Recalage flexible de modèles moléculaires dans les reconstructions 3D de microscopie électronique

Gaël Goret

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Biologie Structurale et Nanobiologie**

Arrêté ministérial : du 7 août 2006

Présentée par

Gaël Goret

Thèse dirigée par **Jorge Navaza**

préparée au sein **de l'Institut de Biologie Structurale**
et de **l'Ecole Doctorale Chimie et Sciences du Vivant**

Recalage flexible de modèles moléculaires dans les reconstructions 3D de microscopie électronique

Thèse soutenue publiquement le **26 septembre 2011 à 10h**,
devant le jury composé de :

M. Gilbert Deleage
Professeur, Université Lyon 1, Président
M. Jean Lepault
Directeur de Recherche, CNRS, Rapporteur
M. Jean-Christophe Taveau
Professeur, Université de Bordeaux 1, Rapporteur
M. Juan Rodríguez-Carvajal
Directeur de Recherche, ILL, Examinateur
M. Stefano Trapani
Maitre de Conférence, Université Montpellier 2, Examinateur
M. Jorge Navaza
Directeur de Recherche, CNRS, Directeur de thèse

“Qui n’est jamais tombé

n’a pas une idée juste de l’effort à faire pour se tenir debout”

Multatuli

Remerciements

J'aimerais tout d'abord remercier mon directeur de thèse Dr Jorge Navaza, pour avoir accepté de me prendre comme élève et m'avoir guidé durant ces années, pour les longues discussions passées à faire les cent pas devant l'IBS, pour être un modèle de rigueur et de précision, un intarissable puits de science, ayant toujours réponse aux questions que je pose (voir celles que j'oublie de poser). Enfin, merci pour sa patience, sa compréhension et son exceptionnelle disponibilité. (PS : Jorge, n'oublie pas de demander à Alda la recette du chimichurri.)

J'aimerais ensuite remercier l'ensemble des membres de mon jury d'avoir accepté, bien volontiers, d'endosser leurs fonctions respectives : je remercie le président du jury Pr Gilbert Deleage de m'avoir fait découvrir, il y a quelques années, la bio-informatique structurale ; je remercie également mes rapporteurs Dr Jean Lepault et Pr Jean-Christophe Taveaux pour le temps et l'attention qu'ils ont accordés à la lecture de mon manuscrit ; un grand merci au Dr Stefano Trapani pour ses minutieuses corrections, ses conseils et pour m'avoir "prêté" ses étudiants le temps d'une journée ; merci au Dr Juan Rodriguez-Carvajal pour avoir été membre de mon comité de suivi de thèse et pour avoir accepté par défaut le statut d'examinateur (merci l'administration !).

J'adresse un remerciement tout particulier à mon amie et professeur Dominique Mouchiroud, qui en plus d'être une source d'inspiration pour moi, est aussi une femme compréhensive douée d'une infinie compassion, sans laquelle je ne me tiendrai pas debout aujourd'hui.

Je tiens, bien sûr, à remercier ma femme Cécile, pour son amour, sa patience et pour avoir toujours cru en moi. Merci à elle d'avoir réussi à me convaincre qu'ensemble nous pourrions gérer en même temps la thèse, un mariage et l'arrivée prochaine d'un bébé ... je vous aime.

Un immense merci à mes parents pour m'avoir tenu la main longtemps après mes premiers pas, pour m'avoir aidé de tout leur possible à trouver ma voie et à toujours faire ressortir le meilleur de moi. Merci pour leur indéfectible soutien durant les heures les plus sombres, mais surtout merci de leur amour, leur respect et leur confiance. (PS : Comme promis vous aurez la barre d'un des T de doctorat.)

Ces trois ans à l'IBS ont été riches de rencontres, de collaborations et d'amitié. Je tiens à remercier le groupe de microscopie électronique qui m'a accueilli avec beaucoup de gentillesse. Un "thank you" en particulier à Winnie Ling avec qui j'ai partagé le bureau dans la tranquillité et la bonne humeur, merci de m'avoir aidé à progresser en anglais. Un grand merci à Leandro Estrozi, qui a joué le rôle de grand frère en me conseillant tout au long de mes développements. Merci également à Emmanuelle Neumann et Frédéric Metoz pour leurs nombreux conseils et corrections concernant mon manuscrit de thèse. Merci à Guy Schoehn pour sa gentillesse (bien dissimulée) et pour l'intérêt et le souci qu'il a manifestés pour moi. Une pensée particulière pour Irina Gutsche, principale collaboratrice microscopiste sans qui je n'aurais pas publié d'articles.

La chanson dit "les copains d'abord", dans l'ordre des remerciements je vais terminer par eux : Christian, Eric, Juliette, Karin, Romain, Richard, Sarah et Vincent merci à tous d'être ce que vous êtes. Dans le désordre, petites dédicaces, je ne doute pas qu'ils retrouveront celle qui leur correspond :

- Grand philosophe à la barbe rousse, cet olibrius spécialiste dans l'art de la sentence, a plus de poils sur le torse que sur la tête.
- Ami de la première heure, la tête dans les étoiles, cet amoureux des voies verticales m'a donné confiance dans mes capacités de thésard.
- Atteint d'une diarrhée verbale chronique, ce montagnard au cœur tendre aime choquer les esprits dans la salle du biacore avec son compère le dandy.
- Dandy parigot à l'humour aussi léger que les atomes qu'il utilise pour phaser ses données, ce grand ténébreux aimerait bien rentrer dans la marine (plus souvent).
- Moustachu à l'œil acéré, grand spécialiste de la cristallographie et de la plantation d'asperges, ce petit bonhomme, mi sage, mi fou a découvert sa panacée (10 tasses de café par jour ...).
- Amie de la première heure, cause première de mes dépassements de forfait, cette jeune femme dynamique, est toujours à l'écoute pour m'entendre raconter mes malheurs (et réciproquement !)
- Celle là, comme le dirait le montagnard c'est : ≪ tu sais celle qui roule du cul ≫... Un brin rêveuse et bohème, cette éternelle sensible, va bientôt se faire dorer la pilule sur les plages de Floride ...
- Cette petite fée a débarqué à l'IBS à peu près en même temps que nous autres thésards, joli brin de fille, touchante et rigolote, elle est aussi une habituée des (parfois nombreuses) pauses cigarette.

En bref, merci à toutes et tous !!!

Table des matières

Table des figures

Chapitre 1

Introduction

La biologie structurale est un domaine de la biologie à l'interface de nombreuses disciplines comme la chimie, la physique, les mathématiques ou encore l'informatique. Elle est devenue un domaine incontournable grâce à des techniques comme la cristallographie aux rayons X, la microscopie électronique (ME) et la résonance magnétique nucléaire (RMN) qui ont permis de mieux observer l'univers microscopique, fournissant une vision nouvelle de la biologie et permettant ainsi de mieux appréhender le vivant.

Aujourd'hui la cristallographie de macromolécules (MX) produit couramment des modèles moléculaires à résolution atomique, allant typiquement des protéines isolées aux complexes protéiques. Cependant, cette technique est particulièrement difficile à mettre en œuvre, voir inapplicable, dans le cas de complexes de taille importante.

D'un autre côté, la microscopie électronique permet de visualiser des particules de grande taille, allant des assemblages macromoléculaires à la cellule, dans des conditions proches de celles *in vivo*. Cependant, la résolution des reconstructions tridimensionnelles (3D) obtenues exclut, en général, leur interprétation directe en termes de structures moléculaires, étape nécessaire à la compréhension des problèmes biologiques.

Il est donc naturel d'essayer de combiner les informations fournies par ces deux techniques pour caractériser la structure des assemblages macromoléculaires. L'idée est de positionner les modèles moléculaires déterminés par MX à l'intérieur de reconstructions 3D issues de la ME, et de comparer la densité électronique associée à la reconstruction 3D avec une densité électronique calculée à partir des modèles. Le problème numérique réside dans la détermination et l'optimisation des variables qui spécifient les positions

des modèles, considérés comme des corps rigides, à l'intérieur de l'assemblage. Cette idée simple a donné lieu au développement d'une méthodologie appelée recalage[1].

Il existe une analogie entre le recalage en microscopie électronique et la construction de modèles moléculaires en MX. En effet, les données expérimentales en MX n'atteignent pas, en général, la résolution atomique. Pourtant, si la résolution est suffisante pour tracer les chaînes principales des protéines et pour en distinguer les résidus, la connaissance de la structure des acides aminés permet de placer les résidus comme des corps rigides et d'effectuer ainsi une interprétation directe des cartes de densité électronique en termes de modèles atomiques. De manière analogue, grâce à l'ajout de l'information provenant des motifs structuraux connus, la méthode de recalage permet d'interpréter les cartes basse-résolution des assemblages macromoléculaires en termes de modèles moléculaires.

