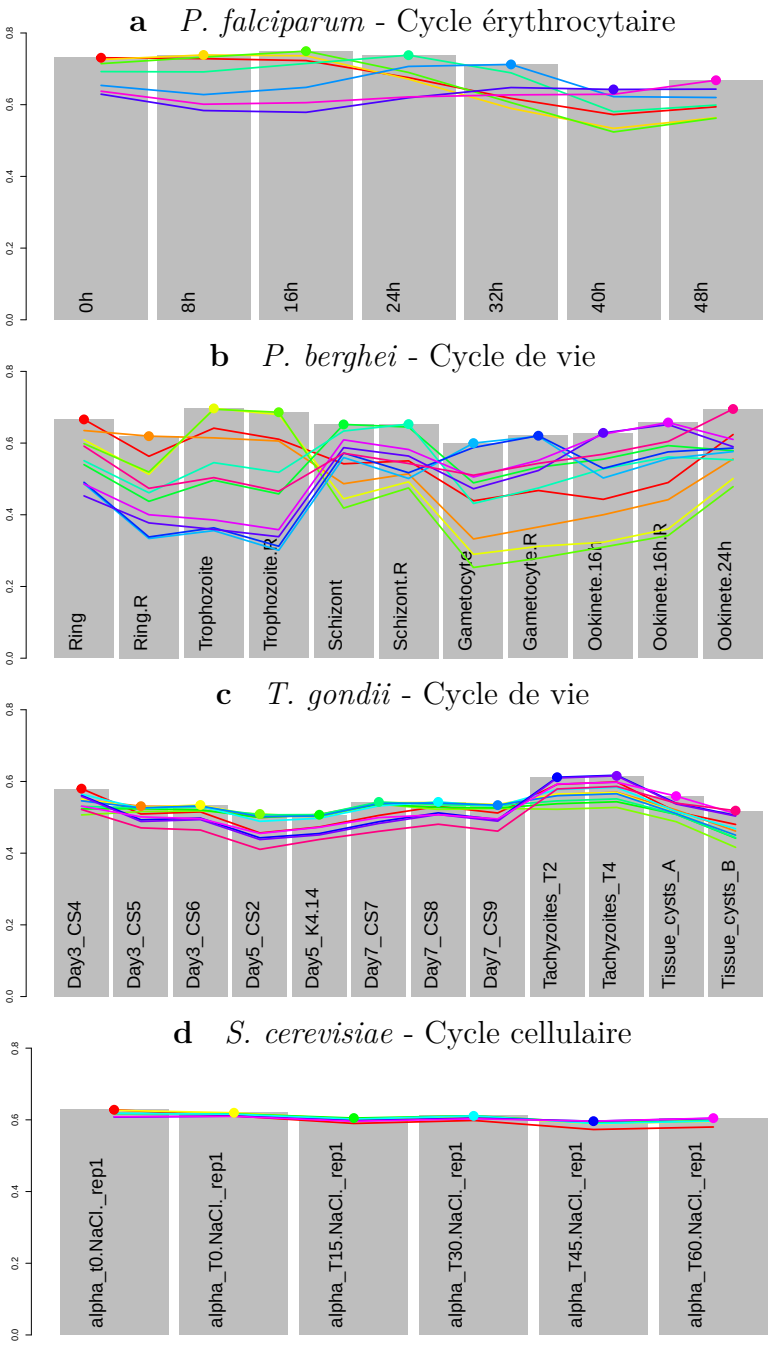


FIGURE 6.12 – Compilation des résultats de DExTER pour prédire l'expression des gènes codants chez différentes espèces (*partie 1/3*)
Les barres grises représentent la corrélation entre l'expression prédite et l'expression observée des prédicteurs appris dans différentes conditions. Les courbes colorées représentent la précision obtenue lorsqu'on utilise un prédicteur appris sur une condition spécifique pour prédire les autres conditions de la même série. L'échelle des abscisses est [0 : 0,8].

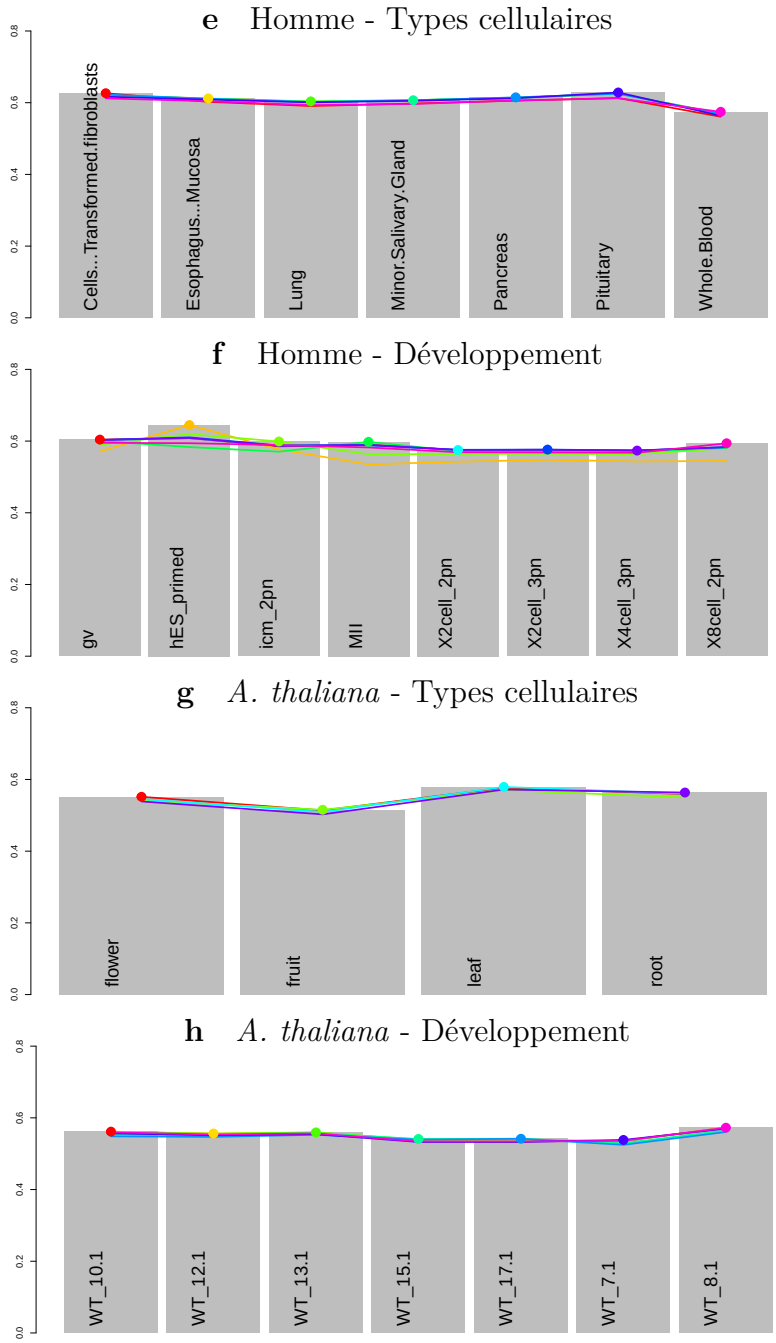


FIGURE 6.12 – Compilation des résultats de DExTER pour prédire l'expression des gènes codants chez différentes espèces (*partie 2/3*)
Les barres grises représentent la corrélation entre l'expression prédite et l'expression observée des prédicteurs appris dans différentes conditions. Les courbes colorées représentent la précision obtenue lorsqu'on utilise un prédicteur appris sur une condition spécifique pour prédire les autres conditions de la même série. L'échelle des abscisses est [0 : 0,8].

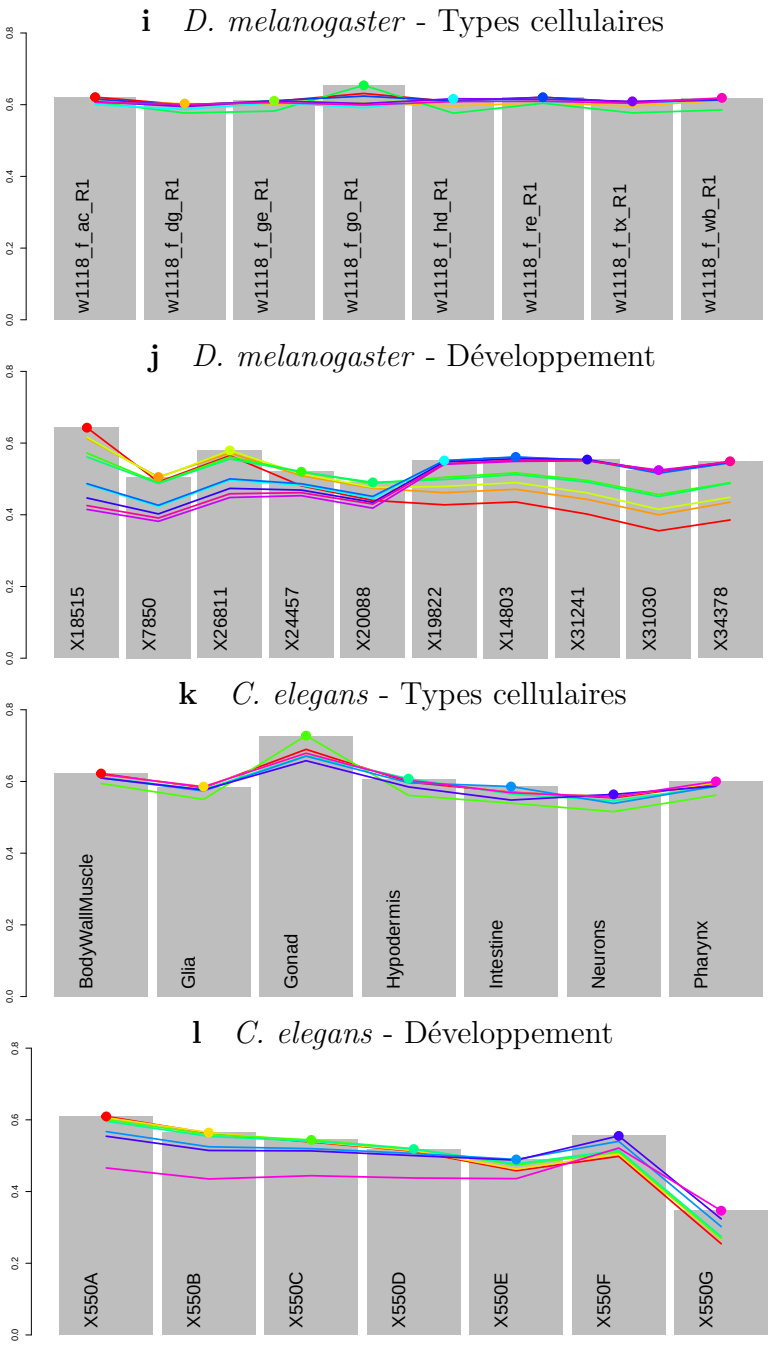


FIGURE 6.12 – Compilation des résultats de DExTER pour prédire l'expression des gènes codants chez différentes espèces (*partie 3/3*)
Les barres grises représentent la corrélation entre l'expression prédite et l'expression observée des prédicteurs appris dans différentes conditions. Les courbes colorées représentent la précision obtenue lorsqu'on utilise un prédicteur appris sur une condition spécifique pour prédire les autres conditions de la même série. L'échelle des abscisses est [0 : 0,8].

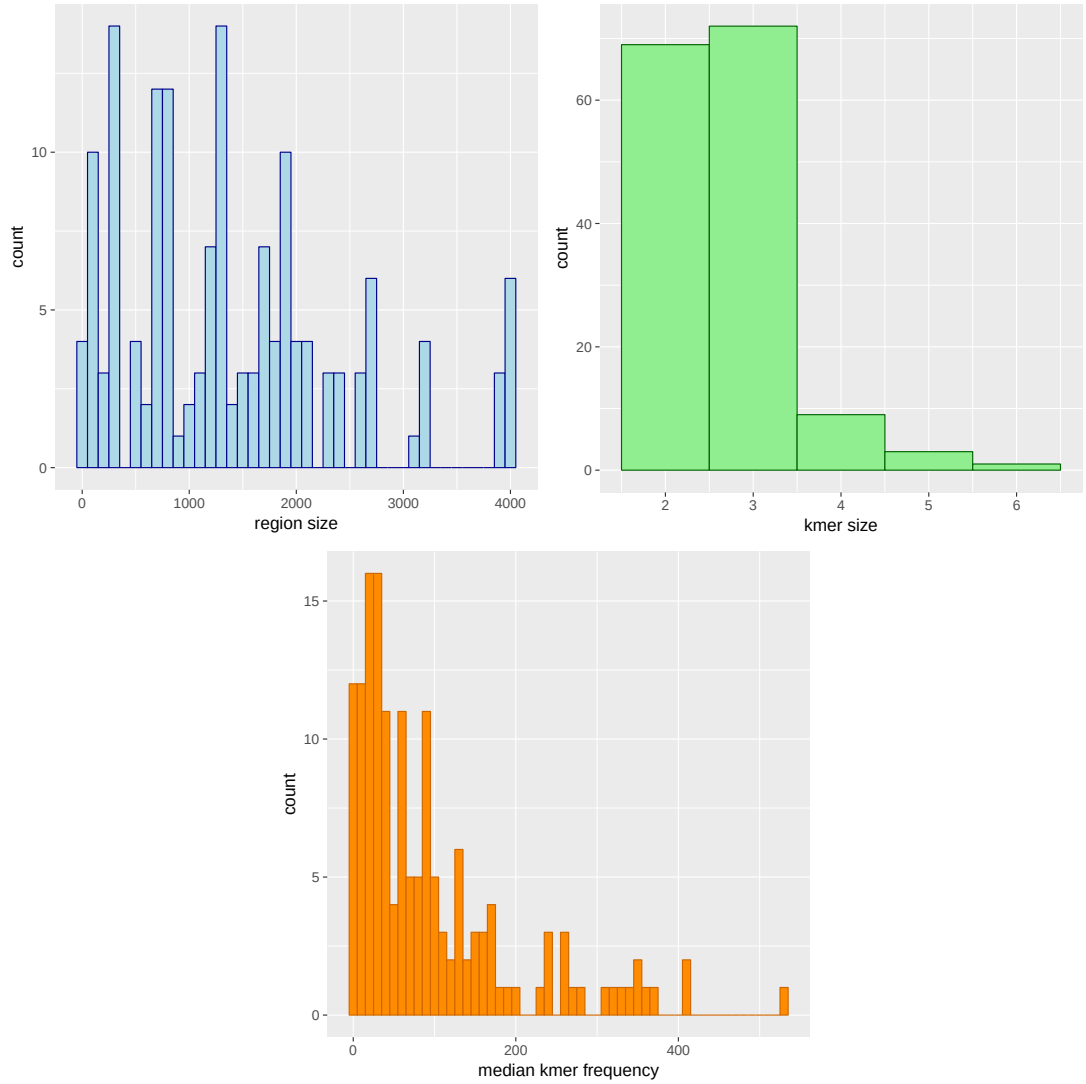
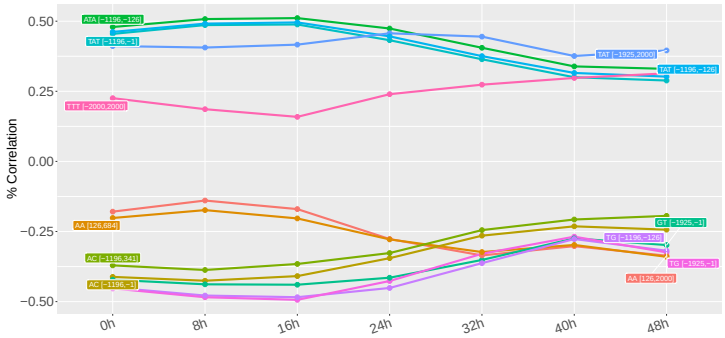


FIGURE 6.13 – Caractéristiques des domaines identifiés dans les différentes espèces et conditions
Le premier histogramme présente la longueur des régions sélectionnées, le deuxième présente la longueur des k-mers sélectionnés, le troisième le nombre médian d'occurrences du k-mer dans la région.

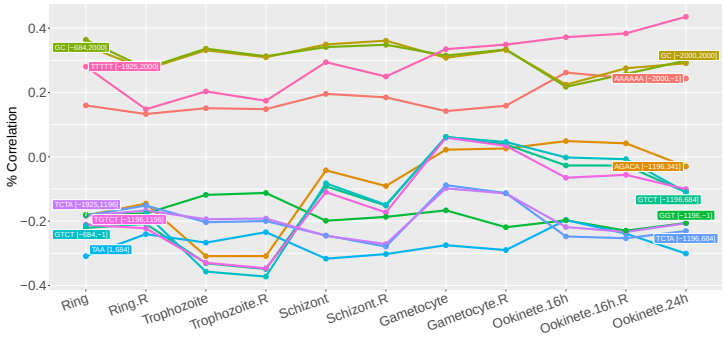
avons sélectionné les dix variables les plus importantes identifiées par le prédicteur dans chaque condition. La Figure 6.14 présente la corrélation de chaque variable à l'expression dans les différentes conditions. Conformément aux expériences précédentes, nous observons que pour *P. falciparum*, *P. berghei*, *T. gondii*, ainsi que pour le développement de *D. melanogaster* et *C. elegans*, la corrélation entre les différentes variables et l'expression est fluctuante entre les conditions, alors que dans les autres jeux de données, les corrélations sont plus stables. Pour *Plasmodium falciparum* on observe même un comportement sinusoïdal, qui n'est pas sans rappeler les motifs sinusoïdaux de l'expression des gènes observés lors du cycle érythrocytaire. Nous pouvons également observer que la localisation des domaines est diversifiée, et qu'elle dépend des espèces et des conditions. Par la suite, nous avons classé les variables en quatre catégories différentes suivant leur localisation : *upstream* ([-2000 : -500] avant le TSS), *downstream* ([+500 : +2000] après le TSS), centre ([-500 : +500] autour du TSS), et toute la séquence. Ensuite, pour chaque variable, nous avons calculé l'importance relative de chaque variable dans le prédicteur (voir Section 6.1.4). La Figure 6.15 présente les scores d'importance des «meilleures» variables et la Figure 6.16 présente l'importance de chaque région dans chaque condition. En observant ces résultats, on remarque plusieurs faits intéressant. Par exemple, nous pouvons observer que la région *upstream* est très importante chez *P. falciparum* (également chez *P. berghei* dans une moindre mesure) comparé aux autres espèces. Chez l'Homme, *D. melanogaster* et *C. elegans*, les prédicteurs favorisent plutôt les variables situées dans la région centre, en particulier dans l'analyse des différents types cellulaires. De manière intéressante, ces variables semblent moins importantes dans les premiers temps du développement chez *D. melanogaster* et *C. elegans*, mais elles gagnent en importance au cours du développement. De façon similaire, chez *P. falciparum* on observe une importance décroissante des variables situées dans la région *upstream* au cours du cycle érythrocytaire.