Cependant, une limitation de cette méthode provient du fait que les molécules constituant les assemblages macromoléculaires peuvent ne pas correspondre aux structures moléculaires obtenues par cristallographie aux rayons X. Souvent les protéines impliquées dans ces assemblages présentent des changements conformationnels importants par rapport à leur structure isolée. Pour pallier cette limitation il est nécessaire d'intégrer dans la méthode de recalage la déformation des modèles moléculaires.

Le type de déformation intéressant, car proche des mouvements fonctionnels des bio-molécules, comprend les déplacements collectifs des domaines structuraux. Ce type de mouvement correspond aux vibrations de basse énergie des molécules et est décrit par les modes normaux de vibration de plus basse fréquence[2]. Le relâchement de la contrainte de rigidité des modèles moléculaires se fait donc par l'application d'une dynamique harmonique, sous la contrainte de la reconstruction 3D de ME. Le problème numérique consiste alors dans la détermination des amplitudes associées aux modes considérés.[3]

La détermination de structures à haute résolution des gros assemblages macromoléculaires, quête majeure des biologistes expérimentateurs, reste et restera extrêmement difficile dans un futur proche. C'est pourquoi ce travail de thèse a eu pour but de fournir aux biologistes un outil leur permettant de construire des modèles pseudo-moléculaires associés aux assemblages produits par ME.

Plusieurs méthodologies ont été développées par le passé pour automa-

tiser l'interprétation des reconstructions de ME à basse résolution en termes de modèles moléculaires haute résolution[1, 4, 5]. La plupart de ces méthodes calculatoires ont également été implémentées sous forme de logiciels disponibles pour la communauté scientifique (*e.g.* EMfit, URO, Situs,...). Plus récemment, afin d'améliorer l'affinement des modèles macromoléculaires résultant du recalage, certains chercheurs se sont penchés sur les techniques de recalage flexible utilisant les modes normaux[6, 3], ou encore utilisant des simulations de dynamique moléculaire[7, 8]. Pourtant, encore aujourd'hui, parmi l'ensemble des logiciels de recalage, seul un petit nombre incluent un support de visualisation intégré (*e.g.* Chimera[9], Sculptor[10] ou UROX[11]).

Dans la plupart des cas, cette absence d'interface graphique dédiée impose, en utilisation courante, des allers-retours incessants entre l'outil de calcul et une plateforme de visualisation externe. En outre, aucun développement ne cumule avec stabilité un environnement graphique convivial, la gestion de la flexibilité et un moteur de calcul performant (calcul rapide, traitement de symétries complexes, utilisation de grands volumes, ...).

C'est pourquoi ce travail de thèse s'est tout d'abord orienté vers le développement d'un logiciel interactif permettant de combiner de manière efficace et conviviale l'aspect calculatoire, en grande partie basé sur URO[4], avec un environnement graphique 2D/3D intégré. Par la suite, le travail a été focalisé sur l'application du logiciel à des cas réels, prouvant son utilité et son efficacité.

Le logiciel issu de ce travail, nommé $\vee\in\supset\wedge$ [1], est aujourd'hui pleinement fonctionnel. Il est utilisé par un nombre croissant de chercheurs, en France et à l'étranger qui lui reconnaissent tous facilité d'utilisation, stabilité, rapidité et qualité des résultats.

1. Pour **V**isual **E**nvironment for **D**ocking **A**lgorithms

Chapitre 2

Méthodologie

Les données utilisées dans la méthode du recalage (coordonnées atomiques pour MX ou RMN et carte de densité électronique pour ME) ne sont pas homogènes. Pourtant, elles décrivent toutes des objets physiques de nature identique (structures moléculaires).

Une première différence, la plus évidente, est bien sûr la taille des objets décrits.La ME fournit une vision globale des assemblages macromoléculaires alors que d'autres techniques comme la MX ou la RMN fournissent celle de composants isolés. Une deuxième différence, plus spécifique au problème de recalage, réside dans l'écart de résolution existant entre les modèles moléculaires issus de techniques dites haute résolution (RMN, MX, théoriquement de résolution infinie) et les reconstructions d'assemblages issues de ME ($\sim$ 10Å-30Å). Ces deux spécificités, liées aux données, vont se contraindre mutuellement dans la problématique du recalage.

Certaines reconstructions de ME considérées à l'heure actuelle comme de haute résolution ($\sim$ 5Å) permettent d'isoler les volumes correspondant aux sous-unités qui les composent. Dans ce cas particulier le recalage peut être effectué sur un volume minimal. Cependant, les reconstructions de ce type sont rares. Dans la grande majorité des cas les cartes produites par ME, à plus basse résolution ($>$ 10Å), rendent impossible la détermination des parties correspondant aux molécules individuelles. Pour pallier cette difficulté il est alors nécessaire de sélectionner de larges volumes permettant de ne pas laisser d'incertitude en ce qui concerne les zones limites séparant les sous-unités à l'intérieur de la densité électronique.

Étendre la sélection du volume utilisé pour le recalage signifie également augmenter le nombre de modèles moléculaires à placer et par conséquent aug-

menter de manière importante le nombre de variables à optimiser. Heureusement les gros assemblages possèdent souvent des symétries intrinsèques (*e.g.* virus, particules virales, complexe microtubule/moteurs moléculaires, etc.). Dans ce cas seules les molécules indépendantes doivent être considérées dans le processus d'optimisation, ce qui réduit substantiellement le nombre de degrés de liberté du problème.

L'ensemble de ces particularités ont conditionné la stratégie suivie pour résoudre le problème de recalage.

2.1 Formulation du problème de recalage en corps rigide

Étant donnée la reconstruction 3D de ME, appelée par la suite volume ME, correspondant à un assemblage macromoléculaire, le but est de construire un assemblage similaire à partir de modèles moléculaires connus. Le problème du recalage est de déterminer les positions de ces modèles moléculaires, considérés dans un premier temps comme des corps rigides, à l'intérieur du volume ME.
Définissons tout d'abord les concepts de base de cette formulation.

2.1.1 Modèles et molécules

Typiquement chaque modèle moléculaire est accessible à travers la Protein Data Bank (PDB) sous la forme d'un fichier contenant les coordonnées cartésiennes de l'ensemble des atomes qui le composent. Ces coordonnées correspondent au modèle dans une position qui est en rapport avec le cristal ayant permis la détermination de la structure moléculaire. Cette position particulière n'est par relevante pour le problème considéré, ce qui permet d'en choisir une autre, appelée position de référence, ayant une réelle utilité numérique et conceptuelle. Dans cette nouvelle position le modèle a son centre de masse à l'origine et ses axes d'inertie principaux parallèles aux axes d'un repère orthonormé. L'ensemble des coordonnées atomiques du modèle ainsi positionné est noté $\{\mathbf{x}^o\}$.

Chaque molécule constituant l'assemblage est représentée par un modèle moléculaire placé à l'intérieur du volume ME. L'ensemble de coordonnées atomiques décrivant la position courante du modèle est noté $\{\mathbf{x}\}$. Cette position est obtenue à partir de la position de référence du modèle par application

de la rotation $\mathbf{R}$ et de la translation $\mathbf{T}$ qui satisfont

$$\{\mathbf{x}\} = \mathbf{R}\,\{\mathbf{x}^o\} + \mathbf{T}$$

$\mathbf{R}$ et $\mathbf{T}$ jouent le rôle de variables de position du modèle car ils en déterminent sa position courante.

Par la suite un modèle moléculaire dans sa position de référence sera appelé simplement modèle et sera identifié par le symbole μ.

Un modèle moléculaire dans une position courante définie par les variables de position $(\mathbf{R}, \mathbf{T})$ sera appelé molécule. Chaque molécule est donc identifiable de manière univoque par un triplet $(\mu, \mathbf{R}, \mathbf{T})$.

2.1.2 Symétrie

Un assemblage symétrique est par définition invariant par rapport à l'application de certaines transformations (rotations et translations) qui forment souvent un groupe au sens mathématique du terme. On notera $\mathcal{M}_g$ et $\mathcal{T}_g$, respectivement, la matrice de rotation et le vecteur de translation de la g-ième opération de symétrie du groupe G.

Les molécules d'un assemblage symétrique ne sont pas toutes indépendantes. Il est en effet possible de générer tout l'assemblage à partir des molécules indépendantes. Pour chaque molécule indépendante identifiée par le triplet $(\mu, \mathbf{R}, \mathbf{T})$, la combinaison des opérations de symétrie avec les variables de position génère d'autres molécules dont les variables de position sont

$$(\mathcal{M}_g, \mathcal{T}_g) \otimes (\mathbf{R}, \mathbf{T}) = (\mathcal{M}_g\mathbf{R}\,,\ \mathcal{M}_g\mathbf{T} + \mathcal{T}_g) \tag{2.1}$$

Ces molécules, identifiées par les triplets $(\mu, \mathcal{M}_g\mathbf{R}\,,\ \mathcal{M}_g\mathbf{T} + \mathcal{T}_g)$ avec $g \in G$, constituent la constellation de la molécule indépendante [1].

Quand la molécule indépendante possède elle même une symétrie, certaines des molécules de sa constellation peuvent coïncider [2]. Par la suite une constellation ne contiendra que des molécules distinctes. Ainsi définies, l'ensemble de constellations des molécules indépendantes constituent l'assemblage tout entier.

1. La constellation inclut, en générale, la molécule indépendante
2. C'est à dire, occuper des positions correspondant à une même sous-unité dans la reconstruction

2.1.3 Densité électronique et facteurs de structure de l'assemblage

Considérons maintenant la fonction de densité électronique $\rho_\mu(\mathbf{r})$ associée aux coordonnées atomiques $\{\mathbf{x}^o\}$ d'un modèle μ. La densité électronique d'une molécule $\rho(\mathbf{r})$, s'écrit alors

$$\rho(\mathbf{r}) = \rho_\mu(\mathbf{R}^{-1}\mathbf{r} - \mathbf{T})$$

ou encore, par simplicité :

$$\rho = (\mathbf{R}, \mathbf{T})\rho_\mu$$

La densité de l'assemblage macromoléculaire ρ^{ass} peut être écrite comme étant la somme des contributions des densités de chacune des molécules constituant l'assemblage :

$$\rho^{ass} = \sum_{m \in M} \sum_{g \in G} (\mathcal{M}_g, \mathcal{T}_g) \otimes (\mathbf{R}_m, \mathbf{T}_m)\rho_{\mu(m)}$$

ou m désigne une molécule indépendante et M l'ensemble de toutes les molécules indépendantes présentes dans l'assemblage.

Par la suite seront considérées non plus seulement des densités mais également leurs transformées de Fourier. Celles des modèles sont appelées **facteurs de diffusion moléculaire** tandis que celles de l'assemblage sont appelées **facteurs de structure**. La transformée de Fourier (TF) de $\rho_\mu(\mathbf{r})$ s'exprime comme suit :

$$\rho_\mu(\mathbf{r}) \xrightarrow{TF} \int_{-\infty}^{\infty} \rho_\mu(\mathbf{r}) e^{2\pi i \mathbf{s}\mathbf{r}} d^3\mathbf{r} = f_\mu(\mathbf{s})$$

Regardons comment s'écrit alors la transformée de Fourier de la densité associée à une molécule :

$$\rho(\mathbf{r}) \xrightarrow{TF} \int_{-\infty}^{\infty} \rho_\mu(\mathbf{R}^{-1}\mathbf{r} - \mathbf{T}) e^{2\pi i \mathbf{s}\mathbf{r}} d^3\mathbf{r} = f_\mu(\mathbf{s}\mathbf{R}) e^{2\pi i \mathbf{s}\mathbf{T}} \tag{2.2}$$

La dernière expression sous forme factorisée illustre le fait que la rotation appliquée à la molécule dans l'espace réel se traduit dans l'espace réciproque par l'évaluation de la fonction f_μ en $\mathbf{sR}$, c'est à dire en une coordonnée tournée. La translation quant à elle se traduit par la multiplication du facteur de diffusion moléculaire par le facteur de phase $e^{2\pi i \mathbf{s}\mathbf{T}}$.

En tenant compte de (2.1) et (2.2), il est plus aisé de comprendre l'expression de la transformée de Fourier de la densité de l'assemblage :

$$F^{ass}(\mathbf{s}) = \sum_{m \in M} \sum_{g \in G} f_{\mu(m)}(\mathbf{s}\mathcal{M}_g\mathbf{R}_m) e^{2\pi i \mathbf{s}(\mathcal{M}_g\mathbf{T}_m + \mathcal{T}_g)}$$

2.1.4 Mesure de qualité du recalage

Maintenant que la densité électronique de l'assemblage ρ^{ass} et F^{ass}, sa transformée de Fourier, sont définies, il est possible de s'intéresser à la mesure de la qualité du recalage. Cette mesure consiste en une comparaison de la densité électronique de la reconstruction, notée ρ^{em}, avec la densité de l'assemblage ρ^{ass} calculée. Pour effectuer cette comparaison l'écart quadratique normalisé est utilisé. Formulé dans l'espace réel il est exprimé comme suit :

$$Q = \frac{\int [\rho^{em}(\mathbf{r}) - \lambda\rho^{ass}(\mathbf{r})]^2 d^3\mathbf{r}}{\int \rho^{em}(\mathbf{r})^2 d^3\mathbf{r}}$$

où l'intégration est étendue à tout le volume EM et où λ est un facteur d'échelle pour tenir compte du fait que les valeurs de la fonction de densité ρ^{em} sont données dans une unité arbitraire.

En utilisant le théorème de Parseval la formule peut être réécrite en termes de variables appartenant à l'espace réciproque :

$$Q = \frac{\int |F^{em}(\mathbf{s}) - \lambda F^{ass}(\mathbf{s})|^2 d^3\mathbf{s}}{\int |F^{em}(\mathbf{s})|^2 d^3\mathbf{s}}$$

où F^{em} désigne la transformée de Fourier de ρ^{em}.

En ce qui concerne le calcul lui même, l'espace réciproque, contrairement à l'espace réel, permet une intégration aisée de la symétrie et donc l'usage d'un grand volume, avec rapidité. L'espace de Fourier permet également de gérer des changements de résolution avec facilité.

Cette dernière formulation décrit exactement le critère implémenté dans le protocole d'optimisation de $\vee\in\supset\wedge$. Pourtant, le critère affiché dans le logiciel pour décrire la qualité du recalage est la corrélation (CC). Toutefois, chercher à minimiser l'écart quadratique est strictement équivalent à maximiser le coefficient de corrélation. Ce dernier, exprimé en pourcentage, peut être formulé de la manière suivante :

$$CC = 100 \times \frac{\int \overline{F^{em}(\mathbf{s})} F^{ass}(\mathbf{s}) d^3\mathbf{s}}{\sqrt{\int |F^{em}(\mathbf{s})|^2 d^3\mathbf{s} \int |F^{ass}(\mathbf{s})|^2 d^3\mathbf{s}}}$$

où $\overline{F^{em}}$ désigne le complexe conjugué de F^{em}.

En plus du coefficient de corrélation une autre mesure appelée “facteur R” est calculée. Il est exprimé comme suit :

$$R = \frac{\int ||F^{em}(\mathbf{s})| - |F^{ass}(\mathbf{s})||d^3\mathbf{s}}{\int |F^{em}(\mathbf{s})|d^3\mathbf{s}}$$

Ce facteur R, spécifique à l'espace réciproque et largement utilisé en cristallographie, permet de donner une indication complémentaire au coefficient de corrélation sur l'accord existant entre le modèle d'assemblage et la reconstruction. Dans le cas précis du recalage ce facteur est notamment utile pour déterminer la borne supérieure de l'intervalle de résolution définissant la sélection des coefficients de Fourier de la TF de la carte. En effet, lorsque la résolution maximale choisie est surestimée par rapport à la résolution réelle de la reconstruction de ME, le facteur R augmente fortement.

2.1.5 Stratégie d'optimisation des variables de position

Lorsqu'on peut extraire un volume minimal de la reconstruction de ME, ou dans le cas du recalage d'une seule molécule indépendante dans un assemblage symétrique, il est possible d'effectuer une recherche exhaustive de position en rotation et en translation. Dans tous les autres cas (grand volume, plusieurs molécules indépendantes) les positions des molécules sont déterminées par une technique d'optimisation, partant de positions initiales convenablement choisies. La notion de positions initiales convenables est malheureusement souvent définie a posteriori (après convergence).

Par nature l'optimisation conduit à une multiplicité de solutions, chacune correspondant à un optimum local qui dépend fortement des positions de départ des molécules. Souvent une simple inspection visuelle des positions moléculaires résultant de l'optimisation permet d'écarter des solutions “visiblement” fausses, comme dans les cas de superposition de molécules adjacentes ou lorsque les solutions sont en contradiction avec des indications fiables concernant l'assemblage. L'assignation de positions initiales des molécules, par une procédure manuelle et visuelle, permet de contrôler que les points de départ ne soient pas aberrants, évitant ainsi la convergence vers de fausses solutions.

2.2 Introduction de la flexibilité

Une des limitations de la méthode de recalage en corps rigide provient du fait que les molécules constituant les assemblages macromoléculaires ne correspondent pas toujours aux structures moléculaires trouvées dans les cristaux.

En effet, les protéines complexées présentent souvent des changements conformationnels importants par rapport à leur structure isolée[12].

Cependant, on peut s'affranchir de cette limitation en étudiant les mouvements des molécules dans le cadre d'une approximation dite harmonique (ou approximation des petits déplacements). Cette dernière permet de décrire le mouvement de chacun des atomes d'un système comme une combinaison de modes de vibration indépendants les uns des autres[2].