Enfin, nous avons cherché à mesurer le degré de conservation de ces régions régulatrices au cours de l'évolution. Pour cela, nous avons collecté les variables identifiées comme les plus importantes dans chaque espèce et chaque condition, et nous avons calculé la corrélation de chacune de ces variables à l'expression dans chacune des conditions. Ensuite, nous avons utilisé un *clustering* non supervisé pour classer ces corrélations (voir Figure 6.17a). Nous pouvons observer que les conditions sont correctement classées sur la seule base des corrélations (toutes les conditions d'une même espèce sont groupées ensemble). De manière intéressante, les conditions de *P. falciparum* et *P. berghei* se regroupent également très clairement et, avec un signal moins clair, avec les conditions de *T. gondii*, tandis que le reste des groupes ne semble pas être conforme à l'arbre phylogénétique des eucaryotes. En examinant la conservation de la corrélation de chaque variable individuellement, on obtient

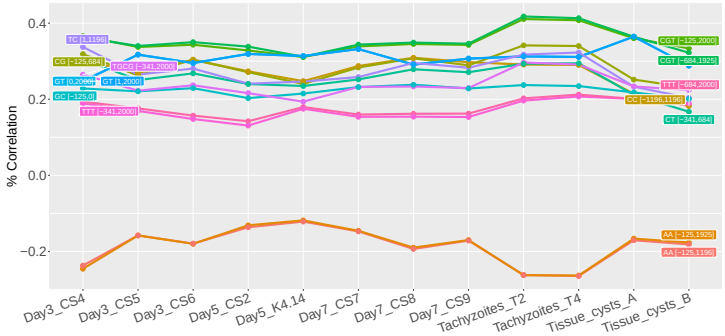
a *P. falciparum* - Cycle érythrocytaire



b *P. berghei* - Cycle de vie



c *T. gondii* - Cycle de vie



d *S. cerevisiae* - Cycle cellulaire

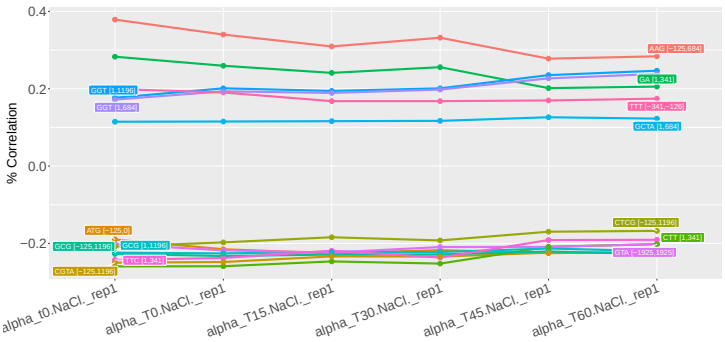
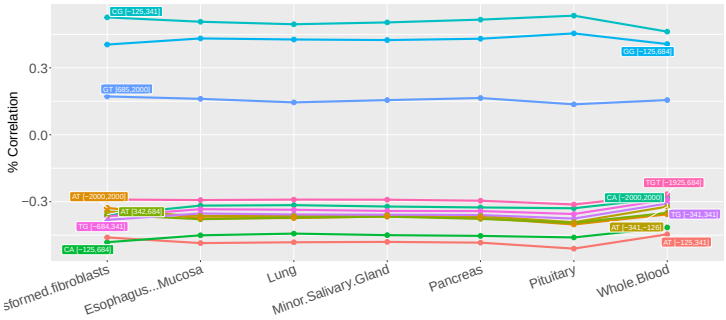
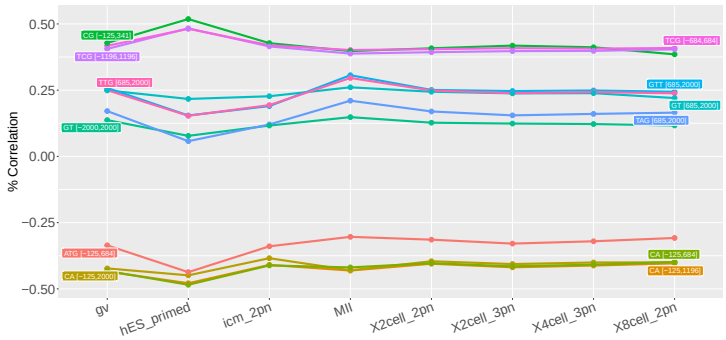


FIGURE 6.14 – Corrélations des variables les plus importantes dans chaque condition/espèce à l'expression des gènes (*partie 1/3*)
Dans chaque série de données, on a identifié les 10 variables les plus importantes de chaque condition, et on a calculé leur corrélation à l'expression dans toutes les conditions de la série.

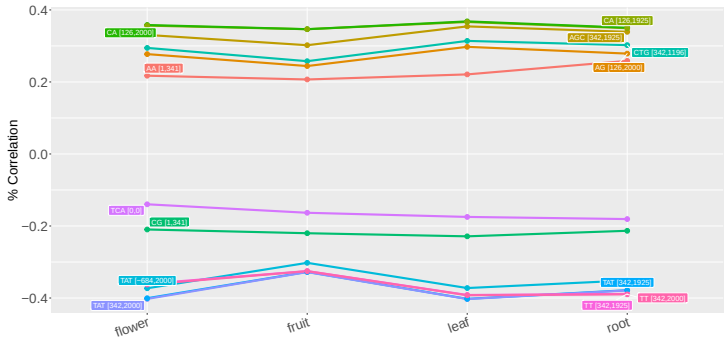
e Homme - Types cellulaires



f Homme - Développement



g *A. thaliana* - Types cellulaires



h *A. thaliana* - Développement

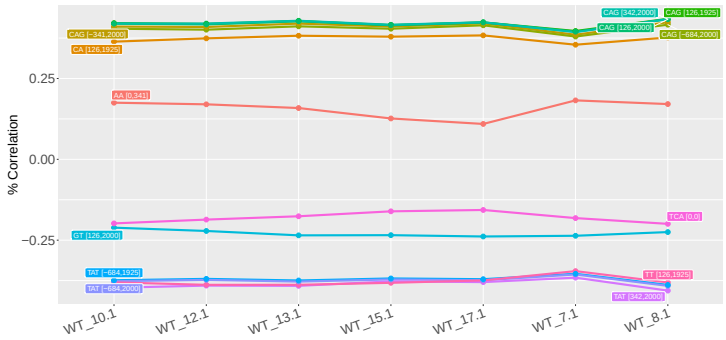
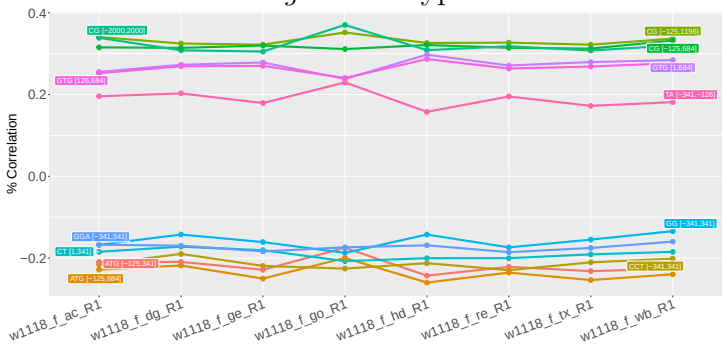
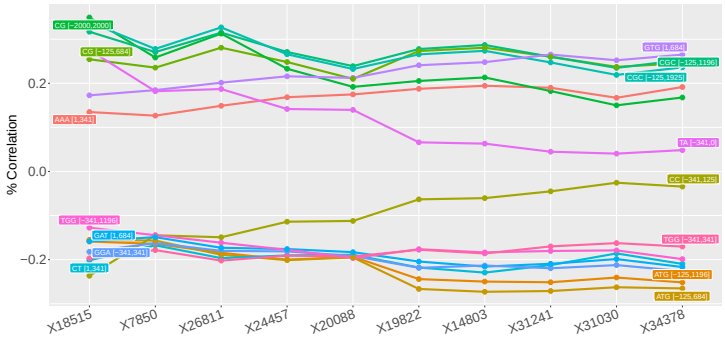


FIGURE 6.14 – Corrélations des variables les plus importantes dans chaque condition/espèce à l'expression des gènes (*partie 2/3*)
Dans chaque série de données, on a identifié les 10 variables les plus importantes de chaque condition, et on a calculé leur corrélation à l'expression dans toutes les conditions de la série.

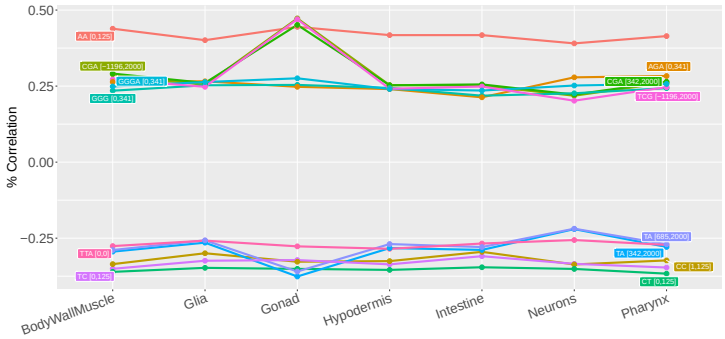
i *D. melanogaster* - Types cellulaires



j *D. melanogaster* - Développement



k *C. elegans* - Types cellulaires



l *C. elegans* - Développement

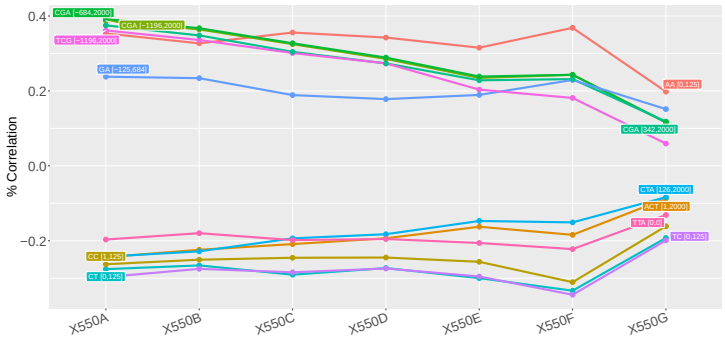


FIGURE 6.14 – Corrélations des variables les plus importantes dans chaque condition/espèce à l'expression des gènes (*partie 3/3*)
Dans chaque série de données, on a identifié les 10 variables les plus importantes de chaque condition, et on a calculé leur corrélation à l'expression dans toutes les conditions de la série.

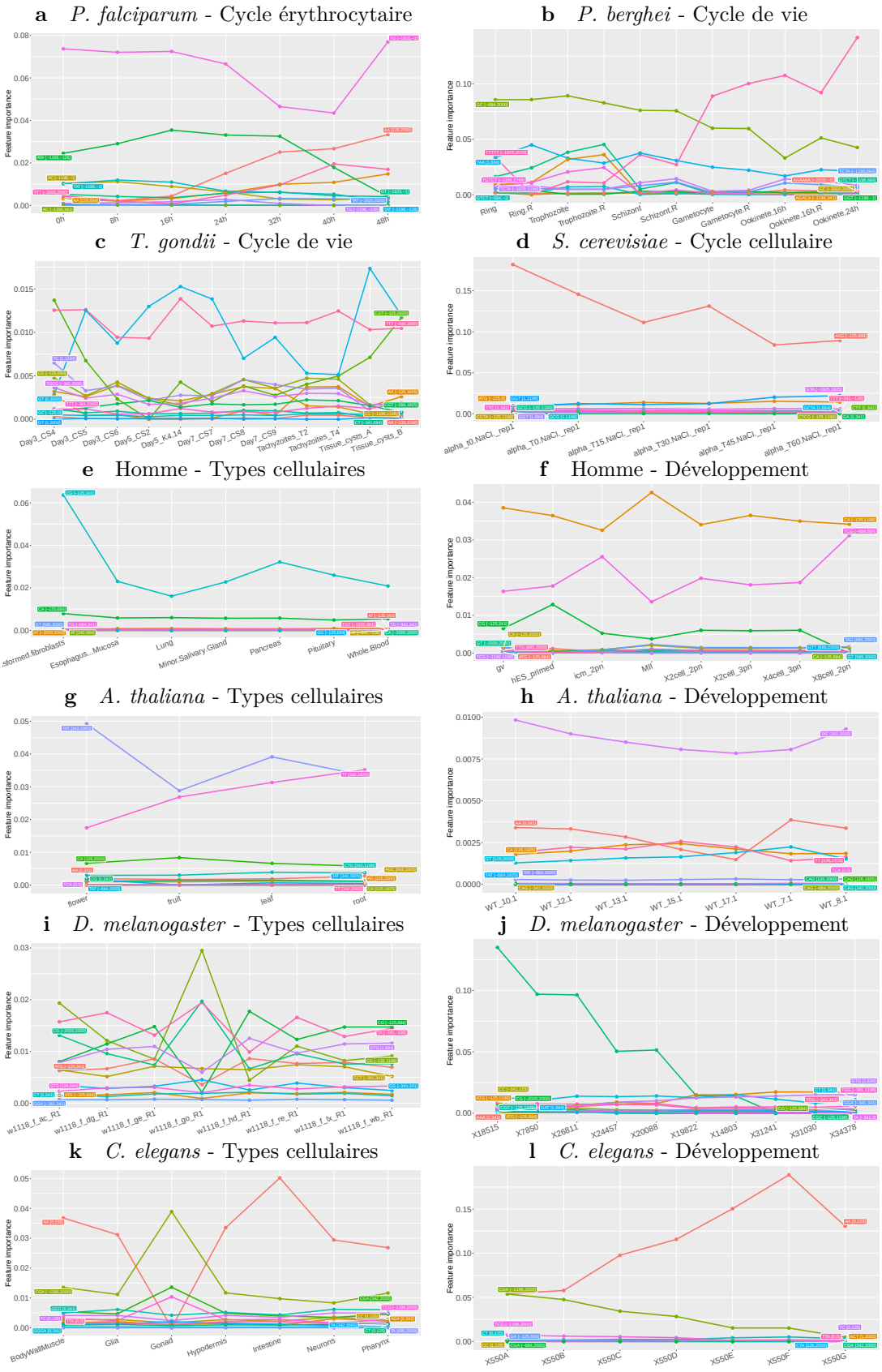


FIGURE 6.15 – Importance des variables sélectionnées dans chaque condition

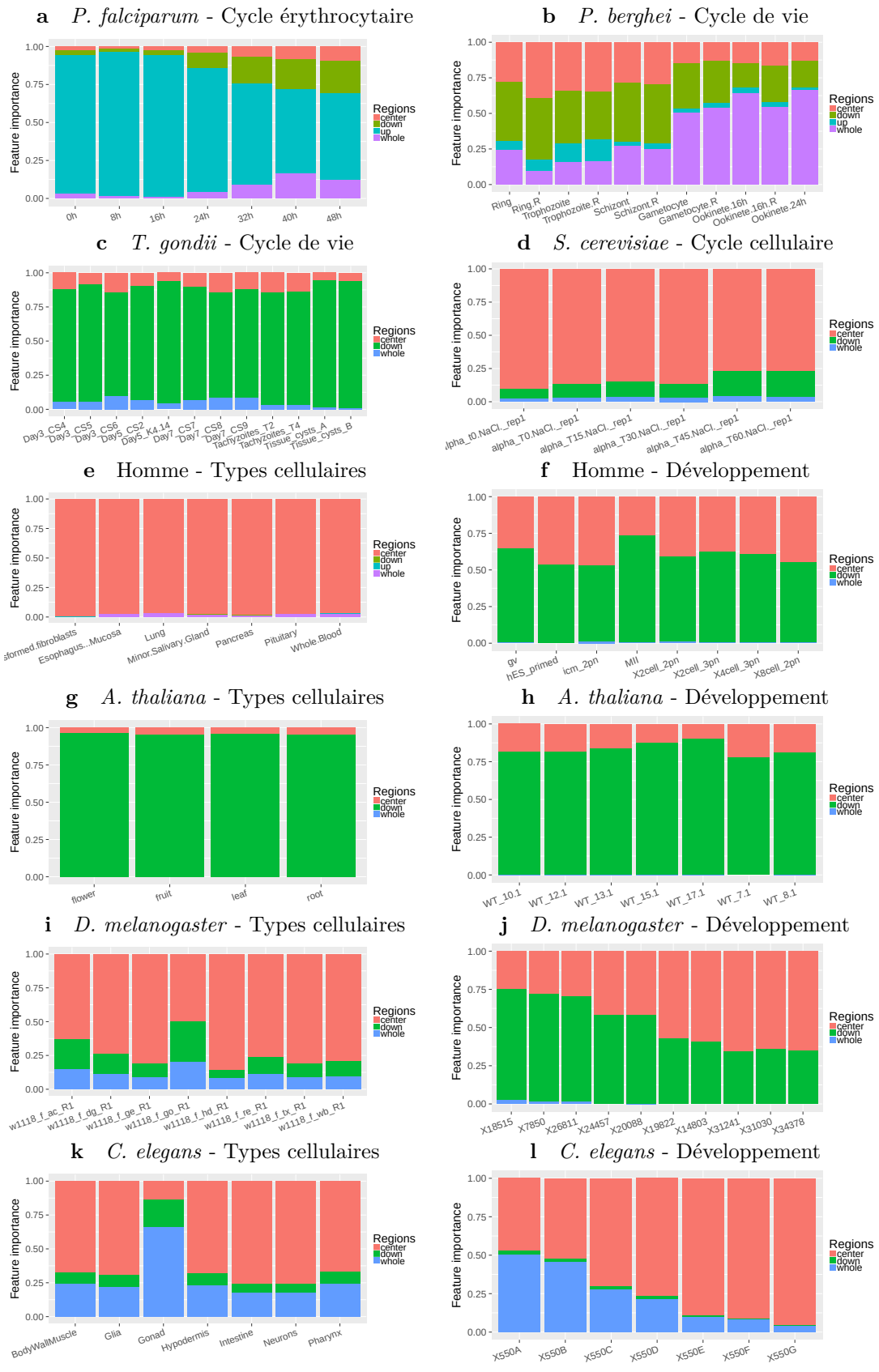


FIGURE 6.16 – Importance des régions *upstream*, *downstream*, centre, ou toute la séquence, pour prédire l'expression dans les différentes conditions
Nous avons identifié les 12 variables les plus importantes pour chaque condition. Ensuite les variables sont classées dans une des quatre régions possibles. Les différentes figures présentent la distribution des différentes régions dans chaque condition.

une vision plus précise de cette tendance générale (voir Figure 6.17b). Plusieurs variables sont corrélées à l'expression à la fois pour *P. falciparum* et *P. berghei* (*e.g.* ATA [-1196 : -126]), mais leur nombre est moindre que celles corrélées dans chaque espèce individuellement. Comme attendu, au niveau des espèces *Apicomplexa* le nombre de variables en commun est encore plus faible (*e.g.* TTT [-684 : 2000]). De manière similaire, certaines variables sont partagées par *D. melanogaster* et *C. elegans*, mais très peu de variables sont partagées au niveau des animaux ou des Unikonts (sauf la variable CG [-125 : 1196]). Par conséquent, s'il semble y avoir un signal phylogénétique dans ces données, la conservation est souvent limitée aux espèces les plus proches. On peut noter toutefois que ces analyses ont été effectuée sur la base de régions strictement identiques dans toutes les espèces. Étant donné que la taille des promoteurs, des UTR, des introns, ... dépendent des espèces, cette relative modicité de la conservation phylogénétique peut être une conséquence des règles strictes utilisées dans cette analyse.