Ces modes de vibration dits normaux sont des modes dans lesquels tous les atomes d'une molécule vibrent à la même fréquence mais dans des directions ou avec des amplitudes différentes en passant simultanément par leur position d'équilibre. Au cours d'une vibration le centre de gravité de la molécule reste inchangé[13].

Ces mouvements de basse fréquence des macromolécules tels qu'on peut les calculer via l'approximation harmonique, ressemblent souvent à des mouvements fonctionnels observables expérimentalement. Ce résultat très robuste a été obtenu, tout d'abord en partant d'une description des molécules à l'échelle atomique, puis, plus récemment, de descriptions à "gros grains" (modèles de type réseau élastique)[14]. En s'appuyant sur ce constat, il est possible d'améliorer notablement la qualité du recalage de structures à l'intérieur des reconstructions de ME.

2.2.1 Modèle de réseau élastique

Pour simuler les mouvements de basse fréquence des modèles moléculaires, ces derniers sont décrits sous forme de réseau moléculaire élastique à gros grains. Les acides aminés de la structure sont représentés par un pseudo-atome en position du C^{α} [3].

Les interactions entre acides aminés sont modélisées par des ressorts harmoniques. Une matrice de voisinage (aussi appelée matrice de connectivité), calculée en fonction d'une longueur seuil (appelée cutoff), permet de déterminer quels pseudo-atomes sont reliés par ces ressorts. Chacun des ressorts possède une constante de force identique et est supposé être détendu dans la conformation de référence du modèle moléculaire.

3. Le C^{α} est l'atome de carbone central des acides aminés

2.2.2 Analyse en modes normaux

En ne s'intéressant qu'aux petits mouvements des atomes d'un modèle moléculaire élastique au voisinage d'une configuration d'équilibre, il est possible d'approximer l'expression de son énergie potentielle en une forme quadratique. Dans ce cas les équations du mouvement ont une solution analytique de la forme :

$$x_i(t) = x_i^o + \frac{1}{\sqrt{m_i}} \sum_{k=1}^{3N} C_k a_{ik} cos(2\pi\nu_k t + \Phi_k)$$

- où m_i est la masse associée à x_i, la i-ième coordonnée du modèle moléculaire,

- où C_k, Φ_k et ν_k sont respectivement l'amplitude, la phase et la fréquence du mode de vibration k,

- et où a_{ik} est le i-ième élément du k-ième vecteur propre de la matrice H, appelé Hessien, qui contient les dérivées partielles secondes de l'énergie potentielle, ici pondérées par les masses atomiques. L'élément ij s'écrit :

$$H_{ij} = \frac{\partial^2 V}{\sqrt{m_i m_j}\partial x_i \partial xj}$$

La diagonalisation du Hessien est la première étape dans l'obtention de solutions.

Le hessien est une matrice réelle symétrique. Il est toujours possible de la diagonaliser lorsque sa taille n'est pas trop grande, c'est-à-dire lorsque l'on peut la stocker dans la mémoire d'un ordinateur. Les algorithmes les plus courants permettent de le faire avec un coût en calcul proportionnel à nN^2, où N est le nombre de coordonnées d'atomes et n est le nombre de valeurs et de vecteurs propres que l'on souhaite déterminer.

Une fois la matrice diagonale obtenue, il est possible de déterminer les fréquences de vibrations du modèle étudié par ses valeurs propres (3N valeurs de fréquences). Il est ensuite possible de calculer de nouvelles coordonnées q (dites coordonnées normales) par combinaisons linéaires des écarts aux positions d'équilibre avec les éléments des vecteurs propres. La k-ième coordonnée normale s'exprime alors :

$$q_k = \sum_{i}^{3N} a_{ik}\sqrt{m_i}(x_i - x_i^o)$$

2.2.3 Construction de modèles moléculaires flexibles

La démarche est ensuite de construire, pour un mode de vibration donné, un ensemble de structures déformées, appelées conformères. Ces conformères correspondent aux coordonnées initiales auxquelles sont rajoutées les coordonnées normales du mode considéré multipliées par des amplitudes obtenues par une procédure d'optimisation inspirée du logiciel NORMA[3](voir section 3.2.7).

2.3 Graphisme, calcul et interactivité

L'interface visuelle de nombreux développements scientifiques est implémentée, bien souvent, dans l'unique but de simplifier les interactions entre l'utilisateur et l'exécution des modules de calcul, dans une optique de confort d'utilisation. L'environnement graphique développé dans le cadre de $\vee\in\supset\wedge$ est différent. L'interaction des graphismes 2D (menu, bouton, etc.) et de la partie de visualisation 3D (représentation polygonale des molécules, des cartes de densité, etc.) avec les modules de calcul est, en effet, fondamentale au processus de recalage.

La principale exigence de la méthode de recalage consiste dans le fait que l'utilisateur doit pouvoir continuellement contrôler la qualité des résultats. Il est donc nécessaire de pouvoir :

- afficher les molécules constituant l'assemblage,
- récupérer les variables de position des molécules indépendantes,
- transmettre ces dernières à la partie calculatoire.

Cette dernière va, à son tour, renvoyer de nouvelles positions, qu'il faudra assigner à l'intérieur de l'environnement graphique, de manière à ce que l'utilisateur puisse contrôler visuellement le résultat.

Une deuxième spécification importante est, pour l'utilisateur, de pouvoir facilement intégrer ses connaissances concernant la structure des assemblages à l'intérieur du processus de modélisation et ce quelque soit la technique utilisée :

- durant le processus d'optimisation, il faut que l'utilisateur ait la possibilité, avant un affinement, d'orienter la convergence des solutions par la spécification de positions initiales ayant, d'après lui, du sens.

- durant un processus de recalage manuel, il faut que l'utilisateur ait la

possibilité de connaître l'évolution de la qualité du recalage, en temps réel, alors qu'il déplace les molécules dans l'espace de la reconstruction.

L'interactivité graphisme/calcul requise pour répondre à l'ensemble de ces spécifications, combinée avec la performance des algorithmes de recherche, font aujourd'hui de $\vee\in\supset\wedge$ un espace d'expérimentation complet et intuitif.

Bien que les aspects de graphisme et d'interactivité ne puissent être formulés sous forme mathématique, ils forment cependant une composante essentielle de la méthode. L'environnement graphique de $\vee\in\supset\wedge$, décrit du point de vue de ses fonctionnalités dans les sections 3.2 et 3.3, a été développé comme un "wrapper". Il est conçus de manière à encapsuler de nombreux composants différents, tout en permettant à l'utilisateur d'interagir avec chacun d'eux de manière cohérente.

Cet environnement repose principalement sur deux bibliothèques logicielles :

- La bibliothèque **VTK**[15] (www.vtk.org), "open-source", gratuite, est dédiée à la visualisation, aux graphismes 3D et au traitement d'images ...
- La bibliothèque **Tkinter**[16], interface de programmation graphique standard pour le langage Python, permet la gestion des interfaces graphiques 2D.

Ces deux bibliothèques, conçues pour être compatibles entre elles, ont été combinées pour créer tous les composants nécessaires à l'accomplissement des spécifications, liées aux graphismes et à l'interaction, indispensables au processus de recalage.

L'environnement graphique et interactif de $\vee\in\supset\wedge$ est bien plus qu'une surcouche des exécutables qu'il empaquette. Certains de ces exécutables, déjà inclus dans le logiciel URO, préexistaient avant le développement de l'environnement de $\vee\in\supset\wedge$, alors que d'autres ont été développés spécifiquement pour être utilisés conjointement à l'environnement graphique.

Le développement de la partie de recalage interactif en temps réel (voir section 3.2.5) est typiquement un bon exemple de réalisation impliquant à la fois de la programmation bas niveau et des besoins en termes d'interactivité et de graphisme.

Cette partie complexe à mettre en œuvre a nécessité la conception d'un protocole permettant d'établir la communication entre deux processus, écrits

dans des langages différents et s'exécutant en parallèle.

Même si la plupart des exécutables ont été conçus pour effectuer des taches très spécifiques, ils forment néanmoins, avec l'environnement graphique, une unité indissociable.

Chapitre 3

Développement

$\vee\in\supset\wedge$ est un programme conséquent, comportant plus de 25000 lignes de codes. Une attention toute particulière a donc été portée à l'architecture de ce logiciel. Bien que ce chapitre n'ait pas vocation à expliquer en détail cette architecture, il se propose tout de même, dans une première partie, d'en présenter quelques idées maîtresses.

FIGURE 3.1 – Logo du logiciel $\vee\in\supset\wedge$

3.1 Architecture du logiciel

Le logiciel $\vee\in\supset\wedge$ a été développé principalement en utilisant deux langages :

– la partie de plus haut niveau, développé en langage Python (programmation objet), gère les données tant au niveau de leur traitement que de leur affichage.