6.2.4 Les domaines de régulation sont associés à des réglementations très dynamiques tout au long du cycle de vie de *Plasmodium falciparum* et ont des termes GO spécifiques

Comme expliqué ci-dessus, la précision des prédictions est particulièrement élevée pour *Plasmodium falciparum*, culminant à 74% au cours des premiers stades du cycle érythrocytaire. Bien que de longues séquences régulatrices soient également identifiées chez tous les eucaryotes étudiés, la précision plus élevée de *Plasmodium falciparum* ainsi que le comportement dynamique observé suggèrent que ces types de séquences régulatrices sont particulièrement importantes pour la régulation de l'expression des gènes chez cette espèce. *Plasmodium falciparum* apparaît donc comme un modèle de choix pour l'étude des mécanismes de régulation associés à de telles séquences.

Pour mesurer dans quelle mesure ces longues séquences régulatrices contrôlent l'expression des gènes tout au long de la vie de *Plasmodium falciparum*, nous avons effectué une analyse des données de Lopez-Barragan *et al.* [LBLQ+11] qui mesurent l'expression des gènes dans les étapes sexuée et asexuée du parasite. Les résultats sont visibles à la Figure 6.18a. Les résultats concordent avec ceux obtenus sur des données ciblant uniquement le cycle érythrocitaire, avec une précision supérieure à 70% pour les étapes Trophozoïte et Gamétocyte II. Ce qui est surprenant toutefois, c'est le comportement dynamique très fort du processus de régulation, déjà observé dans le cycle de vie de *P. berghei* : un modèle très précis sur Gamétocyte a une très faible précision aux stades asexués (particulièrement en Ring), et réciproquement. Ceci peut également être observé par les fluctuations élevées de corrélation entre

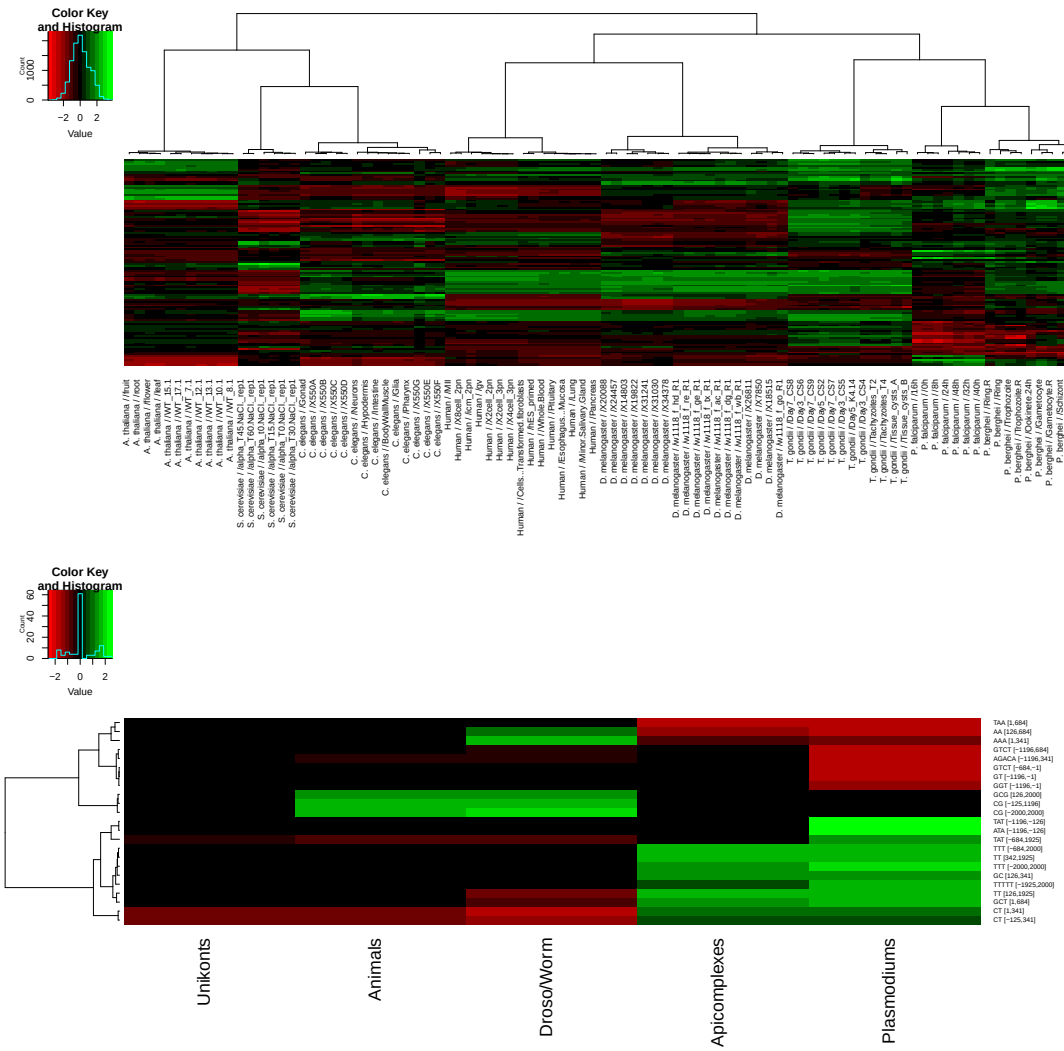


FIGURE 6.17 – Conservation des domaines modérée au cours de l'évolution
Nous avons identifié et sélection les 10 variables les plus importantes dans chaque condition. En haut, un clustering hiérarchique (méthode de Ward) est réalisé pour classer les conditions en fonction des corrélations des variables sélectionnées. En bas, la heatmap représente les variables dont la corrélation à l'expression semble conservée au niveau des taxons. Les variables qui ne montrent pas de conservation ont été masquées pour la lisibilité.

la fréquence des variables et l'expression des gènes (voir Figure 6.18c). Parmi les meilleures variables identifiées par DExTER aux différentes étapes, plusieurs sont similaires à celles identifiées dans les données du cycle érythrocytaire de Otto *et al.* [OWA+10] (par exemple, ATA et TG sur la région *upstream*, ou la répétition de T sur la séquence entière). Certaines autres semblent beaucoup plus corrélées à l'expression au stade sexué qu'au stade asexué (par exemple, TATAT sur la séquence entière a une corrélation qui va de 25% à 50%). Il est intéressant de noter que la variable AG dans la région *downstream* semble être positivement corrélée avec l'expression au stade asexué, mais négativement au niveau sexué. Dans l'ensemble, les variables en *upstream* semblent très importantes au stade asexué, mais n'ont pratiquement aucune utilité pour les stades gamétocyte et ookinète (voir Figure 6.18b).

Nous avons ensuite utilisé la méthode GSEA [STM+05] pour analyser certaines des variables qui montrent la plus forte corrélation avec l'expression dans différentes phases. Il est intéressant de noter que les gènes enrichis en variables spécifiques sont également associés à des termes spécifiques de GO (voir Figure 6.19). Par exemple, les gènes avec un taux élevé de ATA sur la région *upstream* sont associés à une expression élevée dans les premières phases du cycle érythrocytaire et sont impliqués dans la traduction. Les gènes avec un taux élevé de TTT sur la toute séquence ont une expression élevée sur les derniers temps et sont impliqués dans la régulation des transports. De manière similaire, les gènes avec un faible taux de AA sur les régions *downstream* sont associés à une expression élevée à des moments tardifs et sont impliqués dans différents processus métaboliques. Enfin, les gènes avec un taux élevé de TATATA sur la séquence entière sont plus fortement exprimés dans le gamétocyte et semblent impliqués dans l'assemblage de la chromatine.

6.2.5 Liens avec la régulation transcriptionnelle et post-transcriptionnelle chez *Plasmodium falciparum*

Nous avons ensuite analysé plus en détail la chronologie des variables identifiées dans le cycle érythrocytaire. La Figure 6.20a présente les 10 variables les plus importantes identifiées dans chaque étape du cycle erythrocytaire (représentant au total 12 variables différentes). Les heatmaps à gauche et à droite présentent les variables avec la plus forte corrélation en début (0h-16h) et en fin (24h-48h) de cycle erythrocytaire respectivement. En conséquence, avec les résultats présentés à la Figure 6.16, la région *upstream* est davantage corrélée à l'expression dans les premiers temps, tandis que la région *downstream* et la séquence entière sont davantage corrélées aux derniers temps. Ensuite, nous avons estimé la spécificité de brin associée à chaque variable. Pour cela, nous avons calculé la fréquence du

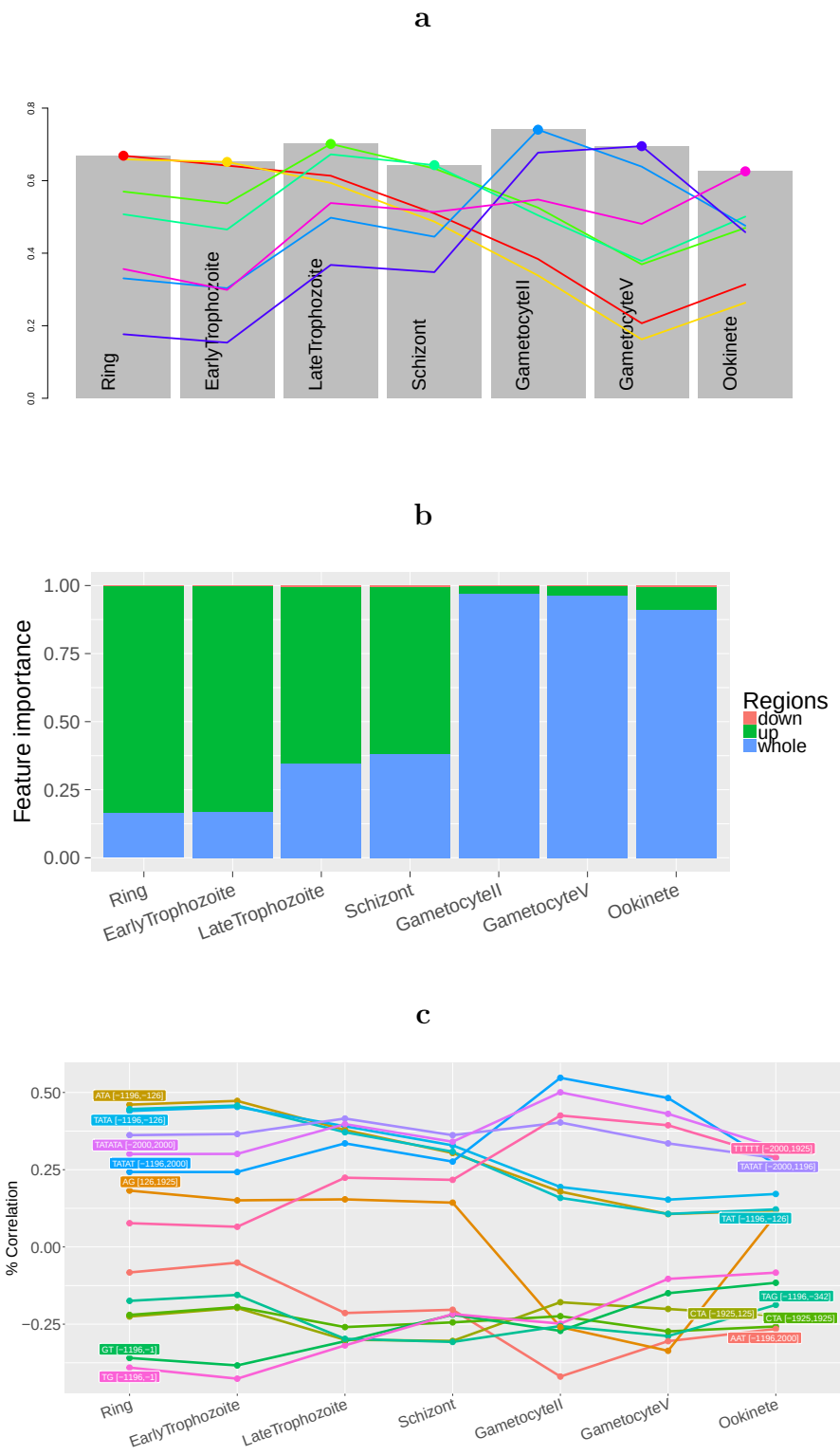


FIGURE 6.18 – Importance des domaines au cours du cycle de vie de *Plasmodium falciparum*
a : en gris la précision du prédicteur, les courbes de couleur représentent la précision d'un modèle appris à un stade donné et appliqué sur les autres stades. **b** : estimation de l'importance des régions pour la prédiction à chaque stade. **c** : corrélation des 12 variables les plus importantes à l'expression des gènes pour chaque stade.

FIGURE 6.19 – Analyse GSEA des variables identifiées

La méthode GSEA permet d'identifier un enrichissement en certains termes GO dans les variables identifiées par notre méthode. Plus la couleur tend vers le rouge, plus l'enrichissement est fort. Dans cette analyse, les gènes ont été classés suivant la fréquence de chacune des variables, et la méthode GSEA a été utilisée pour calculer des enrichissements GO dans les gènes situés, suivant les cas, en tête ou queue de liste. L'enrichissement a été calculé sur les gènes de tête pour les variables corrélées positivement à l'expression, et sur les gènes de queue pour les variables corrélées négativement à l'expression.

k-mer correspondant dans la région identifiée sur le brin + (comme cela est fait dans DExTER) et sur le brin –, et nous avons comparé les corrélations entre ces deux fréquences et expression. Les variables pour lesquelles les corrélations diffèrent entre les brins sont considérées comme spécifiques aux brins. Sur la Figure 6.20a, la spécificité de brin est représentée avec un code couleur allant du bleu (pas de spécificité de brin) à l'orange (spécificité de brin). Il est intéressant de noter que toutes les variables en *upstream* ne présentent que peu ou pas de spécificité de brin, alors que deux des trois variables avec la corrélation la plus élevée aux derniers temps sont spécifiques au brin.

L'absence de spécificité de brin dans les variables *upstream* et la présence de spécificité dans quelques variables *downstream* suggèrent que les variables *upstream* et *downstream* pourraient être impliquées dans les mécanismes de régulation transcriptionnelle et post-transcriptionnelle, respectivement. Pour évaluer ce point, nous avons analysé les données de Painter *et al.* [PCS+18]. Dans cet article, les auteurs mesurent séparément le niveau de la transcription naissante et d'ARNm stabilisé au cours du cycle érythrocytaire. Nous avons exécuté DExTER sur ces deux types de données et pour chaque temps, nous avons identifié les 10 variables les plus importantes dans chaque condition. Parmi les 15 variables différentes, 4 sont nettement plus corrélées au niveau de la transcription naissante qu'au niveau des transcrits stabilisés, tandis que 5 autres sont davantage associées à des transcrits stabilisés qu'à la transcription naissante (voir Figure 6.20b) (les variables restantes ne peuvent pas être clairement associées à la transcription naissante ou aux transcrits stabilisés). De manière remarquable, toutes les variables associées à la transcription naissante sont à la fois en *upstream* et non spécifique au brin, tandis que les variables associées à la stabilisation de l'ARNm sont en *downstream* et spécifiques à un brin.

6.2.6 Lien avec les marques épigénétiques

Comme on l'a vu dans le chapitre précédent, les auteurs de Read *et al.* [RCL+19] ont montré que l'expression des gènes de *Plasmodium falciparum* peut être prédite avec une bonne précision à partir de différentes marques épigénétiques. Notamment, les modification d'histone H2A.Z, H3K9ac et H3K4me3 dans les promoteurs et dans le corps des gènes semblent être parmi les marques les plus prédictives. Nous avons donc cherché à déterminer si certaines des variables prédictives identifiées par notre approche pouvaient en réalité être liées à ces marques spécifiques. Pour ce faire, nous avons utilisé les données de Bartfai *et al.* [BHSA+10] pour calculer les signaux de H2A.Z, H3K9ac et H3K4me3 en *upstream* et en *downstream* du codon AUG de chaque gène et nous avons exécuté DExTER pour prédire ce signal au lieu de l'expression du gène.

Globalement, la précision de la prédiction fluctue autour de 60% pour les dif-

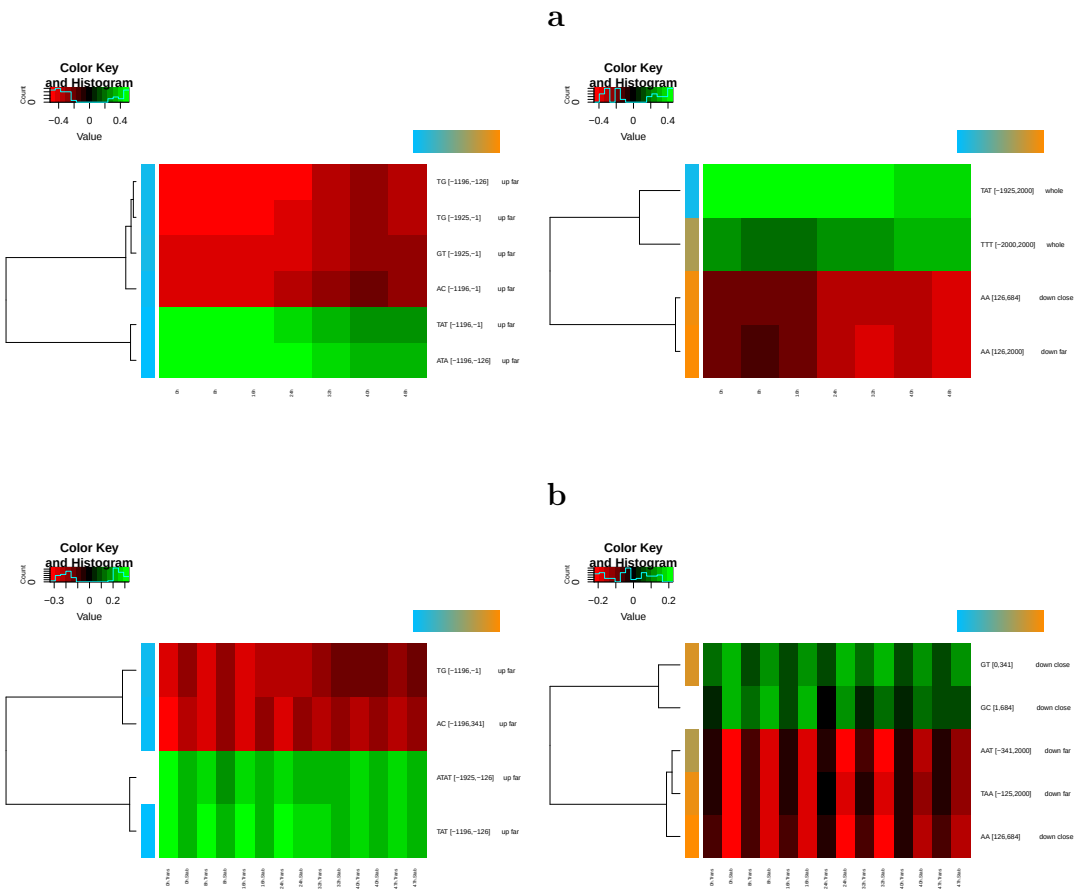


FIGURE 6.20 – Variables identifiées dans le cycle érythrocytaire de *Plasmodium falciparum*
a : Heatmaps des corrélations entre l'expression des gènes et les variables les plus importantes à chaque temps des données de Otto *et al.* [OWA+10]. La heatmap de gauche correspond aux variables avec une forte corrélation dans les premiers temps (0h - 16h), tandis que la heatmap de droite correspond aux variables avec une plus forte corrélation dans les derniers temps (24h - 48h). **b** : Heatmaps des corrélations entre l'expression des gènes et les variables les plus importantes à chaque temps des données de Painter *et al.* [PCS+18]. La heatmap de gauche correspond aux variables avec une plus forte corrélation avec les données de transcription, tandis que la heatmap de droite correspond aux variables avec une plus forte corrélation avec les données de stabilisation des ARNm. La couleur bleu/orange indique la spécificité de brin de chaque variable (orange : spécifique ; bleu : non spécifique).

férentes marques mais sans atteindre la précision obtenue lors de la prédiction de l'expression des gènes (voir Figure 6.21). L'analyse des variables les plus importantes des différents prédicteurs montre que plusieurs variables identifiées pour la prédiction de l'expression des gènes sont également identifiées lors de la prédiction des marques d'histones (voir Figure 6.22). Nous avons alors comparé les valeurs de corrélation calculées avec l'expression des gènes et les marques d'histones. Fait intéressant, parmi toutes les variables, la variable TTT semble être la seule qui soit significativement plus corrélée aux marques d'histones qu'à l'expression des gènes. Alors que cette variable correspond à environ 25% de corrélation avec l'expression des gènes (voir Figure 6.14a), elle atteint presque 40% avec le signal de H2A.Z en *upstream* de l'AUG, ainsi qu'avec le signal H3K4me3 à 40hpi (heures après infection) (voir Figure 6.22).