– au plus bas niveau, une partie constituée d'un ensemble d'exécutables binaires est implémentée en langage FORTRAN. Elle effectue certains traitements comportant les calculs les plus lourds.

L'architecture du composant Python est organisée selon 7 modules principaux à l'intérieur desquels sont répartis les données et les traitements.
Afin d'atteindre la qualité optimale du logiciel, la composition de ces modules doit répondre à un ensemble de critères.

Parmi l'ensemble des critères, les plus importants sont :

– la **validité** : le logiciel effectue exactement les tâches pour lesquelles il a été conçu.

– l'**extensibilité** : il intègre facilement de nouvelles spécifications.

– la **réutilisabilité** : le code source du logiciel est complètement ou en partie réutilisable. Ceci impose lors de la conception une attention particulière à l'organisation du logiciel et à la définition de ses composants

– la **robustesse** : le logiciel peut fonctionner même dans des conditions non prévues lors de la conception. Ce critère est atteint si le logiciel est capable de détecter qu'il se trouve dans une situation anormale.

Il est difficile de respecter ces critères lorsque l'architecture d'un logiciel est monolithique. En effet, dans ce cas le moindre changement de spécification peut avoir des répercussions très importantes sur le logiciel, imposant une lourde charge de travail pour effectuer les mises à jour.

C'est pourquoi $\vee\in\supset\wedge$ a adopté une architecture flexible basée sur plusieurs modules. Le programme peut ainsi être vu comme un ensemble d'entités indépendantes entretenant des relations entre elles. L'intérêt de ce type d'architecture est de concentrer les informations liées à un type de données ou de traitement à l'intérieur d'un module qui est seul habilité à exploiter ces informations.

3.1.1 Méthodes de conception

La définition des modules de $\vee\in\supset\wedge$ a suivi en parallèle deux approches communément admises[17] :

– **La méthode descendante** : procède par décomposition des problèmes. Un problème est ainsi divisé en un certain nombre de sous-problèmes, chacun de complexité moindre. Cette division est ensuite appliquée aux sous-problèmes générés, et ainsi de suite récursivement, jusqu'à ce que chacun des sous problèmes soit trivial.

– **La méthode ascendante** : procède, à l'inverse, par composition de briques logicielles simples, pour obtenir une procédure répondant à un problème donné. Cette approche a été spécifiquement mise en œuvre dans le développement de ∨∈⊃∧ lors de l'utilisation de certaines bibliothèques logicielles.

La combinaison de ces deux approches a permis, dans un premier temps, une conception assez intuitive des modules. Dans un second temps, ces méthodes ont simplifié la partie de codage rendue difficile par la grande complexité des relations que les modules entretiennent entre eux.

3.1.2 Contenu des modules

D'un point de vue général, l'architecture de ∨∈⊃∧ peut être divisée en 3 couches : une couche gérant le rendu graphique 3D, dite couche graphique, une couche de pilotage, combinaison de l'interface homme-machine (2D) et de la partie d'interaction avec les objets 3D, et enfin une couche dédiée aux calculs lourds, appelée couche de calcul.

Ces trois couches sont organisées en plusieurs modules (voir figure 3.2) présentés brièvement ci dessous. Cette architecture a permis de construire un logiciel stable, qui répond bien aux objectifs de définis.

∨∈⊃∧ est un programme complexe. Pour que son code reste compréhensible et que les interactions entre les modules soient organisées à l'intérieur du programme, il est important que la communication intra-modulaire soit contrôlée avec attention.

Pour ce faire, un module central appelé médiateur a été créé selon le patron de conception du même nom. Cette conception résout le problème de dépendances entre les modules en permettant au module médiateur de jouer le rôle d'interface entre les traitements et les données contenues à l'intérieur

de modules différents. Ainsi lorsqu'une méthode [1] doit interagir avec le contenu d'autres modules que le sien, elle doit passer par le médiateur qui se charge de transmettre l'information aux modules concernés.

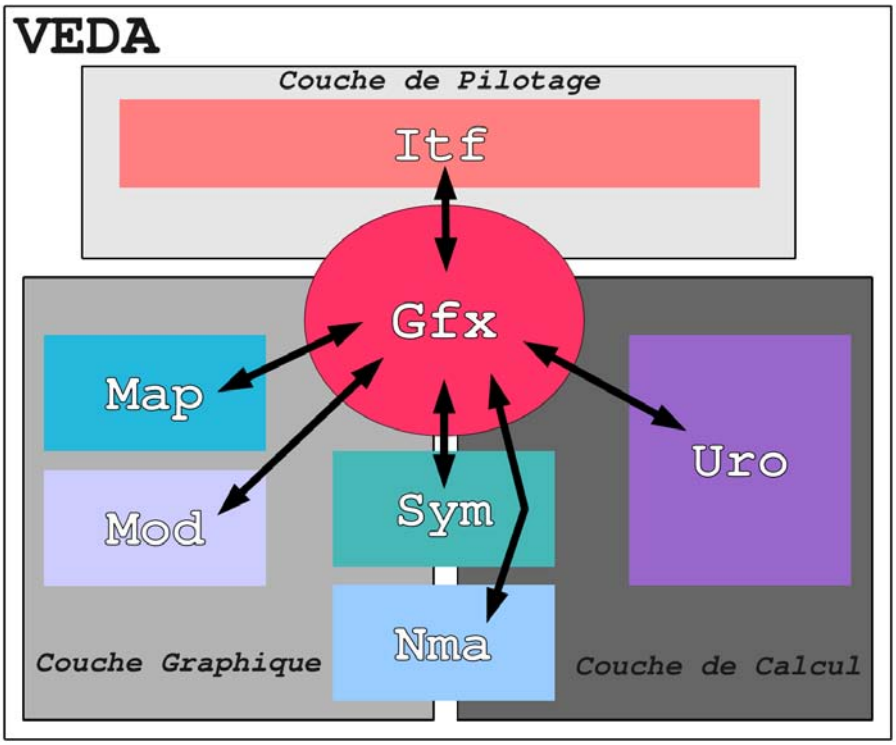


FIGURE 3.2 – Patron de conception du logiciel ∨∈⊃∧ : une architecture modulaire

Ce modèle de conception permet l'intégration de petites variations dans les spécifications n'entraînant qu'un nombre limité de modifications au sein d'un petit nombre de modules, évitant ainsi la remise en cause des relations liant les modules entre eux. Cela a pour conséquence de diminuer l'impact potentiel des modifications sur le reste du logiciel en confinant les erreurs potentiellement introduites dans les modules, là où elles sont apparues (ou éventuellement dans un nombre restreint de modules).

Voici brièvement le contenu des modules de ∨∈⊃∧ :

1. En programmation orientée objet, une méthode désigne une fonction propre à un objet

– Itf : implémentation de l'interface homme machine (fenêtre, menu, bouton...).

– Map : traitement et représentation des cartes de densité électronique et procédures leur étant attachées (chargement, affichage, redimensionnement ...).

– Mod : implémentation des types de modèles, molécules, molécules symétriques et conformères.

– Sym : gestion de la symétrie, calcul des listes d'opérations, affichage des supports de symétrie, calcul des constellations.

– Uro : recalage interactif, affinement automatique, et outils d'analyse (recherche du grossissement optimal de la reconstruction, statistique de convergence, exploration locale du profil de corrélation)

– Nma : calcul des modes normaux, visualisation des séquences de conformères, remplacement in situ, recherche en amplitude.

– Gfx : médiateur de classes. Gère également tous les aspects de la visualisation 3D (fenêtre de rendu, mapping, moteur de rendu).

Pour plus de détails concernant l'architecture logicielle de $\vee\in\supset\wedge$, il est possible de se référer à l'annexe 6.1.