6.3 Discussion

Nous avons appliqué DExTER sur des séquences de 4 001bp centrées sur le TSS des gènes codants de deux espèces de *Plasmodium* et plusieurs autres espèces eucaryotes. Suivant les espèces, la méthode identifie différentes grandes régions (plusieurs dizaines, voir centaines de bp) dont la fréquence en un certain k-mer est corrélée avec l'expression des gènes. Nous avons émis l'hypothèse que ces longues séquences biaisées pourraient constituer une nouvelle classe d'éléments régulateurs, que l'on nomme domaines de régulation, différents des sites de fixation classiques des facteurs de transcription. Les modèles appris sur la base de ces domaines de régulation permettent de prédire l'expression avec une précision comprise entre 50% et 60% suivant les espèces. Chez les *Plasmodium* cette précision dépasse même les 70%, ce qui indique que ces domaines de régulation semblent avoir un rôle prédominant dans ces espèces. De plus, nos analyses montrent que ce mode de régulation est dynamique, avec différentes régions et compositions impliquées lors du cycle de vie des *Plasmodium*. En dehors des apicomplexes, ce mécanisme semble plus statique, à l'exception du développement embryonnaire de *Drosophila melanogaster* et *Caenorhabditis elegans*. D'autres analyses montrent chez *P. falciparum* une dichotomie claire parmi les domaines identifiés : ceux localisés en amont du TSS semblent principalement impliqués dans la régulation transcriptionnelle, tandis que ceux localisés en aval du TSS semblent principalement impliqués dans la régulation post-transcriptionnelle.

Il est intéressant de noter que ce travail pourrait être enrichi sur de nombreux points. Tout d'abord, nous n'avons malheureusement pas eu le temps de réaliser une comparaison des résultats de DExTER avec d'autres méthodes de prédiction de l'expression. À notre connaissance, il n'existe pas à ce jour de méthode d'extraction de variables semblables aux domaines de régulation identifiés par DExTER, mais

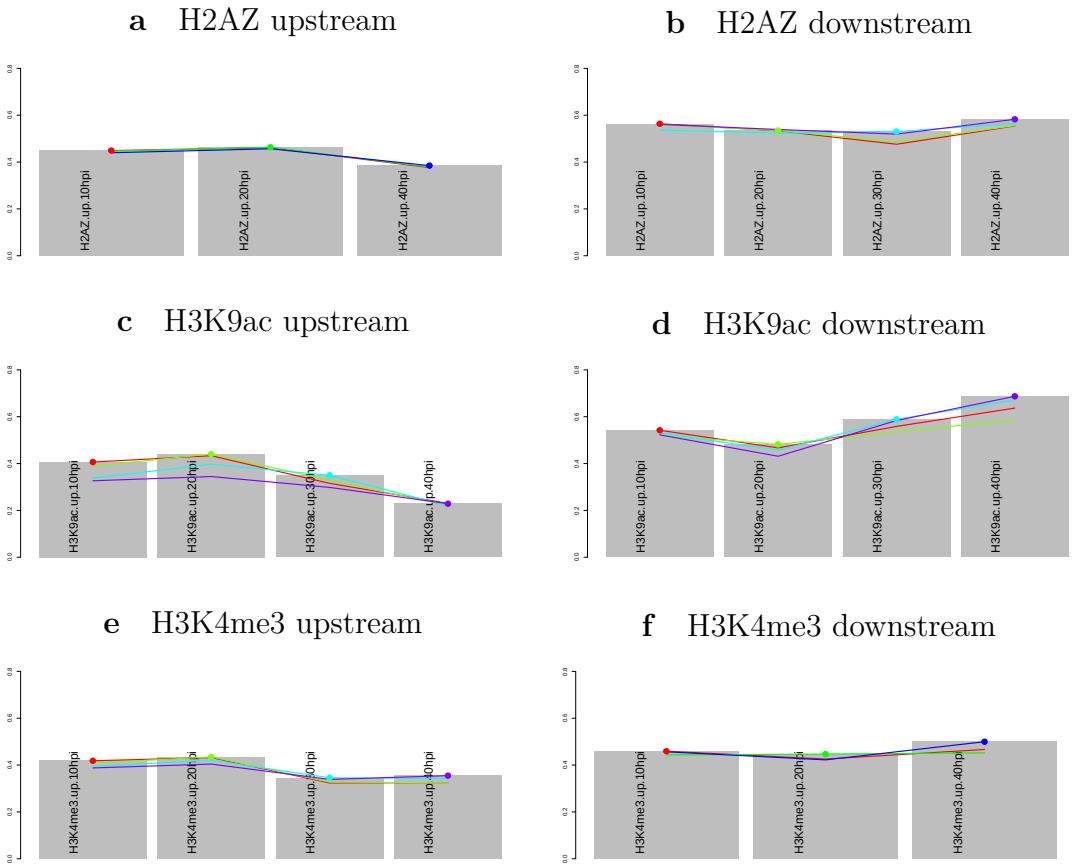


FIGURE 6.21 – Prédictions de différentes marques épigénétiques
Les barres grises représentent la précision des prédicteurs pour différentes marques épigénétiques. Les courbes colorées représentent la précision obtenue lorsqu'on utilise un prédicteur appris sur une condition pour prédire les autres conditions de la même série. L'échelle des abscisses est [0 : 0,8].

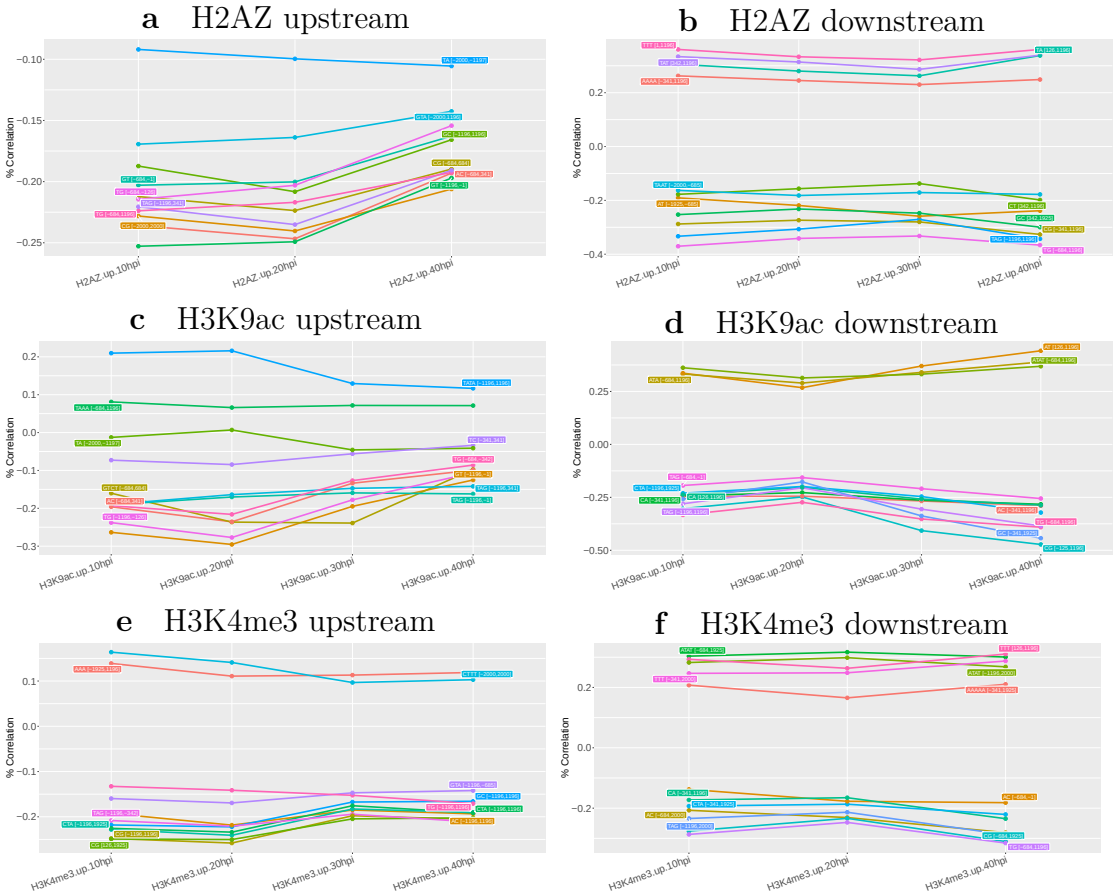


FIGURE 6.22 – Corrélation des variables les plus importantes dans chaque condition
Dans chaque série de données, on a identifié les 10 variables les plus importantes de chaque condition, et on a calculé leur corrélation à l'expression dans toutes les conditions de la série.

nous pourrions tout à fait comparer les performances des prédicteurs appris avec DExTER et les performances d'un réseau de neurones convolutifs appris dans les mêmes conditions. Cela pourrait nous fournir une idée de la quantité d'information restante à extraire dans l'une ou l'autre des approches.

Un autre point d'amélioration serait de continuer notre travail sur la quantification de l'importance des variables identifiées. La procédure que nous avons proposée est une procédure *ad hoc* qui mériterait d'être plus amplement testée et évaluée. Il serait intéressant d'évaluer d'autres solutions comme les valeurs de Shapley afin de voir si on observe des différences significatives dans le degré d'importances des variables identifiées.

Nous pourrions également imaginer une procédure de re-segmentation *a posteriori* des variables les plus importantes. En effet, lors de l'exploration de DExTER nous devons choisir un nombre de régions unitaire et le nombre choisi a un impact à la fois sur la précision des résultats mais aussi sur le temps de calcul de la méthode. D'après nos premières expériences, il ne semble pas nécessaire de découper la séquence en un grand nombre de régions pour obtenir une bonne précision. Cependant, d'un point de vue biologique, il est important d'avoir une définition de la région de régulation la plus précise possible. Or cette précision dépend directement de la longueur des sous-séquences à la base du treillis. Une solution serait donc de re-segmenter les variables importantes afin d'identifier la région fonctionnelle plus précisément. On gagnerait donc en précision pour un coût modéré en temps de calcul.

Une autre extension possible de DExTER serait d'ajouter des informations concernant les régions fonctionnelles connues *a priori* pour chaque gène. Nous pourrions par exemple inclure, en plus du TSS, la position du codon *start*, la position du codon *stop*, la position du premier intron, ... La difficulté est que nous devrions traiter cette fois des séquences de taille différente pour chaque gène. Une solution serait de segmenter chaque région fonctionnelle (les 5' UTR par exemple) en un nombre fixe de sous-régions. Ces sous-régions seraient de taille variable pour chaque gène, mais les treillis de séquences obtenus auraient la même «architecture», et il serait donc tout à fait envisageable de construire le treillis de corrélation et de poursuivre l'exploration comme actuellement. Cette solution présente donc l'avantage de considérer plusieurs points d'alignements pour chaque séquence et d'intégrer les régions fonctionnelles déjà connues. Cela pourrait fournir de nouveaux résultats et de nouvelles hypothèses biologiques. Avec cette nouvelle procédure nous pourrions même imaginer utiliser notre prédicteur pour définir les régions fonctionnelles des gènes mal annotés ou des gènes d'un génome proche du génome utilisé pour apprendre le modèle. Si on prend par exemple un gène pour lequel la région 5' UTR n'est pas annotée (TSS inconnu) on peut imaginer chercher la position approximative du TSS qui permettrait à notre modèle de

prédire l'expression la plus proche de l'expression mesurée pour ce gène.

Nous avons également imaginé une autre stratégie d'apprentissage du prédicteur que nous souhaiterions essayer. À l'heure actuelle nous traitons chaque domaine individuellement sur la seule base de leur corrélation à l'expression. C'est seulement une fois l'exploration terminée que nous apprenons un modèle sur la base des domaines sélectionnés. Une autre stratégie serait de construire le modèle de manière itérative dans l'exploration de l'espace des k-mers et des régions. Pour cela, nous commencerions par apprendre un modèle sur la base des compositions dinucléotidiques calculées sur toute la séquence. Ensuite nous pourrions explorer les différents domaines possibles en se basant non plus sur la corrélation entre fréquence des k-mers et expression, mais entre la fréquence des k-mers et l'erreur résiduelle du modèle appris à l'étape précédente. Ainsi, si la corrélation est significative, nous pourrions intégrer ce nouveau domaine à la liste des variables et entraîner un nouveau modèle, puis répéter l'opération tant que possible. Cette stratégie présente l'avantage de concentrer l'exploration sur le gain de performance du modèle, ce qui n'est pas directement le cas dans la procédure actuelle. L'inconvénient de cette stratégie est le coût en temps de calcul. En effet, la complexité en temps de l'apprentissage d'un modèle de régression avec une pénalisation LASSO est de $\mathcal{O}(K^3 + K^2 n)$, avec K le nombre de variables en entrée et n le nombre d'exemples [EHJ+04]. Nous sommes dans le cas où $K < n$ donc la complexité serait de $\mathcal{O}(K^2 n)$. Si on implémentait cette solution il faudrait donc faire particulièrement attention au nombre de variables que l'on injecte dans le modèle et à quelle fréquence on actualise le modèle (à chaque nouvelle variable, chaque passage aux (k+1)-mers, ...).

D'un point de vue plus technique, il existe encore des améliorations possibles. Par exemple, nous pourrions étudier d'autres solutions pour la recherche des occurrences des k-mers. Nous utilisons actuellement le logiciel MOTIF mais celui-ci n'est pas pleinement efficace car il nécessite de recharger l'index des séquences pour chaque recherche de k-mer. De plus, son installation est relativement difficile. Afin de rendre l'installation du logiciel DExTER la plus simple possible il faudrait effectuer un comparatif des solutions alternatives voire implémenter la transformée de Burrows-Wheeler (BWT) et le FM-index directement dans DExTER. Une autre solution pourrait être de développer une interface web de DExTER et d'héberger le logiciel sur la plateforme ATGC de l'équipe MAB. Ainsi, les utilisateurs n'auraient pas besoin d'installer le logiciel. Cependant, il faudrait réaliser une étude pour estimer les ressources nécessaires pour déployer un tel service.

Une limite de la procédure d'exploration actuelle de DExTER est sa restriction à l'espace des k-mers simples. Nous pourrions étudier une extension de l'algorithme pour intégrer des motifs plus complexes. Pour cela, en plus de l'alphabet classique des nucléotides, nous pourrions utiliser les codes IUPAC. Cela permettrait par

exemple de rechercher des k-mers avec un gap (un nucléotide quelconque). Nous pouvons imaginer un gain de performance dans le sens où si le vrai motif fonctionnel contient des positions faiblement restreintes (par exemple A ou T, tout sauf A, ...) il y a des chances que DExTER n'en capture pas toutes les composantes du fait des règles strictes de l'exploration. Cependant, l'espace des k-mers possibles à explorer passerait de 4^K à 15^K, avec K la longueur maximale des k-mers à explorer. Il serait certainement nécessaire d'adapter la stratégie d'exploration pour gérer ces nouveaux motifs.

La méthode que nous avons développé a été utilisée pour prédire l'expression des gènes. Cependant comme on l'a vu dans les analyses des données de variants d'histones chez *Plasmodium falciparum*, son heuristique d'exploration de l'espace des k-mers et des régions peut tout à fait être adaptée pour prédire d'autres données biologiques. En effet, nous pouvons voir le critère d'optimisation et le prédicteur utilisés comme des composants modulables de la méthode. Nous avons pu expérimenter cela avec Océane Cassan, en stage dans l'équipe, qui a adapté la méthode DExTER pour essayer de prédire des expériences de ChIA-PET concernant les interactions chromosomiques. Lors de son stage, Océane a pu montrer qu'il est possible, dans une certaine mesure, de prédire l'interaction de deux régions chromosomiques, simplement par leur composition nucléotidique. De la même manière, nous pourrions fournir à l'utilisateur une librairie de critères d'optimisation et de prédicteurs pour adapter la méthode à différents types de données et rechercher les déterminants nucléotidiques qui semblent gouverner le mécanisme étudié.

Enfin concernant les résultats obtenus chez *Plasmodium falciparum*, nous sommes actuellement entrain de valider expérimentalement certaines des prédictions de DExTER. Pour cela, nous avons pris contact avec Jose-Juan Lopez-Rubio, spécialiste de la régulation de l'expression chez *Plasmodium falciparum* et de sa manipulation génétique. Cette équipe de l'INSERM a notamment développé un protocole d'édition du génome de *Plasmodium falciparum* en utilisant le système CRISPR-Cas9. En concertation avec son équipe, nous avons imaginé des promoteurs synthétiques (avec les contraintes inhérentes au génome particulier de *P. falciparum*) qui permettraient de contrôler finement l'expression d'un gène rapporteur (par exemple la GFP). Les promoteurs synthétiques sont issus d'une recombinaison de fragments de promoteurs réels que nous avons identifié à l'aide des meilleurs variables de DExTER. Ainsi, nous pouvons prédire l'expression attendu d'un gène avec différents promoteurs et nous pourrons vérifier si cela concorde avec les résultats *in vivo*. Ce travail est en cours. Les constructions synthétiques ont été réalisées et intégrées au génome de *Plasmodium falciparum*. Nous attendons maintenant les résultats de la phase de contrôle, et les mesures d'expression de la GFP.

Conclusion générale

Nous nous sommes intéressé dans ce manuscrit au développement de deux nouvelles méthodes pour la découverte de nouvelles familles de motifs dans les séquences biologiques, et plus précisément de motifs longs que l'on désigne sous le terme de domaines. Ces motifs se trouvent classiquement dans les séquences protéiques, où ils prennent le nom de domaines protéiques, mais également, et c'est plus nouveau, dans les séquences nucléiques où ils semblent liés à la régulation de l'expression.

La première partie du manuscrit présente un état de l'art avec un premier chapitre consacré aux méthodes bio-informatique. Nous avons discuté des méthodes d'alignement de séquences, de la modélisation des motifs et domaines, des principales stratégies de découverte de nouveaux motifs, des problématiques de segmentation, et pour finir du modèle de régression linéaire avec une pénalisation LASSO. Tous ces outils sont nécessaires à la compréhension des méthodes développées par la suite. Dans le deuxième chapitre, nous avons présenté le sujet de notre étude de cas : *Plasmodium falciparum*. Nous avons choisi cette organisme en raison de l'enjeu majeur en terme de santé mondiale qu'il représente. De plus, il s'agit d'un organisme dont le fonctionnement biologique semble encore assez obscure du fait des nombreuses différences majeures que l'on a pu observées lorsqu'on le compare aux autres espèces habituellement étudiées. Ce pathogène représente un véritable défi pour beaucoup de méthodes et d'analyses bio-informatiques.

La deuxième partie de cette thèse est consacrée à la découverte de domaines protéiques. Le troisième chapitre présente un état de l'art du fonctionnement et de la structuration des protéines et de domaines protéiques. Ce chapitre présente également quelques bases de données de domaines protéiques et les processus de création de ces bases. On y distingue alors plusieurs cas. Les bases de données construites automatiquement sur la base d'un algorithme de découverte (par exemple Pfam-B) permettent généralement de fournir de très nombreuses annotations, mais comme nous l'avons observé ensuite, la qualité des prédictions ne sont généralement pas satisfaisantes. Il existe également des bases de données construites manuellement en collectant les résultats expérimentaux, comme la PDB par exemple. Les annotations de ces bases sont de grande qualité mais malheu-

reusement souvent peu nombreuses et on ne retrouve généralement pas, ou peu, d'annotations pour *Plasmodium falciparum*. Enfin, on retrouve les bases de données construites à l'aide d'une procédure mixte comme Pfam-A. Cette approche permet de proposer un grand nombre d'annotations tout en contrôlant la qualité des prédictions. Cependant nous avons pu observé que les statistiques d'annotations de Pfam-A sur *Plasmodium falciparum* sont inférieures comparées aux autres espèces modèles. Cette observation a motivé le sujet du chapitre suivant qui porte sur le développement d'une méthode de découverte de nouvelles familles de domaines protéiques à partir des outils de comparaisons de paires de séquences. Dans ce chapitre nous avons développé une heuristique d'analyse des résultats de BLAST qui exploite la propriété de co-occurrence des domaines protéiques. Ce propriété forte nous dit qu'il existe un répertoire limité des combinaisons des domaines protéiques sur une même protéine. Notre procédure parcours donc les résultats de BLAST pour identifier des sous-séquences qui reviennent ensembles sur les mêmes protéines plus fréquemment qu'attendu par hasard. En rassemblant les sous-séquences ainsi identifiées, nous avons montré qu'il est possible de produire de nouvelles familles de domaines protéiques. De plus, nous avons défini quatre mesures pour évaluer la qualité d'un alignement multiple de séquences et nous avons comparé nos nouvelles familles aux familles existantes dans les autres bases de données. Nous avons ainsi observé que nos résultats étaient relativement proches des résultats des familles de la base Pfam-A mais surtout que nos résultats étaient de bien meilleure qualité que les autres bases automatiques comme Pfam-B. Enfin, nous avons proposé plusieurs perspectives à la suite de ce travail et notamment le fait que nous pourrions étendre le principe de co-occurrence à d'autres composantes des protéines comme la présence de séquences répétées en tandem ou encore la co-occurrence de structures secondaires.

La troisième et dernière partie de cette thèse est consacrée à l'étude de la régulation de l'expression des gènes. Dans le cinquième chapitre nous avons présenté un rapide état de l'art des méthodes de prédiction de l'expression. Nous avons notamment présenté les travaux de Bessière *et al.* [BTP+18] portant sur la prédiction de l'expression des gènes humains à l'aide d'un modèle de régression linéaire avec une pénalisation LASSO. Ces auteurs ont mis en avant l'existence de régions ADN spécifiques dont la composition dinucléotidique est informative pour la prédiction de l'expression. Dans ce travail, les auteurs ont utilisé une segmentation manuelle basée sur les connaissances *a priori* de l'architecture des gènes humains et n'ont pas poussé l'étude à l'analyse de compositions plus complexes au delà des dinucléotides. Dans ce chapitre nous avons également pu constater qu'à l'heure actuelle, il n'existe (à notre connaissance) qu'une seule publication scientifique traitant de la prédiction de l'expression chez *Plasmodium falciparum* [RCL+19] et que celle ci n'exploite que très partiellement la séquence mais base plutôt ses prédictions

sur les données expérimentales. Tout ceci nous a donc conduit à développer une méthode pour la découverte de longues régions en lien avec l'expression des gènes dans le sixième et dernier chapitre. Nous avons nommé cette méthode DExTER et l'avons appliquée à différentes espèces. Nous avons observé que les prédictions sont d'une étonnante précision (50 à 70% de corrélation entre les prédictions et les observations) au regard de la simplicité des variables prédictives utilisées. De plus, nous observons que ces longues régions semblent présentes dans toutes les espèces eucaryotes analysées. Nous avons donc proposé l'hypothèse de l'existence de nouveaux mécanismes de régulation au travers de ces longues régions. Ces séquences, que l'on nomme domaines de régulation, semblent partagés à l'échelle des espèces eucaryotes mais leur importance relative varie d'une espèce à l'autre. D'un point de vue méthodologique nous avons montré que notre procédure d'exploration de l'espace des k-mers et des régions est efficace et qu'elle présente la possibilité d'être généralisable pour analyser d'autres types de données qui peuvent s'expliquer à partir de variables composition/région. Pour finir, face aux résultats très encouragent que nous avons obtenus sur *Plasmodium falciparum*, nous avons entrepris de valider expérimentalement certains des domaines identifiés en partenariat avec Jose-Juan Lopez-Rubio et son équipe. Nous attendons maintenant les résultats avec impatience.

Bibliographie

[ABB+00] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene Ontology : tool for the unification of biology. *Nature Genetics*, 25(1) :25–29, May 2000.

[ABB+10] Cristina Aurrecoechea, John Brestelli, Brian P. Brunk, Steve Fischer, Bindu Gajria, Xin Gao, Alan Gingle, Greg Grant, Omar S. Harb, Mark Heiges, Frank Innamorato, John Iodice, Jessica C. Kissinger, Eileen T. Kraemer, Wei Li, John A. Miller, Vishal Nayak, Cary Pennington, Deborah F. Pinney, David S. Roos, Chris Ross, Ganesh Srinivasamoorthy, Christian J. Stoeckert, Ryan Thibodeau, Charles Treatman, and Haiming Wang. EuPathDB : a portal to eukaryotic pathogen databases. *Nucleic Acids Research*, 38(Database issue) :D415–419, January 2010.

[ACK+16] Sophie H. Adjalley, Christophe D. Chabbert, Bernd Klaus, Vicent Pelechano, and Lars M. Steinmetz. Landscape and Dynamics of Transcription Initiation in the Malaria Parasite Plasmodium falciparum. *Cell Reports*, 14(10) :2463–2475, March 2016.

[AE86] Stephen F. Altschul and Bruce W. Erickson. Optimal sequence alignment using affine gap costs. *Bulletin of Mathematical Biology*, 48(5-6) :603–616, September 1986.

[AFS93] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. In *International*

conference on foundations of data organization and algorithms, pages 69–84. Springer, 1993.

[AGM+90] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3) :403–410, October 1990.

[AGT01] Gordana Apic, Julian Gough, and Sarah A Teichmann. Domain combinations in archaeal, eubacterial and eukaryotic proteomes11edited by G. von Heijne. *Journal of Molecular Biology*, 310(2) :311–325, July 2001.

[AHB+04] Antonina Andreeva, Dave Howorth, Steven E. Brenner, Tim J. P. Hubbard, Cyrus Chothia, and Alexey G. Murzin. SCOP database in 2004 : refinements integrate structure and sequence family data. *Nucleic Acids Research*, 32(suppl_1) :D226–D229, January 2004.

[AHT03] Gordana Apic, Wolfgang Huber, and Sarah A. Teichmann. Multi-domain protein families and domain pairs : comparison with known structures and a random model of domain recombination. *Journal of Structural and Functional Genomics*, 4(2-3) :67–78, 2003.

[AK14] Erik A trumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3) :647–665, December 2014.

[ALBL17] Felipe Albrecht, Markus List, Christoph Bock, and Thomas Lengauer. DeepBlueR : large-scale epigenomic analysis in R. *Bioinformatics (Oxford, England)*, 33(13) :2063–2064, July 2017.

[Alt97] S. Altschul. Gapped BLAST and PSI-BLAST : a new generation of protein database search programs. *Nucleic Acids Research*, 25(17) :3389–3402, September 1997.

[AOA06] Sigrid D. Auweter, Florian C. Oberstrass, and Frédéric H.-T. Allain. Sequence-specific binding of single-stranded RNA : is there a code for recognition ? *Nucleic Acids Research*, 34(17) :4943–4959, October 2006.

[AS18] Vikram Agarwal and Jay Shendure. Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks. *bioRxiv*, page 416685, October 2018.

[AW92] Naoki Abe and Manfred K. Warmuth. On the Computational Complexity of Approximating Distributions by Probabilistic Automata. *Machine Learning*, 9(2-3) :205–260, July 1992.

[AWS+19] Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Amr Alexandari, Sabrina Krueger, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, and Julia Zeitlinger. Deep learning at base-resolution reveals motif syntax of the cis-regulatory code. *bioRxiv*, page 737981, August 2019.

[Bau] Leonard E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. In Oved Shisha, editor, *Inequalities III : Proceedings of the Third Symposium on Inequalities*, pages 1–8. Academic Press.

[BBA13] Erich Bornberg-Bauer and M. Mar Albà. Dynamics and adaptive benefits of modular protein evolution. *Current Opinion in Structural Biology*, 23(3) :459–466, June 2013.

[BBBK+05] E. Bornberg-Bauer, F. Beaussart, S. K. Kummerfeld, S. A. Teichmann, and J. Weiner. The evolution of domain arrangements in proteins and interaction networks. *Cellular and molecular life sciences : CMLS*, 62(4) :435–445, February 2005.

[BBC+03] Amit Bahl, Brian Brunk, Jonathan Crabtree, Martin J. Fraunholz, Bindu Gajria, Gregory R. Grant, Hagai Ginsburg, Dinesh Gupta, Jessica C. Kissinger, Philip Labo, Li Li, Matthew D. Mailman, Arthur J. Milgram, David S. Pearson, David S. Roos, Jonathan Schug, Christian J. Stoeckert, and Patricia Whetzel. PlasmoDB : the Plasmodium genome resource. A database integrating experimental and computational data. *Nucleic Acids Research*, 31(1) :212–215, January 2003.

[BBIA05] S. Balaji, M. Madan Babu, Lakshminarayan M. Iyer, and L. Aravind. Discovery of the principal specific transcription factors of Apicomplexa and their implication for the evolution of the AP2-integrase DNA binding domains. *Nucleic Acids Research*, 33(13) :3994–4006, 2005.

[BCC+05] Catherine Bru, Emmanuel Courcelle, Sébastien Carrère, Yoann Beausse, Sandrine Dalmar, and Daniel Kahn. The ProDom database of protein domain families : more emphasis on 3d. *Nucleic Acids Research*, 33(Database issue) :D212–215, January 2005.

[BCH08] J. Benesty, J. Chen, and Y. Huang. On the Importance of the Pearson Correlation Coefficient in Noise Reduction. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(4) :757–765, May 2008.

[BD93] E. Brodsky and B. S. Darkhovsky. *Nonparametric Methods in Change Point Problems*. Mathematics and Its Applications. Springer Netherlands, 1993.

[BE] Leonard E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. 73(3) :360–363.

[BE95] Timothy L Bailey and Charles Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine learning*, 21(1-2) :51–80, 1995.

[BFHBBC15] Tristan Bitard-Feildel, Magdalena Heberlein, Erich Bornberg-Bauer, and Isabelle Callebaut. Detection of orphan domains in Drosophila using "hydrophobic cluster analysis". *Biochimie*, March 2015.

[BHK+93] Michael Brown, Richard Hughey, Anders Krogh, I. Saira Mian, Kimmen Sjölander, and David Haussler. (1) Using Dirichlet Mixture Priors to Derive Hidden Markov Models for Protein Families. *AAAI Press*, pages 47–55, 1993.

[BHN03] Helen Berman, Kim Henrick, and Haruki Nakamura. Announcing the worldwide Protein Data Bank. *Nature Structural Biology*, 10(12) :980, December 2003.

[BHSA+10] Richárd Bártfai, Wieteke A. M. Hoeijmakers, Adriana M. Salcedo-Amaya, Arne H. Smits, Eva Janssen-Megens, Anita Kaan, Moritz Treeck, Tim-Wolf Gilberger, Kees-Jan Françoijs, and Hendrik G. Stunnenberg. H2a.Z demarcates intergenic regions of the plasmodium falciparum epigenome that are dynamically marked by H3k9ac and H3k4me3. *PLoS pathogens*, 6(12) :e1001223, December 2010.

[Bir87] Adrian P. Bird. CpG islands as gene markers in the vertebrate nucleus. *Trends in Genetics*, 3 :342–347, January 1987.

[BJVU98] A. Brazma, I. Jonassen, J. Vilo, and E. Ukkonen. Predicting gene regulatory elements in silico on a genomic scale. *Genome Research*, 8(11) :1202–1215, November 1998.

[BLP+03] Zbynek Bozdech, Manuel Llinas, Brian Lee Pulliam, Edith D. Wong, Jingchun Zhu, and Joseph L. DeRisi. The Transcriptome of the Intraerythrocytic Developmental Cycle of Plasmodium falciparum. *PLOS Biology*, 1(1) :e5, August 2003.

[BLR+04] Olivier Bastien, Sylvain Lespinats, Sylvaine Roy, Karine Métayer, Bernard Fertil, Jean-Jacques Codani, and Eric Maréchal. Analysis of the compositional biases in Plasmodium falciparum genome and proteome using Arabidopsis thaliana as a reference. *Gene*, 336(2) :163–173, July 2004.

[BLS01] H. J. Bussemaker, H. Li, and E. D. Siggia. Regulatory element detection using correlation with expression. *Nature Genetics*, 27(2) :167–171, February 2001.

[BMS77] Susan M Berget, Claire Moore, and Phillip A Sharp. Spliced segments at the 5' terminus of adenovirus 2 late mRNA. *Proceedings of the National Academy of Sciences*, 74(8) :3171–3175, 1977.

[BN93] Michèle Basseville and Igor V. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. Prentice Hall, Inc. - http ://people.irisa.fr/Michele.Basseville/kniga/, 1993.

[BP] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. 37(6) :1554–1563.

[BPSW] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. 41(1) :164–171.

[BS] Leonard E. Baum and George Sell. Growth transformations for functions on manifolds. 27(2) :211–227.

[BSA+12] Grzegorz M Boratyn, Alejandro A Schäffer, Richa Agarwala, Stephen F Altschul, David J Lipman, and Thomas L Madden. Domain enhanced lookup time accelerated BLAST. *Biology Direct*, 7(1) :12, 2012.

[BTP+18] Chloé Bessière, May Taha, Florent Petitprez, Jimmy Vandel, Jean-Michel Marin, Laurent Bréhélin, Sophie Lèbre, and Charles-Henri Lecellier. Probing instructions for expression regulation in gene nucleotide compositions. *PLOS Computational Biology*, 14(1) :e1005921, January 2018.

[Bur14] Fabien Burki. The Eukaryotic Tree of Life from a Global Phylogenomic Perspective. *Cold Spring Harbor Perspectives in Biology*, 6(5) :a016147, January 2014.

[BVZC16] J.S. Bernardes, F.R.J. Vieira, G. Zaverucha, and A. Carbone. A multi-objective optimization approach accurately resolves protein domain architectures. *Bioinformatics*, 32(3) :345–353, February 2016.

[BW94] Michael Burrows and David J Wheeler. A block-sorting lossless data compression algorithm. 1994.

[BZVC16] Juliana Bernardes, Gerson Zaverucha, Catherine Vaquero, and Alessandra Carbone. Improvement in Protein Domain Identification Is Reached by Breaking Consensus, with the Agreement of Many Profiles and Domain Co-occurrence. *PLoS computational biology*, 12(7) :e1005038, 2016.

[Cas90] F. Casacuberta. Some Relations Among Stochastic Finite State Networks Used in Automatic Speech Recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 12(7) :691–695, 1990.

[CF99] Kin-Pong Chan and Wai-Chee Fu. Efficient time series matching by wavelets. In *icde*, page 126. IEEE, 1999.

[CG14] Jie Chen and Arjun K. Gupta. *Parametric statistical change point analysis : With applications to genetics, medicine, and finance*. Birkhauser Boston, January 2014.

[CGBR77] L. T. Chow, R. E. Gelinas, T. R. Broker, and R. J. Roberts. An amazing sequence arrangement at the 5' ends of adenovirus 2 messenger RNA. *Cell*, 12(1) :1–8, September 1977.

[CGKC11] Pimwadee Chaovalit, Aryya Gangopadhyay, George Karabatis, and Zhiyuan Chen. Discrete wavelet transform-based time series analysis and mining. *ACM Computing Surveys (CSUR)*, 43(2) :6, 2011.

[CHO04] Richard M. R. Coulson, Neil Hall, and Christos A. Ouzounis. Comparative Genomics of Transcriptional Control in the Human Malaria Parasite Plasmodium falciparum. *Genome Research*, 14(8) :1548–1554, January 2004.

[CLN+16] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. Gene expression inference with deep learning. *Bioinformatics (Oxford, England)*, 32(12) :1832–1839, 2016.

[CPM+05] Isabelle Callebaut, Karine Prat, Edwige Meurice, Jean-Paul Mornon, and Stanislas Tomavo. Prediction of the general transcription factors associated with RNA polymerase II in Plasmodium falciparum : conserved features and differences relative to other eukaryotes. *BMC genomics*, 6 :100, July 2005.