3.2 Environnement graphique

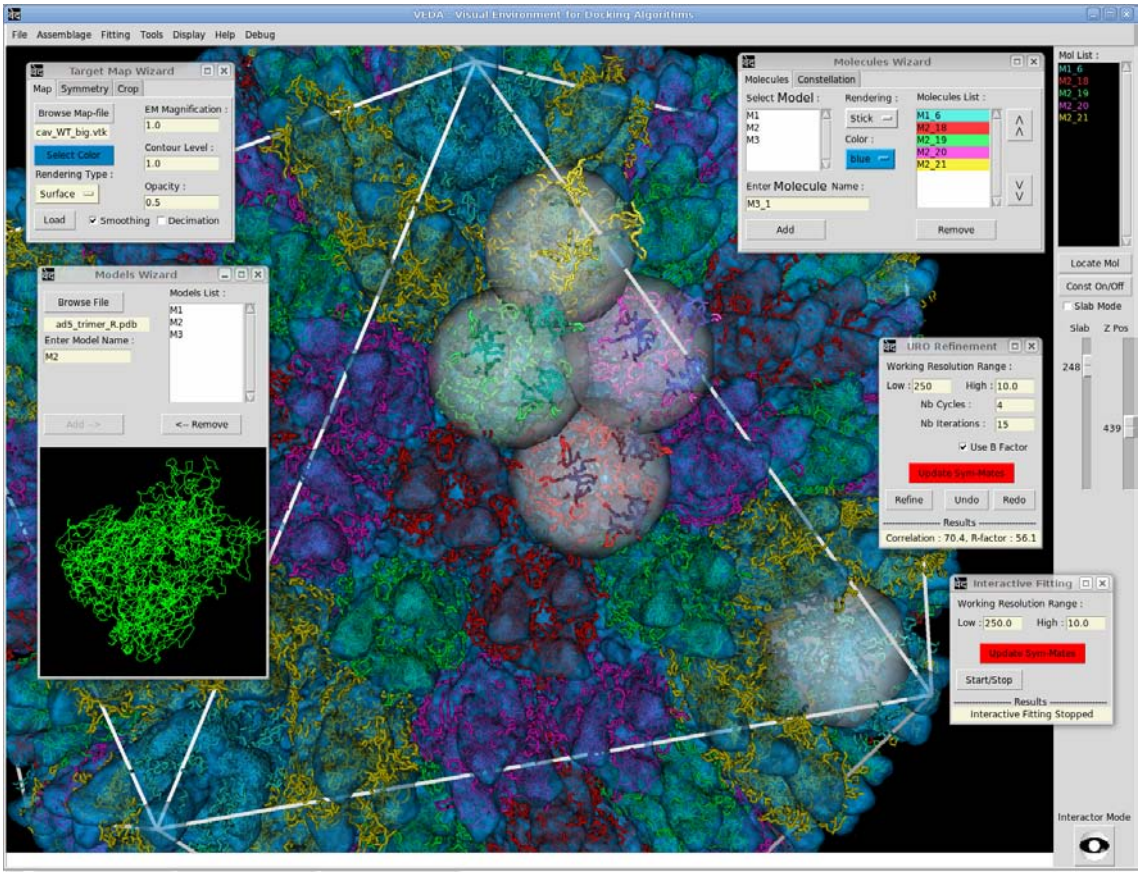


FIGURE 3.3 – Capture d'écran du logiciel $\vee\in\supset\wedge$

L'aspect graphique de $\vee\in\supset\wedge$ est l'une des plus fortes spécificités du programme. Cette couche graphique a pour but premier d'afficher les molécules constituant l'assemblage, de récupérer les variables de position des molécules indépendantes et de les transmettre à la partie calculatoire. Conçu à l'origine comme sur-couche du "package" de recalage URO[4] , l'interface graphique de $\vee\in\supset\wedge$ s'est au fil des développements grandement enrichie.

Désormais véritable environnement dédié au recalage, la couche graphique de $\vee\in\supset\wedge$ est dotée d'une interface homme-machine (IHM) complète permettant de gérer les données et de piloter l'ensemble des fonctionnalités facilement (voir figure 3.3).

La toute première utilité de l'interface homme machine est de permet-

tre l'entrée des données à l'intérieur de $\vee\in\supset\wedge$. L'exécution du logiciel n'est pas linéaire, il n'y a pas de chemin d'exécution type. Toutefois, certaines actions nécessitent des pré-requis (chargement de données, chargement de la symétrie, génération de la constellation, setup ...). Si l'utilisateur omet une ou plusieurs étapes pré-requises, le programme lui indiquera la marche à suivre avant de pouvoir effectuer l'action ayant provoquée l'apparition du message. La liberté d'action à l'intérieur de $\vee\in\supset\wedge$ est grande. Un utilisateur débutant, éventuellement déstabilisé, pourra alors suivre un protocole "standard".

Ce protocole est composé de 7 étapes :

1. Chargement d'une carte cible et configuration de la symétrie
2. Chargement des modèles
3. Construction de l'assemblage moléculaire
4. Setup : préparation des données pour le recalage
5. Procédure de recalage interactif et/ou d'affinement automatique
6. Sauvegarde des molécules et/ou des constellations
7. Déformation élastique des modèles moléculaires

Pour effectuer ce protocole standard, il suffit de se rappeler d'une règle simple conditionnant le parcours des menus et des onglets à l'intérieur des sous-fenêtres de l'IHM : "aller toujours de haut en bas et de gauche à droite".

L'environnement graphique de $\vee\in\supset\wedge$ est convivial. Lors de chacune des étapes du protocole ci-dessus, une sous-fenêtre de l'IHM est spécifiquement dédié à la réalisation de la tâche en cours. Ces fenêtres, appelées "wizards", ont pour rôle de guider l'utilisateur tout au long de l'utilisation de $\vee\in\supset\wedge$. Le premier contact de l'utilisateur avec un wizard a lieu lors du chargement et de la configuration des données sur lesquelles sera effectué le recalage.

3.2.1 Chargement d'une carte cible et configuration de sa symétrie

$\vee\in\supset\wedge$ gère uniquement les cartes de densité électronique 3D dans un format ASCII appelé EZD "easy density". Pour toutes les conversions depuis d'autres formats (PROTEIN, FFT-Y, TENEYCK2, CCP4, X-PLOR, MASK, BINXPLOR, BRICK, DSN6, 3DMATRIX, TNT, PHASES, FSMASK, BRIX, XPLOR, CNS, EM08, OMAP, MPI et AMBER) le programme MAPMAN [18] peut être utilisé simplement.

Le format EZD contient une partie d'en-tête dans laquelle est défini un ensemble d'attributs propres à la carte. Tout le reste du fichier est consacré au stockage des valeurs de densité correspondant à un volume échantillonné. Ces valeurs sont stockées sous la forme d'un simple tableau unidimensionnel.

Chargement d'une carte cible

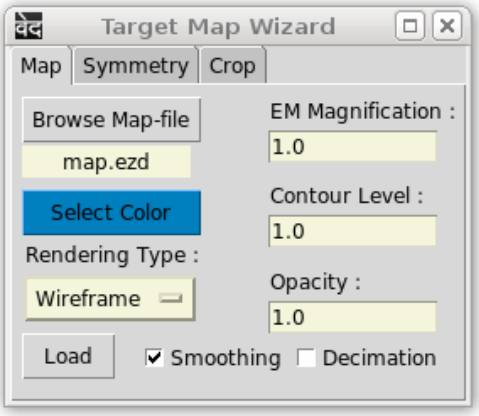
Le chargement de la carte cible est effectué grâce au wizard "Target Map" (voir figure 3.4a). Il commence par la recherche d'un fichier sur le disque dur, se poursuit par la définition d'un ensemble de paramètres ("magnification", niveau de contour, mode de représentation, ...) et se termine lorsque le bouton "Load" du wizard est actionné. Lors de cette action, la procédure de lecture du fichier de carte est lancée tandis que l'ensemble des attributs, liés au paramétrage, est enregistré dans la mémoire.

La lecture de la carte consiste en l'indexation des valeurs de densité à l'intérieur d'une grille tridimensionnelle. L'espacement de cette grille, appelé spacing, est déterminé expérimentalement par le microscopiste. Les coordonnées entières de la grille, données en pixel, sont alors converties, grâce au spacing [2], en angström. Chaque valeur de densité est alors associée à un point dans l'espace de la reconstruction.

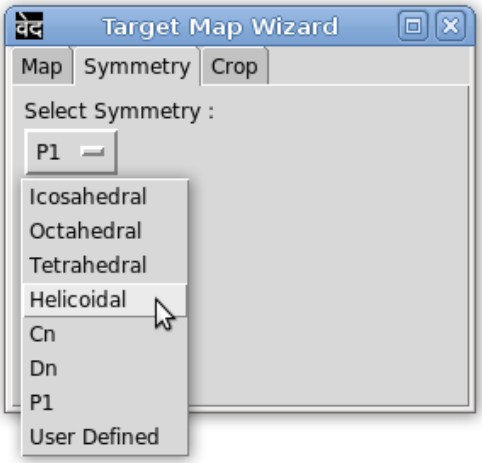
Une fois que toutes les valeurs de densités sont chargées en mémoire principale, une procédure calcule l'approximation d'une surface occupant une valeur de densité constante (niveau de contour) spécifiée par l'utilisateur. Cette iso-surface, calculée grâce à l'algorithme de "marching cube" implémenté dans VTK est finalement représentée en 3D par le moteur graphique de $\vee\in\supset\wedge$, à l'intérieur de la fenêtre principale de rendu (appelée par la suite "espace de travail") .

Le chargement d'une carte 3D ne s'arrête pourtant pas à l'affichage de l'iso-surface. Dans le cas d'une reconstruction symétrique le paramétrage continue, au travers de la sélection et de la configuration de la symétrie intrinsèque à la reconstruction.