[CPR+17] Junyue Cao, Jonathan S. Packer, Vijay Ramani, Darren A. Cusanovich, Chau Huynh, Riza Daza, Xiaojie Qiu, Choli Lee, Scott N. Furlan, Frank J. Steemers, Andrew Adey, Robert H. Waterston, Cole Trapnell, and Jay Shendure. Comprehensive single cell transcriptional profiling of a multicellular organism. *Science (New York, N.Y.)*, 357(6352) :661–667, August 2017.

[CYY+11] Chao Cheng, Koon-Kiu Yan, Kevin Y. Yip, Joel Rozowsky, Roger Alexander, Chong Shou, and Mark Gerstein. A statistical framework for modeling gene expression using chromatin features and application to modENCODE datasets. *Genome Biology*, 12(2) :R15, 2011.

[DB11] Aimée M. Deaton and Adrian Bird. CpG islands and the regulation of transcription. *Genes & Development*, 25(10) :1010–1022, May 2011.

[DCJ+17] Zhana Duren, Xi Chen, Rui Jiang, Yong Wang, and Wing Hung Wong. Modeling gene regulation from paired expression and chromatin accessibility data. *Proceedings of the National Academy of Sciences of the United States of America*, 114(25) :E4914–E4923, 2017.

[DE77] M.O. Dayhoff and R.V. Eck. A Model of Evolutionary Change in Proteins. *Atlas Protein Seq. Struct.*, 5, 1977.

[DGK+12] Xianjun Dong, Melissa C. Greven, Anshul Kundaje, Sarah Djebali, James B. Brown, Chao Cheng, Thomas R. Gingeras, Mark Gerstein, Roderic Guigó, Ewan Birney, and Zhiping Weng. Modeling gene expression using chromatin features in various cellular contexts. *Genome Biology*, 13(9) :R53, September 2012.

[DHS+18] Carrie A. Davis, Benjamin C. Hitz, Cricket A. Sloan, Esther T. Chan, Jean M. Davidson, Idan Gabdank, Jason A. Hilton, Kriti Jain, Ulugbek K. Baymuradov, Aditi K. Narayanan, Kathrina C. Onate, Keenan Graham, Stuart R. Miyasato, Timothy R. Dreszer, J. Seth Strattan, Otto Jolanki, Forrest Y. Tanaka, and J. Michael Cherry. The Encyclopedia of DNA elements (ENCODE) : data portal update. *Nucleic Acids Research*, 46(D1) :D794–D801, 2018.

[Dil85] Ken A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24(6) :1501–1509, March 1985.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1) :1–38, 1977.

[DMS14] Randal Douc, Eric Moulines, and David Stoffer. *Nonlinear time series : theory, methods and applications with R examples*. Texts in statistical science. Chapman et Hall - CRC Press, 2014.

[DNMO17] Heledd M. Davies, Stephanie D. Nofal, Emilia J. McLaughlin, and Andrew R. Osborne. Repetitive sequences in malaria parasite proteins. *FEMS microbiology reviews*, 41(6) :923–940, 2017.

[DP73] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica : The International Journal for Geographic Information and Geovisualization*, 10(2) :112–122, 1973.

[DREKJM98] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme J. Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*, volume 3. January 1998.

[DS06] Shailesh V. Date and Christian J. Stoeckert. Computational modeling of the Plasmodium falciparum interactome reveals protein function on a genome-wide scale. *Genome Research*, 16(4) :542–549, January 2006.

[Dud73] Richard O. Duda. *Pattern classification and scene analysis*. Wiley, New York, 1973. Open Library ID : OL5287711M.

[Edd95] S. R. Eddy. Multiple alignment using hidden Markov models. *Proceedings. International Conference on Intelligent Systems for Molecular Biology*, 3 :114–120, 1995.

[Edd98] S. R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9) :755–763, October 1998.

[Edd03] Sean Eddy. HMMER User's Guide. Biological Sequence Analysis Using Profile Hidden Markov Models. 2003.

[Edd04] Sean R. Eddy. What is dynamic programming ? *Nature Biotechnology*, 22(7) :909–910, July 2004.

[Edg04a] Robert C. Edgar. MUSCLE : a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, 5 :113, August 2004.

[Edg04b] Robert C. Edgar. MUSCLE : multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5) :1792–1797, 2004.

[EGMB+19] Sara El-Gebali, Jaina Mistry, Alex Bateman, Sean R. Eddy, Aurélien Luciani, Simon C. Potter, Matloob Qureshi, Lorna J. Richardson, Gustavo A. Salazar, Alfredo Smart, Erik L. L. Sonnhammer, Layla Hirsh, Lisanna Paladin, Damiano Piovesan, Silvio C. E. Tosatto, and Robert D. Finn. The Pfam protein families database in 2019. *Nucleic Acids Research*, 47(D1) :D427–D432, January 2019.

[EHJ+04] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, and others. Least angle regression. *The Annals of statistics*, 32(2) :407–499, 2004.

[FCE+16] Robert D. Finn, Penelope Coggill, Ruth Y. Eberhardt, Sean R. Eddy, Jaina Mistry, Alex L. Mitchell, Simon C. Potter, Marco Punta, Matloob Qureshi, Amaia Sangrador-Vegas, Gustavo A.

Salazar, John Tate, and Alex Bateman. The Pfam protein families database : towards a more sustainable future. *Nucleic Acids Research*, 44(D1) :D279–D285, January 2016.

[Fer99] Alan Fersht. *Structure and mechanism in protein science : a guide to enzyme catalysis and protein folding*. Macmillan, 1999.

[FM00] Paolo Ferragina and Giovanni Manzini. Opportunistic data structures with applications. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 390–398. IEEE, 2000.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[GE03] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar) :1157–1182, 2003.

[GFG+14] Amel Ghouila, Isabelle Florent, Fatma Zahra Guerfali, Nicolas Terrapon, Dhafer Laouini, and Sadok Ben Yahia. Identification of divergent protein domains by combining HMM-HMM comparisons and co-occurrence detection. *PloS One*, 9(6) :e95275, 2014.

[GGF87] M. Gardiner-Garden and M. Frommer. CpG Islands in vertebrate genomes. *Journal of Molecular Biology*, 196(2) :261–282, July 1987.

[GGG10] M. Gouy, S. Guindon, and O. Gascuel. SeaView Version 4 : A Multiplatform Graphical User Interface for Sequence Alignment and Phylogenetic Tree Building. *Molecular Biology and Evolution*, 27(2) :221–224, February 2010.

[GHF+02] Malcolm J. Gardner, Neil Hall, Eula Fung, Owen White, Matthew Berriman, Richard W. Hyman, Jane M. Carlton, Arnab Pain, Karen E. Nelson, Sharen Bowman, Ian T. Paulsen, Keith James, Jonathan A. Eisen, Kim Rutherford, Steven L. Salzberg, Alister Craig, Sue Kyes, Man-Suen Chan, Vishvanath Nene, Shamira J. Shallom, Bernard Suh, Jeremy Peterson, Sam Angiuoli, Mihaela Pertea, Jonathan Allen, Jeremy Selengut, Daniel Haft, Michael W. Mather, Akhil B. Vaidya, David M. A. Martin, Alan H. Fairlamb, Martin J. Fraunholz, David S. Roos,

Stuart A. Ralph, Geoffrey I. McFadden, Leda M. Cummings, G. Mani Subramanian, Chris Mungall, J. Craig Venter, Daniel J. Carucci, Stephen L. Hoffman, Chris Newbold, Ronald W. Davis, Claire M. Fraser, and Bart Barrell. Genome sequence of the human malaria parasite Plasmodium falciparum. *Nature*, 419(6906) :498–511, October 2002.

[GHT14] Stefanie Gerstberger, Markus Hafner, and Thomas Tuschl. A census of human RNA-binding proteins. *Nature Reviews Genetics*, 15(12) :829–845, December 2014.

[GKG+09] Deepti Gangwar, Mridul K. Kalita, Dinesh Gupta, Virander S. Chauhan, and Asif Mohmmed. A systematic classification of Plasmodium falciparum P-loop NTPases : structural and functional correlation. *Malaria Journal*, 8(1) :69, April 2009.

[GS01] Xianping Ge and Padhraic Smyth. Segmental semi-markov models for endpoint detection in plasma etching. *IEEE Transactions on Semiconductor Engineering*, 259 :201–209, 2001.

[GTE13] GTEx Consortium. The Genotype-Tissue Expression (GTEx) project. *Nature Genetics*, 45(6) :580–585, June 2013.

[HBB+06] Nicolas Hulo, Amos Bairoch, Virginie Bulliard, Lorenzo Cerutti, Edouard De Castro, Petra S. Langendijk-Genevaux, Marco Pagni, and Christian J. A. Sigrist. The PROSITE database. *Nucleic Acids Research*, 34(Database issue) :D227–230, January 2006.

[HBF92] D. G. Higgins, A. J. Bleasby, and R. Fuchs. CLUSTAL V : improved software for multiple sequence alignment. *Computer applications in the biosciences : CABIOS*, 8(2) :189–191, April 1992.

[HCK+10] Guangan Hu, Ana Cabrera, Maya Kono, Sachel Mok, Balbir K. Chaal, Silvia Haase, Klemens Engelberg, Sabna Cheemadan, Tobias Spielmann, Peter R. Preiser, Tim-W. Gilberger, and Zbynek Bozdech. Transcriptional profiling of growth perturbations of the human malaria parasite *Plasmodium falciparum*. *Nature Biotechnology*, 28(1) :91–98, January 2010.

[Hei89] J. Hein. A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous

sequences, when the phylogeny is given. *Molecular Biology and Evolution*, 6(6) :649–668, November 1989.

[Her93] N. Hernandez. TBP, a universal eukaryotic transcription factor ? *Genes & Development*, 7(7B) :1291–1308, July 1993.

[HG97] Paul S Heckbert and Michael Garland. Survey of polygonal surface simplification algorithms. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1997.

[HG01] Hedi Hegyi and Mark Gerstein. Annotation Transfer for Genomics : Measuring Functional Divergence in Multi-Domain Proteins. *Genome Research*, 11(10) :1632–1640, October 2001.

[HH92] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22) :10915–10919, November 1992.

[HH94] S. Henikoff and J. G. Henikoff. Protein family classification based on searching a database of blocks. *Genomics*, 19(1) :97–107, January 1994.

[HH03] Andreas Heger and Liisa Holm. Exhaustive Enumeration of Protein Domain Families. *Journal of Molecular Biology*, 328(3) :749–767, May 2003.

[HK70] Arthur E. Hoerl and Robert W. Kennard. Ridge Regression : Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1) :55–67, February 1970.

[HK11] Jan Hauke and Tomasz Kossowski. Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data. *Quaestiones geographicae*, 30(2) :87–93, 2011.

[HKMS93] D. Haussler, A. Krogh, I. S. Mian, and K. Sjolander. Protein modeling using hidden Markov models : analysis of globins. In *[1993] Proceedings of the Twenty-sixth Hawaii International Conference on System Sciences*, volume i, pages 792–802 vol.1, January 1993.

[HM99] Jim Hunter and Neil McIntosh. Knowledge-Based Event Detection in Complex Time Series Data. In Werner Horn, Yuval Shahar, Greger Lindberg, Steen Andreassen, and Jeremy Wyatt, editors, *Artificial Intelligence in Medicine*, pages 271–280, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[HSAS+13] Wieteke A. M. Hoeijmakers, Adriana M. Salcedo-Amaya, Arne H. Smits, Kees-Jan Françoijs, Moritz Treeck, Tim-Wolf Gilberger, Hendrik G. Stunnenberg, and Richárd Bártfai. H2a.Z/H2b.Z double-variant nucleosomes inhabit the AT-rich promoter regions of the Plasmodium falciparum genome. *Molecular Microbiology*, 87(5) :1061–1073, March 2013.

[HSH+18] Yi-Hsuan Ho, Evgenia Shishkova, James Hose, Joshua J. Coon, and Audrey P. Gasch. Decoupling yeast cell division and stress defense implicates mRNA repression in translational reallocation during stress. *Current biology : CB*, 28(16) :2673–2680.e4, August 2018.

[HTHI95] Makoto Hirosawa, Yasushi Totoki, Masaki Hoshida, and Masato Ishikawa. Comprehensive study on iterative algorithms of multiple sequence alignment. *Bioinformatics*, 11(1) :13–18, February 1995.

[ISHS83] M. Ishijima, S. B. Shin, G. H. Hostetter, and J. Sklansky. Scan-along polygonal approximation for data compression of electrocardiograms. *IEEE transactions on bio-medical engineering*, 30(11) :723–729, November 1983.

[JHP+18] Christoph Jansen, Stephan Hodel, Thomas Penzel, Martin Spott, and Dagmar Krefting. Feature relevance in physiological networks for classification of obstructive sleep apnea. *Physiological Measurement*, 39(12) :124003, 2018.

[KA90] S. Karlin and S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences of the United States of America*, 87(6) :2264–2268, March 1990.

[Kar93] Richard M. Karp. Mapping the Genome : Some Combinatorial Problems Arising in Molecular Biology. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 278–285, New York, NY, USA, 1993. ACM.

[KBD+05] Patrick J. Keeling, Gertraud Burger, Dion G. Durnford, B. Franz Lang, Robert W. Lee, Ronald E. Pearlman, Andrew J.

Roger, and Michael W. Gray. The tree of eukaryotes. *Trends in Ecology & Evolution*, 20(12) :670–676, December 2005.

[KBM+94] Anders Krogh, Michael Brown, I. Saira Mian, Kimmen Sjölander, and David Haussler. Hidden Markov Models in Computational Biology : Applications to Protein Modeling. *Journal of Molecular Biology*, 235(5) :1501–1531, February 1994.

[KCPM01] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowledge and Information Systems*, 3(3) :263–286, August 2001.

[Kim83] Motoo Kimura. *The Neutral Theory of Molecular Evolution*. Cambridge University Press, 1983. Google-Books-ID : olIoSumPevYC.

[KJM95] Antti Koski, Martti Juhola, and Merik Meriste. Syntactic recognition of ECG signals by attributed finite automata. *Pattern Recognition*, 28(12) :1927 – 1940, 1995.

[KNHK91] Bernhard Knapp, Uwe Nau, Erika Hundt, and HA Kupper. Demonstration of alternative splicing of a pre-mRNA expressed in the blood stage form of Plasmodium falciparum. *Journal of Biological Chemistry*, 266(11) :7148–7154, 1991.

[KP98] Eamonn J Keogh and Michael J Pazzani. An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback. In *Kdd*, volume 98, pages 239–243, 1998.

[KP99] Eamonn J. Keogh and Michael J. Pazzani. Relevance Feedback Retrieval of Time Series Data. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 183–190, New York, NY, USA, 1999. ACM. event-place : Berkeley, California, USA.

[KRB+18] David R. Kelley, Yakir A. Reshef, Maxwell Bileschi, David Belanger, Cory Y. McLean, and Jasper Snoek. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Research*, 28(5) :739–750, January 2018.

[KS97] Eamonn J Keogh and Padhraic Smyth. A probabilistic approach to fast pattern matching in time series databases. In *Kdd*, volume 1997, pages 24–30, 1997.

[KSR16] David R. Kelley, Jasper Snoek, and John L. Rinn. Basset : learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research*, 26(7) :990–999, 2016.

[LAB+93] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals : a Gibbs sampling strategy for multiple alignment. *Science*, 262(5131) :208–214, October 1993.

[LAC+16] Michal Levin, Leon Anavy, Alison G. Cole, Eitan Winter, Natalia Mostov, Sally Khair, Naftalie Senderovich, Ekaterina Kovalev, David H. Silver, Martin Feder, Selene L. Fernandez-Valverde, Nagayasu Nakanishi, David Simmons, Oleg Simakov, Tomas Larsson, Shang-Yun Liu, Ayelet Jerafi-Vider, Karina Yaniv, Joseph F. Ryan, Mark Q. Martindale, Jochen C. Rink, Detlev Arendt, Sandie M. Degnan, Bernard M. Degnan, Tamar Hashimshony, and Itai Yanai. The mid-developmental transition and the evolution of animal body plans. *Nature*, 531(7596) :637–641, March 2016.

[Lan02] Kenneth Lange. *Mathematical and Statistical Methods for Genetic Analysis*. Statistics for Biology and Health. Springer-Verlag, New York, 2 edition, 2002.

[LB98] Alexander V. Lukashin and Mark Borodovsky. GeneMark.hmm : New solutions for gene finding. *Nucleic Acids Research*, 26(4) :1107–1115, February 1998.

[LBLQ+11] María J. López-Barragán, Jacob Lemieux, Mariam Quiñones, Kim C. Williamson, Alvaro Molina-Cruz, Kairong Cui, Carolina Barillas-Mury, Keji Zhao, and Xin-zhuan Su. Directional gene expression and antisense transcripts in sexual and asexual stages of Plasmodium falciparum. *BMC genomics*, 12 :587, November 2011.

[LBS+07] Liana F. Lareau, Angela N. Brooks, David A. W. Soergel, Qi Meng, and Steven E. Brenner. The coupling of alternative splicing and nonsense-mediated mRNA decay. *Advances in Experimental Medicine and Biology*, 623 :190–211, 2007.

[LFLN18] Yang Lu, Yingying Fan, Jinchi Lv, and William Stafford Noble. DeepPINK : reproducible feature selection in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 8676–8686, 2018.

[LHS+15] Marina Lizio, Jayson Harshbarger, Hisashi Shimoji, Jessica Severin, Takeya Kasukawa, Serkan Sahin, Imad Abugessaisa, Shiro Fukuda, Fumi Hori, Sachi Ishikawa-Kato, Christopher J. Mungall, Erik Arner, J. Kenneth Baillie, Nicolas Bertin, Hidemasa Bono, Michiel de Hoon, Alexander D. Diehl, Emmanuel Dimont, Tom C. Freeman, Kaori Fujieda, Winston Hide, Rajaram Kaliyaperumal, Toshiaki Katayama, Timo Lassmann, Terrence F. Meehan, Koro Nishikata, Hiromasa Ono, Michael Rehli, Albin Sandelin, Erik A. Schultes, Peter AC 't Hoen, Zuotian Tatum, Mark Thompson, Tetsuro Toyoda, Derek W. Wright, Carsten O. Daub, Masayoshi Itoh, Piero Carninci, Yoshihide Hayashizaki, Alistair RR Forrest, Hideya Kawaji, and the FANTOM consortium. Gateways to the FANTOM5 promoter level mammalian expression atlas. *Genome Biology*, 16(1) :22, January 2015.

[LJC+18] Samuel A. Lambert, Arttu Jolma, Laura F. Campitelli, Pratyush K. Das, Yimeng Yin, Mihai Albu, Xiaoting Chen, Jussi Taipale, Timothy R. Hughes, and Matthew T. Weirauch. The Human Transcription Factors. *Cell*, 172(4) :650–665, February 2018.

[LJX+12] Jun Liu, Choonkyun Jung, Jun Xu, Huan Wang, Shulin Deng, Lucia Bernad, Catalina Arenas-Huertero, and Nam-Hai Chua. Genome-Wide Analysis Uncovers Regulation of Long Intergenic Noncoding RNAs in Arabidopsis[C][W]. *The Plant Cell*, 24(11) :4333–4345, November 2012.

[LLS91] Eric S. Lander, Robert Langridge, and Damian M. Saccocio. Mapping and Interpreting Biological Information. *Commun. ACM*, 34(11) :32–39, November 1991.

[LLZ14] Yue Li, Minggao Liang, and Zhaolei Zhang. Regression analysis of combined gene expression regulation in acute myeloid leukemia. *PLoS computational biology*, 10(10) :e1003908, October 2014.

[LPB+16] Catherine L. Lawson, Ardan Patwardhan, Matthew L. Baker, Corey Hryc, Eduardo Sanz Garcia, Brian P. Hudson, Ingvar Lagerstedt, Steven J. Ludtke, Grigore Pintilie, Raul Sala, John D. Westbrook, Helen M. Berman, Gerard J. Kleywegt, and Wah Chiu. EMDataBank unified data resource for 3dem. *Nucleic Acids Research*, 44(D1) :D396–D403, January 2016.

[LQLM10] Xiao Li, Gerald Quon, Howard D. Lipshitz, and Quaid Morris. Predicting in vivo binding sites of RNA-binding proteins using mRNA secondary structure. *RNA*, 16(6) :1096–1107, June 2010.

[LRJF+04] Karine G. Le Le Roch, Jeffrey R. Johnson, Laurence Florens, Yingyao Zhou, Andrey Santrosyan, Munira Grainger, S. Frank Yan, Kim C. Williamson, Anthony A. Holder, Daniel J. Carucci, John R. Yates, and Elizabeth A. Winzeler. Global analysis of transcript and protein levels across the Plasmodium falciparum life cycle. *Genome Research*, 14(11) :2308–2318, January 2004.

[LRZB+03] Karine G. Le Roch, Yingyao Zhou, Peter L. Blair, Muni Grainger, J. Kathleen Moch, J. David Haynes, Patricia De La Vega, Anthony A. Holder, Serge Batalov, Daniel J. Carucci, and Elizabeth A. Winzeler. Discovery of gene function by expression profiling of the malaria parasite life cycle. *Science (New York, N.Y.)*, 301(5639) :1503–1508, September 2003.

[LSC12] Boris Lenhard, Albin Sandelin, and Piero Carninci. Metazoan promoters : emerging characteristics and insights into transcriptional regulation. *Nature Reviews. Genetics*, 13(4) :233–245, April 2012.

[LSL+00] Victor Lavrenko, Matt Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan. Mining of concurrent text and time series. In *KDD-2000 Workshop on Text Mining*, volume 2000, pages 37–44, 2000.

[LT00] Bryan Lemon and Robert Tjian. Orchestrated response : a symphony of transcription factors for gene control. *Genes & Development*, 14(20) :2551–2569, October 2000.

[LVC+05] Douglas J. LaCount, Marissa Vignali, Rakesh Chettier, Amit Phansalkar, Russell Bell, Jay R. Hesselberth, Lori W. Schoenfeld, Irene Ota, Sudhir Sahasrabudhe, Cornelia Kurschner,

Stanley Fields, and Robert E. Hughes. A protein interaction network of the malaria parasite *Plasmodium falciparum*. *Nature*, 438(7064) :103–107, November 2005.

[LYC98] Chung-Sheng Li, Philip S Yu, and Vittorio Castelli. MALM : A framework for mining sequence database at multiple abstraction levels. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 267–272. ACM, 1998.

[MA16] Azam Moosavi and Ali Motevalizadeh Ardekani. Role of Epigenetics in Biology and Human Diseases. *Iranian Biomedical Journal*, 20(5) :246–258, November 2016.

[MAB+19] Alex L. Mitchell, Teresa K. Attwood, Patricia C. Babbitt, Matthias Blum, Peer Bork, Alan Bridge, Shoshana D. Brown, Hsin-Yu Chang, Sara El-Gebali, Matthew I. Fraser, Julian Gough, David R. Haft, Hongzhan Huang, Ivica Letunic, Rodrigo Lopez, AurĂ©lien Luciani, Fabio Madeira, Aron Marchler-Bauer, Huaiyu Mi, Darren A. Natale, Marco Necci, Gift Nuka, Christine Orengo, Arun P. Pandurangan, Typhaine Paysan-Lafosse, Sebastien Pesseat, Simon C. Potter, Matloob A. Qureshi, Neil D. Rawlings, Nicole Redaschi, Lorna J. Richardson, Catherine Rivoire, Gustavo A. Salazar, Amaia Sangrador-Vegas, Christian J. A. Sigrist, Ian Sillitoe, Granger G. Sutton, Narmada Thanki, Paul D. Thomas, Silvio C. E. Tosatto, Siew-Yit Yong, and Robert D. Finn. InterPro in 2019 : improving coverage, classification and access to protein sequence annotations. *Nucleic Acids Research*, 47(D1) :D351–D360, January 2019.

[MFA+16] Anthony Mathelier, Oriol Fornes, David J. Arenillas, Chih-Yu Chen, Grégoire Denay, Jessica Lee, Wenqiang Shi, Casper Shyr, Ge Tan, Rebecca Worsley-Hunt, Allen W. Zhang, François Parcy, Boris Lenhard, Albin Sandelin, and Wyeth W. Wasserman. JASPAR 2016 : a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 44(D1) :D110–115, January 2016.

[MFC+06] Jun Miao, Qi Fan, Long Cui, Junsuo Li, Jianyong Li, and Liwang Cui. The malaria parasite Plasmodium falciparum histones : organization, expression, and acetylation. *Gene*, 369 :53–65, March 2006.

[MGB18] Christophe Menichelli, Olivier Gascuel, and Laurent Bréhélin. Improving pairwise comparison of protein sequences with domain co-occurrence. *PLOS Computational Biology*, 14(1) :e1005889, January 2018.

[MHKK14] Ann-Kristin Mueller, Christiane Hammerschmidt-Kamper, and Annette Kaiser. RNAi in Plasmodium. *Current Pharmaceutical Design*, 20(2) :278–283, 2014.

[MLY17] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. Deep learning in bioinformatics. *Briefings in Bioinformatics*, 18(5) :851–869, 2017.

[MMR18] D. Martin, V. Maillol, and E. Rivals. Fast and Accurate Genome-Scale Identification of DNA-Binding Sites. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 201–205, December 2018.

[MWR+08] Fabio Mohn, Michael Weber, Michael Rebhan, Tim C. Roloff, Jens Richter, Michael B. Stadler, Miriam Bibel, and Dirk Schubeler. Lineage-Specific Polycomb Targets and De Novo DNA Methylation Define Restriction and Potential of Neuronal Progenitors. *Molecular Cell*, 30(6) :755–766, June 2008.

[NCMCM+18] Nga Thi Thuy Nguyen, Bruno Contreras-Moreira, Jaime A. Castro-Mondragon, Walter Santana-Garcia, Raul Ossio, Carla Daniela Robles-Espinoza, Mathieu Bahin, Samuel Collombet, Pierre Vincens, Denis Thieffry, Jacques van Helden, Alejandra Medina-Rivera, and Morgane Thomas-Chollier. RSAT 2018 : regulatory sequence analysis tools 20th anniversary. *Nucleic Acids Research*, 46(W1) :W209–W214, July 2018.

[NHH00] Cédric Notredame, Desmond G Higgins, and Jaap Heringa. T-coffee : a novel method for fast and accurate multiple sequence alignment11edited by J. Thornton. *Journal of Molecular Biology*, 302(1) :205–217, September 2000.

[NW70] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3) :443–453, March 1970.

[OBJ+14] Thomas D Otto, Ulrike Böhme, Andrew P Jackson, Martin Hunt, Blandine Franke-Fayard, Wieteke A M Hoeijmakers,

Agnieszka A Religa, Lauren Robertson, Mandy Sanders, Solabomi A Ogun, Deirdre Cunningham, Annette Erhart, Oliver Billker, Shahid M Khan, Hendrik G Stunnenberg, Jean Langhorne, Anthony A Holder, Andrew P Waters, Chris I Newbold, Arnab Pain, Matthew Berriman, and Chris J Janse. A comprehensive evaluation of rodent malaria parasite genomes and gene expression. *BMC Biology*, 12, October 2014.

[OGC+18] Thomas D. Otto, Aude Gilabert, Thomas Crellen, Ulrike Böhme, Céline Arnathau, Mandy Sanders, Samuel O. Oyola, Alain Prince Okouga, Larson Boundenga, Eric Willaume, Barthélémy Ngoubangoye, Nancy Diamella Moukodoum, Christophe Paupy, Patrick Durand, Virginie Rougeron, Benjamin Ollomo, François Renaud, Chris Newbold, Matthew Berriman, and Franck Prugnolle. Genomes of all known members of a Plasmodium subgenus reveal paths to virulent human malaria. *Nature Microbiology*, 3(6) :687–697, June 2018.

[OLS11] Alejandro Ochoa, Manuel Llinás, and Mona Singh. Using context to improve protein domain identification. *BMC bioinformatics*, 12 :90, 2011.

[Ope18] CNX OpenStax. *OpenStax, biology.* 2018.

[OS17] Alejandro Ochoa and Mona Singh. Domain prediction with probabilistic directional context. *Bioinformatics*, 33(16) :2471–2478, August 2017.

[OT05] Christine A. Orengo and Janet M. Thornton. Protein families and their evolution-a structural perspective. *Annual Review of Biochemistry*, 74 :867–900, 2005.

[OWA+10] Thomas D. Otto, Daniel Wilinski, Sammy Assefa, Thomas M. Keane, Louis R. Sarry, Ulrike Böhme, Jacob Lemieux, Bart Barrell, Arnab Pain, Matthew Berriman, Chris Newbold, and Manuel Llinás. New insights into the blood-stage transcriptome of Plasmodium falciparum using RNA-Seq. *Molecular Microbiology*, 76(1) :12–24, 2010.

[PB15] Ananth Prakash and Alex Bateman. Domain atrophy creates rare cases of functional partial protein domains. *Genome Biology*, 16 :88, April 2015.

[PBD+13] Len A. Pennacchio, Wendy Bickmore, Ann Dean, Marcelo A. Nobrega, and Gill Bejerano. Enhancers : five essential questions. *Nature reviews. Genetics*, 14(4) :288–295, April 2013.

[PBJ+98] Tomasino Pace, Cecilia Birago, Chris J Janse, Leonardo Picci, and Marta Ponzi. Developmental regulation of a Plasmodium gene involves the generation of stage-specific 5' untranslated sequences. *Molecular and biochemical parasitology*, 97(1-2) :45–53, 1998.

[PCS+18] Heather J. Painter, Neo Christopher Chung, Aswathy Sebastian, Istvan Albert, John D. Storey, and Manuel Llinás. Genome-wide real-time in vivo transcriptional dynamics during Plasmodium falciparum blood-stage development. *Nature Communications*, 9(1) :2656, 2018.

[PFH+13] Nadia Ponts, Lijuan Fu, Elena Y. Harris, Jing Zhang, Duk-Won D. Chung, Michael C. Cervantes, Jacques Prudhomme, Vessela Atanasova-Penichon, Enric Zehraoui, Evelien Bunnik, Elisandra M. Rodrigues, Stefano Lonardi, Glenn R. Hicks, Yinsheng Wang, and Karine G. Le Roch. Genome-wide mapping of DNA methylation in the human malaria parasite Plasmodium falciparum. *Cell host & microbe*, 14(6) :696–706, December 2013.

[Pic05] Franck Picard. Process segmentation/clustering. Application to the analysis of CGH microarray data. November 2005.

[PL88] W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8) :2444–2448, April 1988.

[PLC99] S. Park, D. Lee, and W. W. Chu. Fast retrieval of similar subsequences in long sequence databases. In *Proceedings 1999 Workshop on Knowledge and Data Engineering Exchange (KDEX'99) (Cat. No.PR00453)*, pages 60–67, November 1999.

[PM02] Loïc Ponger and Dominique Mouchiroud. CpGProD : identifying CpG islands associated with transcription start sites in large genomic mammalian sequences. *Bioinformatics (Oxford, England)*, 18(4) :631–633, April 2002.

[PROC14] Fabiano Sviatopolk-Mirsky Pais, Patrícia de Cássia Ruy, Guilherme Oliveira, and Roney Santos Coimbra. Assessing the efficiency of multiple sequence alignment programs. *Algorithms for Molecular Biology : AMB*, 9 :4, March 2014.

[PRT02] Dana Pe'er, Aviv Regev, and Amos Tanay. Minreg : inferring an active regulator set. *Bioinformatics (Oxford, England)*, 18 Suppl 1 :S258–267, 2002.

[PSL+13] Michaela Petter, Shamista A. Selvarajah, Chin Chin Lee, Wai Hoe Chin, Archna P. Gupta, Zbynek Bozdech, Graham V. Brown, and Michael F. Duffy. H2a.Z and H2b.Z double-variant nucleosomes define intergenic regions and dynamically occupy var gene promoters in the malaria parasite Plasmodium falciparum. *Molecular Microbiology*, 87(6) :1167–1182, March 2013.

[QWW98] Yunyao Qu, Changzhou Wang, and Xiaoyang Sean Wang. Supporting Fast Search in Time Series for Movement Patterns in Multiple Scales. In *CIKM*, volume 98, pages 251–258. Citeseer, 1998.

[Rab89] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, February 1989.

[Ram72] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3) :244–256, 1972.

[RB07] Emma Redhead and Timothy L. Bailey. Discriminative motif discovery in DNA and protein sequences using the DEME algorithm. *BMC bioinformatics*, 8 :385, October 2007.

[RCL+19] David F. Read, Kate Cook, Yang Y. Lu, Karine G. Le Roch, and William Stafford Noble. Predicting gene expression in the human malaria parasite Plasmodium falciparum using histone modification, nucleosome positioning, and 3d localization features. *PLoS computational biology*, 15(9) :e1007329, September 2019.

[RH05] Gajendra PS Raghava and Joon H. Han. Correlation and prediction of gene expression level from amino acid and dipeptide composition of its protein. *BMC Bioinformatics*, 6(1) :59, March 2005.

[RH18] Alejandro Reyes and Wolfgang Huber. Alternative start and termination sites of transcription drive most transcript isoform differences across human tissues. *Nucleic Acids Research*, 46(2) :582–592, January 2018.

[RHEC98] F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nature Biotechnology*, 16(10) :939–945, October 1998.

[RKC+13] Debashish Ray, Hilal Kazan, Kate B. Cook, Matthew T. Weirauch, Hamed S. Najafabadi, Xiao Li, Serge Gueroussov, Mihai Albu, Hong Zheng, Ally Yang, Hong Na, Manuel Irimia, Leah H. Matzat, Ryan K. Dale, Sarah A. Smith, Christopher A. Yarosh, Seth M. Kelly, Behnam Nabet, Desirea Mecenas, Weimin Li, Rakesh S. Laishram, Mei Qiao, Howard D. Lipshitz, Fabio Piano, Anita H. Corbett, Russ P. Carstens, Brendan J. Frey, Richard A. Anderson, Kristen W. Lynch, Luiz O. F. Penalva, Elissa P. Lei, Andrew G. Fraser, Benjamin J. Blencowe, Quaid D. Morris, and Timothy R. Hughes. A compendium of RNA-binding motifs for decoding gene regulation. *Nature*, 499(7457) :172–177, 2013.

[RKM+15] Roadmap Epigenomics Consortium, Anshul Kundaje, Wouter Meuleman, Jason Ernst, Misha Bilenky, Angela Yen, Alireza Heravi-Moussavi, Pouya Kheradpour, Zhizhuo Zhang, Jianrong Wang, Michael J. Ziller, Viren Amin, John W. Whitaker, Matthew D. Schultz, Lucas D. Ward, Abhishek Sarkar, Gerald Quon, Richard S. Sandstrom, Matthew L. Eaton, Yi-Chieh Wu, Andreas R. Pfenning, Xinchen Wang, Melina Claussnitzer, Yaping Liu, Cristian Coarfa, R. Alan Harris, Noam Shoresh, Charles B. Epstein, Elizabeta Gjoneska, Danny Leung, Wei Xie, R. David Hawkins, Ryan Lister, Chibo Hong, Philippe Gascard, Andrew J. Mungall, Richard Moore, Eric Chuah, Angela Tam, Theresa K. Canfield, R. Scott Hansen, Rajinder Kaul, Peter J. Sabo, Mukul S. Bansal, Annaick Carles, Jesse R. Dixon, Kai-How Farh, Soheil Feizi, Rosa Karlic, Ah-Ram Kim, Ashwinikumar Kulkarni, Daofeng Li, Rebecca Lowdon, GiNell Elliott, Tim R. Mercer, Shane J. Neph, Vitor Onuchic, Paz Polak, Nisha Rajagopal, Pradipta Ray, Richard C. Sallari, Kyle T. Siebenthall, Nicholas A. Sinnott-Armstrong, Michael Stevens, Robert E. Thurman, Jie Wu, Bo Zhang, Xin Zhou, Arthur E.

Beaudet, Laurie A. Boyer, Philip L. De Jager, Peggy J. Farnham, Susan J. Fisher, David Haussler, Steven J. M. Jones, Wei Li, Marco A. Marra, Michael T. McManus, Shamil Sunyaev, James A. Thomson, Thea D. Tlsty, Li-Huei Tsai, Wei Wang, Robert A. Waterland, Michael Q. Zhang, Lisa H. Chadwick, Bradley E. Bernstein, Joseph F. Costello, Joseph R. Ecker, Martin Hirst, Alexander Meissner, Aleksandar Milosavljevic, Bing Ren, John A. Stamatoyannopoulos, Ting Wang, and Manolis Kellis. Integrative analysis of 111 reference human epigenomes. *Nature*, 518(7539) :317–330, February 2015.

[RLB00] P. Rice, I. Longden, and A. Bleasby. EMBOSS : the European Molecular Biology Open Software Suite. *Trends in genetics : TIG*, 16(6) :276–277, June 2000.

[RMW+19] Chandra Ramakrishnan, Simone Maier, Robert A. Walker, Hubert Rehrauer, Deborah E. Joekel, Rahel R. Winiger, Walter U. Basso, Michael E. Grigg, Adrian B. Hehl, Peter Deplazes, and Nicholas C. Smith. An experimental genetically attenuated live vaccine to prevent transmission of Toxoplasma gondii by cats. *Scientific Reports*, 9, February 2019.

[RSdCREMC+05] Omar K. Ruvalcaba-Salazar, Ma del Carmen Ramírez-Estudillo, Dvorak Montiel-Condado, Félix Recillas-Targa, Miguel Vargas, and Rosaura Hernández-Rivas. Recombinant and native Plasmodium falciparum TATA-binding-protein binds to a specific TATA box element in promoter regions. *Molecular and Biochemical Parasitology*, 140(2) :183–196, April 2005.