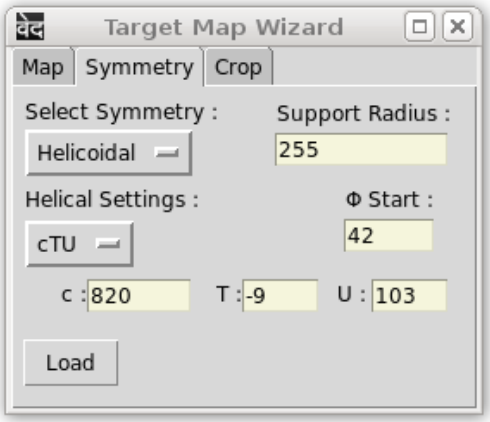
2. le spacing est intrinsèquement lié au grossissement de la carte de densité, or le grossissement n'est pas toujours déterminé avec précision ($\sim$ 5% de marge d'erreur). C'est pourquoi il fait l'objet d'une procédure d'ajustement (voir section 3.3.2).



(a) Chargement d'une carte cible



(b) Sélection d'une symétrie



(c) Configuration d'une symétrie

FIGURE 3.4 – "Target Map Wizard"

Configuration de la symétrie

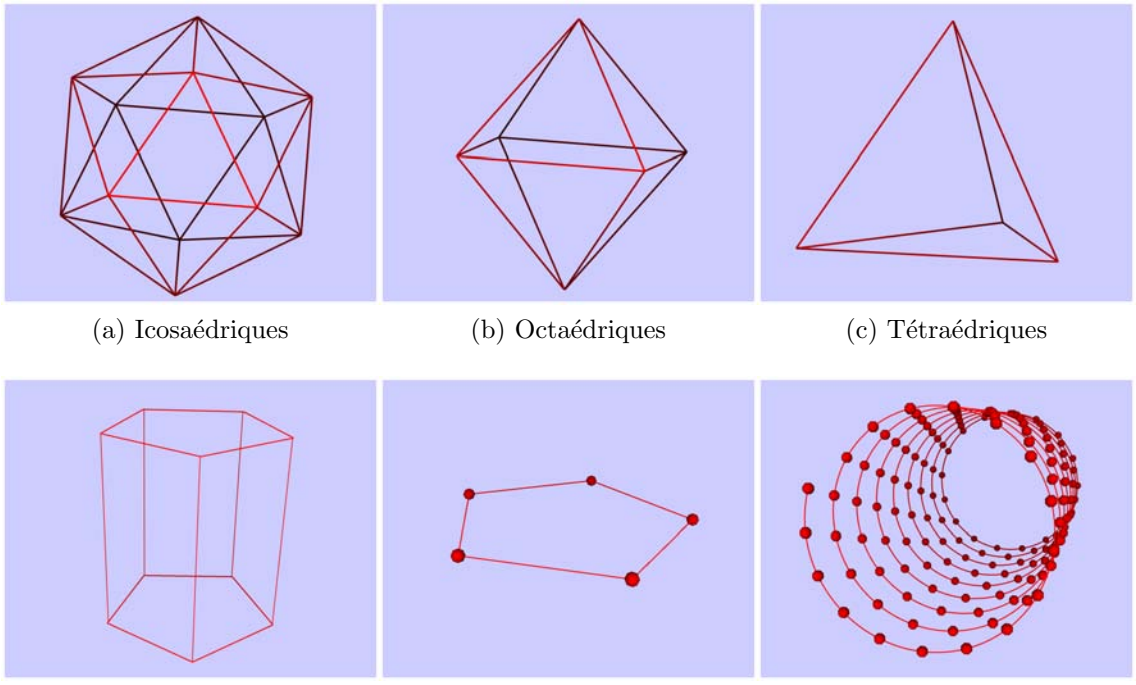
La gestion de la symétrie est l'une des spécificités les plus intéressantes de ∨∈⊃∧. Tirer partie de la symétrie des assemblages permet de simplifier grandement les calculs en limitant le nombre de molécules pour lesquelles on doit déterminer les variables de position.

Par exemple, dans le cas d'un assemblage de symétrie icosaédrique, l'imposition de la symétrie dans le recalage permet de diviser le nombre de

molécules nécessaires (et donc le nombre de paramètres à estimer) par 60.

Le paramétrage de la symétrie est effectué dans un onglet supplémentaire du wizard “Target Map” (voir figure 3.4b). Au départ le seul élément présent dans la fenêtre est un menu déroulant contenant le nom des symétries gérées par $\vee \in \supset \wedge$ (toutes les symétries ponctuelles (icosaédriques, octaédriques, tétraédriques, cycliques, diédriques, P1) ainsi que la symétrie hélicoïdale). Lorsqu'une symétrie est sélectionnée dans la liste, un ensemble de composants permettant la configuration de la symétrie choisie apparaît dans la fenêtre (voir figure 3.4c).

Chacune des symétries est associée à un support conçu pour la représenter visuellement et pour permettre de vérifier si la configuration des paramètres utilisés est en adéquation avec la carte de densité.



(a) Icosaédriques (b) Octaédriques (c) Tétraédriques

(d) Diédriques (e) Cycliques (f) Hélicoïdales

FIGURE 3.5 – Supports représentatifs des symétries.

Par exemple, la symétrie icosaédrique est représentée par un icosaèdre (voir figure 3.5a) qui change d'orientation pour chacune des 26 configurations différentes de cette symétrie. Ces 26 options représentent toutes les

combinaisons d'alignement d'au moins deux axes de symétrie 2,3,5 et les axes X, Y, Z du repère de référence.

Autre exemple, la symétrie hélicoïdale a pour support graphique une hélice continue dont certains points représentés par des sphères symbolisent les sous-unités (voir figure 3.5f). Plusieurs paramétrisations sont possibles pour cette symétrie :

- **C,T,U** : où **C** est la longueur du "repeat" [3] de l'hélice élémentaire, **T** le nombre de tour d'hélice élémentaire à l'intérieur du "repeat" et **U** le nombre de sous unités à l'intérieur du "repeat".

- $\boldsymbol{\Delta\Phi, \Delta z, \#s}$: où $\boldsymbol{\Delta\Phi}$ est le décalage angulaire entre deux unités, $\boldsymbol{\Delta z}$ le décalage en z entre deux unités et $\boldsymbol{\#s}$ le nombre de départ d'hélice(s).

Une fois la symétrie configurée, un fichier (appelé "SYM") contenant toutes les opérations du groupe de symétrie [4] est généré. Au même moment un support graphique représentant la symétrie apparaît dans l'espace de travail de ∨∈⊃∧ (voir figure 3.5).

3.2.2 Chargement des modèles

En ce qui concerne les modèles atomiques, ∨∈⊃∧ utilise des fichiers au format standard PDB [19]. Issus de la Protein Data Bank, les fichiers PDB contiennent les coordonnées atomiques issues de techniques expérimentales comme la MX ou la RMN. Les fichiers PDB ne contiennent pas uniquement des coordonnées. Ils contiennent également d'autres sections qui renseignent le nom des molécules décrites par le fichier, des informations concernant les structures primaires et secondaires, etc.

3. Le "repeat" est la plus petite partie de l'hélice possédant une symétrie translationnelle pure

4. ou du moins une partie suffisante en ce qui concerne la symétrie hélicoïdale

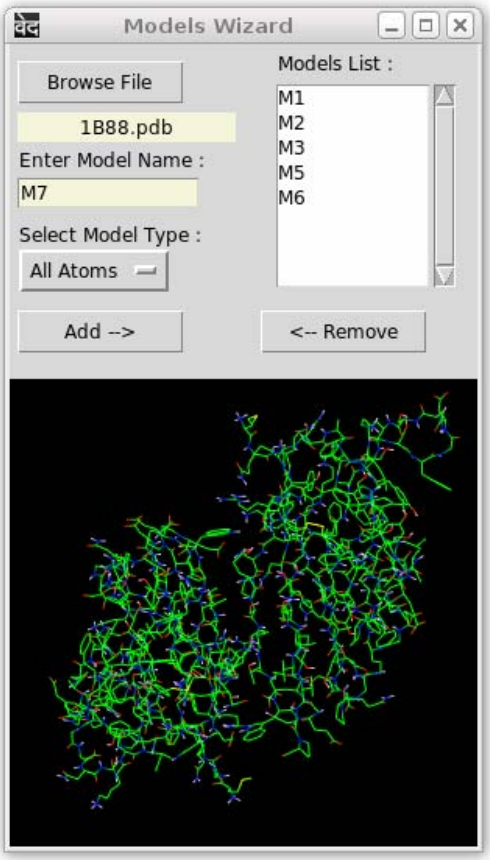


FIGURE 3.6 – "Models Wizard"

$\vee\in\supset\wedge$ utilise uniquement les données contenues dans les lignes du fichier correspondant à l'attribut "ATOM". Chacune de ces lignes contiennent :

- Des caractères (C, H, N, Mg, ...) définissant un type d'atome en nomenclature Mendeleïev.