[SAE+16] Andrew Schneider, Delasa Aghamirzaie, Haitham Elmarakeby, Arati N. Poudel, Abraham J. Koo, Lenwood S. Heath, Ruth Grene, and Eva Collakova. Potential targets of VIVIPAROUS1/ABI3-LIKE1 (VAL1) repression in developing Arabidopsis thaliana embryos. *The Plant Journal : For Cell and Molecular Biology*, 85(2) :305–319, January 2016.

[Sag98] Marie-France Sagot. Spelling approximate repeated or common motifs using a suffix tree. In *Latin American Symposium on Theoretical Informatics*, pages 374–390. Springer, 1998.

[SB08] Miho M. Suzuki and Adrian Bird. DNA methylation landscapes : provocative insights from epigenomics. *Nature Reviews. Genetics*, 9(6) :465–476, June 2008.

[SBL05] J. Soding, A. Biegert, and A. N. Lupas. The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Research*, 33(Web Server) :W244–W248, July 2005.

[SCFK14] Anna Stroynowska-Czerwinska, Agnieszka Fiszer, and Wlodzimierz J. Krzyzosiak. The panorama of miRNA-mediated mechanisms in mammalian cells. *Cellular and Molecular Life Sciences*, 71(12) :2253–2270, June 2014.

[SCK04] Meena Kishore Sakharkar, Vincent T. K. Chow, and Pandjassarame Kangueane. Distributions of exons and introns in the human genome. *In Silico Biology*, 4(4) :387–393, 2004.

[Ser02] F. Servant. ProDom : Automated clustering of homologous domains. *Briefings in Bioinformatics*, 3(3) :246–251, January 2002.

[SGG+17] Florian Schmidt, Nina Gasparoni, Gilles Gasparoni, Kathrin Gianmoena, Cristina Cadenas, Julia K. Polansky, Peter Ebert, Karl Nordström, Matthias Barann, Anupam Sinha, Sebastian Fröhler, Jieyi Xiong, Azim Dehghani Amirabad, Fatemeh Behjati Ardakani, Barbara Hutter, Gideon Zipprich, Bärbel Felder, Jürgen Eils, Benedikt Brors, Wei Chen, Jan G. Hengstler, Alf Hamann, Thomas Lengauer, Philip Rosenstiel, Jörn Walter, and Marcel H. Schulz. Combining transcription factor binding affinities with open-chromatin data for accurate gene expression prediction. *Nucleic Acids Research*, 45(1) :54–66, January 2017.

[SGS] Stephan Seifert, Sven Gundlach, and Silke Szymczak. Surrogate minimal depth as an importance measure for variables in random forests. *Bioinformatics*.

[SH00] G. A. Singer and D. A. Hickey. Nucleotide bias causes a genomewide bias in the amino acid composition of proteins. *Molecular Biology and Evolution*, 17(11) :1581–1588, November 2000.

[SH14] Fabian Sievers and Desmond G. Higgins. Clustal Omega, Accurate Alignment of Very Large Numbers of Sequences. In *Multiple Sequence Alignment Methods*, Methods in Molecular Biology, pages 105–116. Humana Press, Totowa, NJ, 2014.

[Sha53] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28) :307–317, 1953.

[SKB+96] Kimmen Sjölander, Kevin Karplus, Michael Brown, Richard Hughey, Anders Krogh, I. Saira Mian, and David Haussler. Dirichlet mixtures : a method for improved detection of weak but significant protein sequence homology. *Bioinformatics*, 12(4) :327–345, August 1996.

[SLS+14] Shula Shazman, Hunjoong Lee, Yakov Socol, Richard S. Mann, and Barry Honig. OnTheFly : a database of Drosophila melanogaster transcription factors and their binding sites. *Nucleic Acids Research*, 42(Database issue) :D167–171, January 2014.

[SN87] N. Saitou and M. Nei. The neighbor-joining method : a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4) :406–425, July 1987.

[SO94a] Andreas Stolcke and Stephen Omohundro. Inducing probabilistic grammars by Bayesian model merging. In *Grammatical Inference and Applications*, Lecture Notes in Computer Science, pages 106–118. Springer, Berlin, Heidelberg, September 1994.

[SO94b] N Sugiura and RT Ogden. Testing change-points with linear trend. *Communications in Statistics-Simulation and Computation*, 23(2) :287–322, 1994.

[SPR+04] Naresh Singh, Peter Preiser, Laurent Rénia, Bharath Balu, John Barnwell, Peter Blair, William Jarra, Tatiana Voza, Irene Landau, and John H. Adams. Conservation and developmental control of alternative splicing in maebl among malaria parasites. *Journal of Molecular Biology*, 343(3) :589–599, October 2004.

[SSGE82] Gary D. Stormo, Thomas D. Schneider, Larry Gold, and Andrzej Ehrenfeucht. Use of the 'perceptron' algorithm to distinguish translational initiation sites in e. coli. *Nucleic Acids Research*, 10(9) :2997–3011, 1982.

[ST00] Saurabh Sinha and Martin Tompa. A statistical method for finding transcription factor binding sites. In *ISMB*, volume 8, pages 344–354, 2000.

[STM+05] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis : a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43) :15545–15550, October 2005.

[Sto00] Gary D. Stormo. DNA binding sites : representation and discovery. *Bioinformatics*, 16(1) :16–23, January 2000.

[STS96] M. K. Shaw, J. Thompson, and R. E. Sinden. Localization of ribosomal RNA and Pbs21-mRNA in the sexual stages of Plasmodium berghei using electron microscope in situ hybridization. *European journal of cell biology*, 71(3) :270–276, November 1996.

[SW81] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1) :195–197, March 1981.

[SWH+15] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and the UniProt Consortium. UniRef clusters : a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6) :926–932, March 2015.

[SYK03] E. Segal, R. Yelensky, and D. Koller. Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics (Oxford, England)*, 19 Suppl 1 :i273–282, 2003.

[SZ96] Hagit Shatkay and Stanley B. Zdonik. Approximate queries and representations for large data sequences. *Proceedings of the Twelfth International Conference on Data Engineering*, pages 536–545, 1996.

[Tay86] W. R. Taylor. The classification of amino acid conservation. *Journal of Theoretical Biology*, 119(2) :205–218, March 1986.

[TGMB09] N. Terrapon, O. Gascuel, E. Marechal, and L. Bréhélin. Detection of new protein domains using co-occurrence : application to Plasmodium falciparum. *Bioinformatics*, 25(23) :3077–3083, December 2009.

[TGMB12] Nicolas Terrapon, Olivier Gascuel, Éric Maréchal, and Laurent Bréhélin. Fitting hidden Markov models of protein domains to a target species : application to Plasmodium falciparum. *BMC Bioinformatics*, 13(1) :67, 2012.

[The15] The UniProt Consortium. UniProt : a hub for protein information. *Nucleic Acids Research*, 43(D1) :D204–D212, January 2015.

[The19] The UniProt Consortium. UniProt : a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1) :D506–D515, January 2019.

[THG94] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W : improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22) :4673–4680, November 1994.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[TJ02] Daiya Takai and Peter A. Jones. Comprehensive analysis of CpG islands in human chromosomes 21 and 22. *Proceedings of the National Academy of Sciences of the United States of America*, 99(6) :3740–3745, March 2002.

[TMP+18] Miguel C Teixeira, Pedro T Monteiro, Margarida Palma, Catarina Costa, Cláudia P Godinho, Pedro Pais, Mafalda Cavalheiro, Miguel Antunes, Alexandre Lemos, Tiago Pedreira, and Isabel Sá-Correia. YEASTRACT : an upgraded database for the analysis of transcription regulatory networks in Saccharomyces cerevisiae. *Nucleic Acids Research*, 46(Database issue) :D348–D353, January 2018.

[Tom99] Martin Tompa. An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. In *ISMB*, volume 99, pages 262–271, 1999.

[TP15] Deborah A. Triant and William R. Pearson. Most partial domains in proteins are alignment and annotation artifacts. *Genome Biology*, 16 :99, May 2015.

[TS92] J. Takami and S. Sagayama. A successive state splitting algorithm for efficient allophone modeling. In *[Proceedings] ICASSP-92 : 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 573–576 vol.1, March 1992.

[TV07] Hélène Touzet and Jean-Stéphane Varré. Efficient and accurate P-value computation for Position Weight Matrices. *Algorithms for molecular biology : AMB*, 2 :15, December 2007.

[VBB+04] Christine Vogel, Carlo Berzuini, Matthew Bashton, Julian Gough, and Sarah A. Teichmann. Supra-domains : evolutionary units larger than single protein domains. *Journal of Molecular Biology*, 336(3) :809–823, February 2004.

[vHACV98] J. van Helden, B. André, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of Molecular Biology*, 281(5) :827–842, September 1998.

[vHRCV00] J. van Helden, A. F. Rios, and J. Collado-Vides. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Research*, 28(8) :1808–1818, April 2000.

[vLPJ+01] Leonard H. M. van Lin, Tomasino Pace, Chris J. Janse, Cecilia Birago, Jai Ramesar, Leonardo Picci, Marta Ponzi, and Andrew P. Waters. Interspecies conservation of gene order and intron-exon structure in a genomic locus of high gene density and complexity in Plasmodium. *Nucleic Acids Research*, 29(10) :2059–2068, May 2001.

[VTPL05] Christine Vogel, Sarah A. Teichmann, and Jose Pereira-Leal. The Relationship Between Domain Duplication and Recombination. *Journal of Molecular Biology*, 346(1) :355–365, February 2005.

[VVV97] H. J. L. M. Vullings, M. H. G. Verhaegen, and H. B. Verbruggen. ECG Segmentation Using Time-Warping. In *Proceedings of the Second International Symposium on Advances in Intelligent Data Analysis, Reasoning About Data*, IDA '97, pages 275–285, London, UK, UK, 1997. Springer-Verlag.

[WAJ97] M. Wickens, P. Anderson, and R. J. Jackson. Life and death in the cytoplasm : messages from the 3' end. *Current Opinion in Genetics & Development*, 7(2) :220–232, April 1997.

[WDKK96] E. Wingender, P. Dietze, H. Karas, and R. Knüppel. TRANSFAC : a database on transcription factors and their DNA binding sites. *Nucleic Acids Research*, 24(1) :238–241, January 1996.

[WJ94] L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology : A Journal of Computational Molecular Cell Biology*, 1(4) :337–348, 1994.

[WL04] Yong Wang and Frederick C. C. Leung. An evaluation of new criteria for CpG islands in the human genome as gene markers. *Bioinformatics (Oxford, England)*, 20(7) :1170–1177, May 2004.

[WMBB08] January Weiner, Andrew D. Moore, and Erich Bornberg-Bauer. Just how versatile are domains ? *BMC Evolutionary Biology*, 8(1) :285, October 2008.

[Woo94] John C. Wootton. Non-globular domains in protein sequences : Automated segmentation using complexity measures. *Computers & Chemistry*, 18(3) :269–285, September 1994.

[WW00] Changzhou Wang and Xiaoyang Wang. Supporting Content-based Searches on Time Series via Approximation. In *Proceedings of the International Conference on Scientific and Statistical Database Management, SSDBM*, pages 69 – 81, 2000.

[WXL+18] Jingyi Wu, Jiawei Xu, Bofeng Liu, Guidong Yao, Peizhe Wang, Zili Lin, Bo Huang, Xuepeng Wang, Tong Li, Senlin Shi, Nan Zhang, Fuyu Duan, Jia Ming, Xiangyang Zhang, Wenbin Niu, Wenyan Song, Haixia Jin, Yihong Guo, Shanjun Dai, Linli Hu, Lanlan Fang, Qiujun Wang, Yuanyuan Li, Wei Li, Jie Na, Wei Xie, and Yingpu Sun. Chromatin analysis in human early development reveals epigenetic transition during ZGA. *Nature*, 557(7704) :256–260, 2018.

[XF03] H. Y. Xue and D. R. Forsdyke. Low-complexity segments in Plasmodium falciparum proteins are primarily nucleic acid level adaptations. *Molecular and Biochemical Parasitology*, 128(1) :21–32, April 2003.

[Ye04] Y. Ye. Comparative Analysis of Protein Domain Organization. *Genome Research*, 14(3) :343–353, March 2004.

[YJP+18] Haiwang Yang, Maria Jaime, Maxi Polihronakis, Kelvin Kanegawa, Therese Markow, Kenneth Kaneshiro, and Brian Oliver. Re-annotation of eight Drosophila genomes. *Life Science Alliance*, 1(6), December 2018.

[YLMR19] Lee M. Yeoh, V. Vern Lee, Geoffrey I. McFadden, and Stuart A. Ralph. Alternative Splicing in Apicomplexan Parasites. *mBio*, 10(1) :e02866–18, February 2019.

[ZDXF16] Haomiao Zhou, Zhihong Deng, Yuanqing Xia, and Mengyin Fu. A new sampling method in particle filter based on Pearson correlation coefficient. *Neurocomputing*, 216 :208–215, December 2016.

[ZG11] Christian M. Zmasek and Adam Godzik. Strong functional patterns in the evolution of eukaryotic genomes revealed by the reconstruction of ancestral protein domain repertoires. *Genome Biology*, 12(1) :R4, 2011.

[ZJL+07] Jing Zhang, Bo Jiang, Ming Li, John Tromp, Xuegong Zhang, and Michael Q. Zhang. Computing exact P-values for DNA motifs. *Bioinformatics*, 23(5) :531–537, March 2007.

[ZL09] Zheng Zhao and Huan Liu. Searching for interacting features in subset selection. *Intelligent Data Analysis*, 13(2) :207–228, 2009.

[ZSM+98] Z. Zhang, A. A. Schäffer, W. Miller, T. L. Madden, D. J. Lipman, E. V. Koonin, and S. F. Altschul. Protein sequence similarity searches using patterns as seeds. *Nucleic Acids Research*, 26(17) :3986–3990, September 1998.

[ZT15] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10) :931–934, October 2015.

[ZTY+18] Jian Zhou, Chandra L. Theesfeld, Kevin Yao, Kathleen M. Chen, Aaron K. Wong, and Olga G. Troyanskaya. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nature genetics*, 50(8) :1171–1179, August 2018.

[ZY17] Harel Zalts and Itai Yanai. Developmental constraints shape the evolution of the nematode mid-developmental transition. *Nature ecology & evolution*, 1(5) :0113, 2017.

Résumé : Identifier les différentes parties d'une séquence biologique (séquence nucléique, ou séquence d'acides aminés) constitue un premier pas vers la compréhension de la biologie de l'organisme dont elle est issue. Étant donné un ensemble de séquences biologiques d'un organisme, nous nous intéressons dans cette thèse à la découverte de «domaines», c-à-d de sous-séquences relativement grandes (plusieurs dizaines de nucléotides ou d'acides aminés) que l'on retrouve dans un nombre important de séquences. Cette thèse est décomposée en deux axes correspondant à la découverte de domaines dans les séquences protéiques et dans les séquences nucléiques. Dans chaque axe, les méthodes développées sont appliquées à *Plasmodium falciparum*, le pathogène responsable du paludisme chez l'Homme, et pour lequel les méthodes bioinformatiques classiques peinent à produire des annotations satisfaisantes. Le premier axe développé porte sur la découverte de domaines dans les séquences protéiques. Une approche commune pour identifier les domaines d'une protéine consiste à exécuter des comparaisons de paires de séquences avec des outils d'alignements locaux comme BLAST. Cependant, ces approches manquent parfois de sensibilité, en particulier pour les espèces phylogénétiquement éloignées des organismes de référence classiques. Nous proposons ici une approche pour augmenter la sensibilité des comparaisons de paires de séquences. Cette nouvelle approche utilise le fait que les domaines protéiques ont tendance à apparaître avec un nombre limité d'autres domaines sur une même protéine. Chez *Plasmodium falciparum*, cette méthode permet la découverte de 2 240 nouveaux domaines pour lesquels, dans la majorité des cas, il n'existe pas de modèle semblable dans les bases de données de domaines. Le deuxième axe développé porte sur la découverte de domaines dans les séquences régulatrices (séquences ADN). Plusieurs études ont montré qu'il existe un lien fort entre la composition nucléotidique de régions particulières (séquences promotrices notamment) et l'expression des gènes. Nous proposons ici une nouvelle approche permettant de découvrir de manière automatique ces régions, que l'on nomme domaines de régulation. Plus précisément notre approche est basée sur une stratégie d'exploration itérative des compositions nucléotidiques, des plus simples (dinucléotides) aux plus complexes (k-mers), ainsi qu'une stratégie de segmentation supervisée pour découvrir les compositions et les régions d'intérêt. En utilisant les domaines ainsi identifiés, nous montrons que l'on peut prédire l'expression des gènes de *Plasmodium falciparum* avec une étonnante précision. Appliquée à différentes autres espèces eucaryotes, cette approche montre des résultats très différents suivant les espèces (entre 40 et 70% de corrélation) ce qui laisse entrevoir un mécanisme de régulation sans doute partagé par toutes les espèces eucaryotes mais dont l'importance varie d'une espèce à l'autre.

Mots-clés : domaines protéiques, domaines de régulation, régulation de la transcription, paludisme, machine learning, feature extraction

---

Abstract: Identifying the different parts of a biological sequence (nucleic sequence, or amino acid sequence) is a first step toward understanding the biology of the organism from which it originates. Given a set of biological sequences of an organism, we are interested in this thesis to the discovery of «domains», ie of relatively large subsequences (several tens of nucleotides or amino acids) that the we can find in a large number of sequences. This thesis is decomposed into two parts corresponding to the discovery of domains in the protein sequences and in the nucleic sequences. In each part, the methods developed are applied to *Plasmodium falciparum*, the pathogen responsible for malaria in humans, and for which conventional bioinformatic methods struggle to produce satisfactory annotations. The first developed part relates to the discovery of domains in protein sequences. A common approach to identifying domains of a protein is to perform sequence-sequence comparisons with local alignment tools such as BLAST. However, these approaches sometimes lack sensitivity, particularly for species phylogenetically distant from conventional reference organisms. Here we propose an approach to increase the sensitivity of sequence-sequence comparisons. This new approach uses the fact that protein domains tend to appear with a limited number of other domains on the same protein. In *Plasmodium falciparum*, this method allows the discovery of 2 240 new domains for which, in the majority of cases, there is no similar model in domain databases. The second developed part relates to the discovery of domains in regulatory sequences (DNA sequences). Several studies have shown that there is a strong link between the nucleotide composition of particular regions (promoter sequences in particular) and the expression of genes. We propose here a new approach to discover automatically these regions, which we call regulatory domains. More specifically, our approach is based on a strategy of iterative exploration of nucleotide compositions, from the simplest (dinucleotides) to the most complex (k-mers), as well as a supervised segmentation strategy to discover compositions and regions of interest. Using the domains thus identified, we show that the expression of *Plasmodium falciparum* genes can be predicted with good precision. Applied to various other eukaryotic species, this approach shows very different results depending on the species (between 40 and 70% correlation) which suggests a regulatory mechanism probably shared by all eukaryotic species but whose importance varies from one species to another.

Keywords: protein domains, regulatory domains, transcriptional regulation, malaria, machine learning, feature extraction