- Les coordonnées cartésiennes décrivant la position de l'atome dans un référentiel particulier sous forme de 3 nombres réels.

- Le facteur B, décrivant l'agitation thermique de l'atome en question, sous la forme d'un nombre réel.

Dans le cas du chargement des modèles atomiques, la procédure est assez similaire à celle de chargement de la carte (sans la partie concernant la

symétrie). Une fenêtre appelée “Models Wizard” (voir figure 3.6) permet d'effectuer une recherche de fichier PDB, puis de spécifier un identificateur et un type de modèle (tout atome, “backbone” [5], C^{α}).

Lorsque le bouton “Load” du wizard est actionné, le modèle est enregistré dans sa position de référence, comme expliqué dans la section 2.1.1. Le nom identifiant le modèle apparaît ensuite dans une liste alors qu'une représentation 3D de la structure (en accord avec le mode de représentation choisi) est affichée dans la fenêtre de rendu intégrée au wizard .

3.2.3 Construction de l'assemblage moléculaire

Ajout des molécules indépendantes

Une fois que la reconstruction 3D (et sa symétrie associée) ainsi que les modèles sont chargés, il est possible de générer les molécules dans l'espace de travail de $\vee\in\supset\wedge$. Encore une fois, un wizard est présent (voir figure 3.7a). Cette fenêtre appelé, “Molecules Wizard” permet de sélectionner dans une liste un des modèles chargés précédemment.

Lors de cette sélection, une procédure assigne à la molécule un identificateur, un type de rendu et une couleur de manière automatique dans le but d'accélérer ce paramétrage. Bien sûr, si l'utilisateur veut spécifier lui même ces attributs, il lui suffit de modifier les valeurs définies par défaut avant d'actionner le bouton “ADD”. Une fois les molécules ajoutées, leurs identificateurs associés à leurs couleurs apparaissent alors dans la fenêtre principale, ainsi que dans certains wizards, sous forme d'un item à l'intérieur d'une liste.

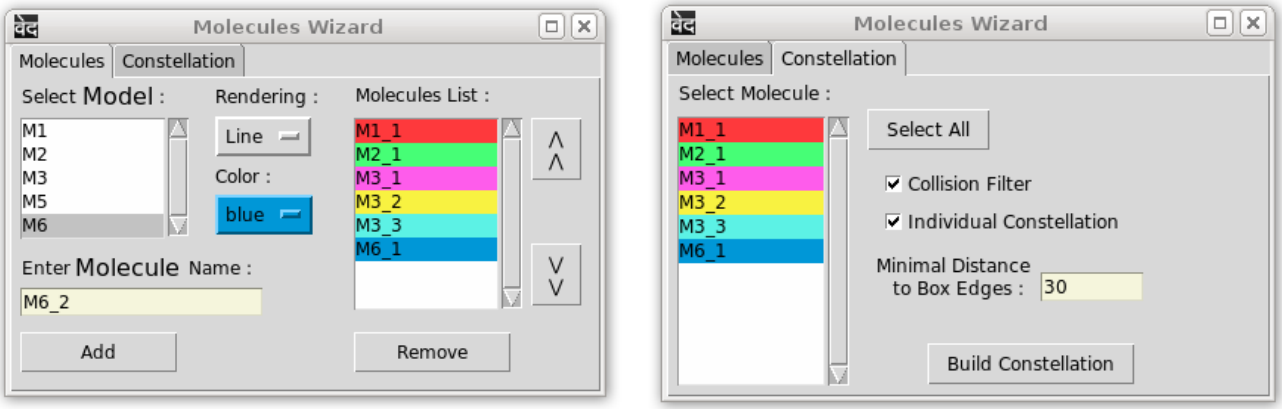
L'apparition de ces items dans les listes de molécules coïncide avec l'affichage dans l'espace de travail de $\vee\in\supset\wedge$ d'une représentation polygonale des molécules. Cette représentation graphique dépend d'une combinaison entre, le type de modèle dont la molécule est issue (tout atome, “backbone”, C^{α}) et le type de rendu associé à la molécule elle même [6] (lignes, sphère, bâtonnet, lignes et sphères, bâtonnet et sphère).

Les positions initiales des molécules ajoutées à l'espace de travail de $\vee\in\supset\wedge$

5. Le “backbone” est, pour un polymère, la série d'atomes liée de manière covalente, qui crée la continuité de la chaîne principale

6. A l'avenir, le nom molécule, s'appliquera aussi bien à l'objet 3D représentant la molécule, qu'à la molécule telle qu'elle a été définie dans la partie 2

sont les même qu'à l'intérieur du fichiers PDB d'où sont issus les modèles.



(a) Onglet molécules

(b) Onglet constellation

FIGURE 3.7 – Le "Molecules Wizard" permet de générer les molécules et leurs constellations

Placement des molécules à l'intérieur de la carte cible et construction de leurs constellations

L'étape consistant à générer l'ensemble des molécules symétriques associées à chaque molécule indépendante est cruciale pour la suite des événements. Bien que la définition de ces constellations soit simple (voir section 2.1.2), pratiquement, leur construction est plus délicate.

Les positions initiales des molécules indépendantes ajoutées à l'espace de travail de $\vee\in\supset\wedge$ ne sont pas nécessairement pertinentes pour le recalage. Afin de générer les constellations, il est nécessaire dans un premier temps, de replacer les molécules indépendantes à l'intérieur de la carte. Si aucune information a priori concernant l'assemblage ne permet d'inférer de "bonnes" positions de départ, et que la carte de densité est de trop basse résolution pour fournir quelques indices, le placement doit alors être effectué de manière à respecter au maximum 2 critères principaux.

– Les molécules indépendantes et symétriques ne doivent pas se chevaucher entre elles (y compris entre molécules indépendantes ou entre molécules

symétriques)

– Le volume ME doit être le plus rempli possible (si l'on cherche à reconstruire un assemblage correspondant à toute la densité)

A moins de bien connaître la symétrie de l'assemblage et la manière dont les opérateurs de symétrie sont sélectionnés, il est difficile d'estimer la position des molécules symétriques avant d'avoir généré les constellations.

Sélection des opérateurs de symétrie

Lors de la configuration de la symétrie, un fichier SYM contenant toutes les opérations de symétrie (ou du moins un nombre suffisant [7]) a été généré. Dans le cas hélicoïdal, ou si l'utilisateur travaille avec une symétrie ponctuelle sur une carte non entière, il est nécessaire de sélectionner pour chaque molécule indépendante, une liste d'opérateurs qui vont être utilisés pour générer les molécules symétriques.

La première étape de sélection des opérateurs de symétrie consiste à combiner chacun de ces opérateurs aux variables de position des molécules indépendantes. Si cette combinaison "envoie" le centre de masse d'une des molécules à l'intérieur du volume ME, alors l'opérateur en question est ajouté à une liste, appelée "symlist", spécifique à chaque molécule indépendante. La suite de la procé-dure consiste en un affinement successif de ces listes d'opérateurs. La liste des opérateurs de symétrie sélectionnés en fin de procédure donnera lieu à la création de molécules symétriques et sera donc impliquée dans le calcul des facteurs de structure de l'assemblage (voir section 2.1.3). La façon de construire ces "symlists" peut avoir un impact important sur le résultat final du recalage.

Dans l'onglet symétrie du "Molecules Wizard" (voir figure 3.7b) plusieurs options peuvent être configurées pour influencer la construction de ces listes d'opérateurs. Une option "independant constellation", activée par défaut, permet de spécifier si chaque molécule indépendante peut être associée, ou non, à différents "symlists".
La couche de calcul requiert une liste générale construite à partir de l'ensemble des "symlists" propres à chaque molécule indépendante. Le "symlist" général peut être basé soit sur l'union des "symlists" spécifiques, lorsque

7. Dans le cas d'une symétrie hélicoïdale, il est possible de générer une infinité d'opérations. Le remplissage du fichier SYM tient alors compte de la longueur de la reconstruction (le long de l'axe de symétrie) pour générer un set d'opérations suffisant.

l'option "indépendant constellation" est activée, soit sur leurs intersections.

Une autre option, appelée "collision filter" permet d'activer une procédure calculant les distances entre les positions des centres de masse des molécules symétriques potentielles [8]. Si les distances indiquent que plusieurs molécules symétriques potentielles sont en collision (distance inférieure à un seuil calculé au chargement du modèle), les "symlists" sont alors filtrés de manière à éviter ces collisions.

Création des molécules symétriques

Une fois les "symlists" créés et filtrés, vient le moment de construire les constellations. Pour chaque molécule indépendante et pour chaque opération du "symlist" spécifique à cette dernière, correspond une molécule symétrique.

Chaque molécule symétrique se voit associer une représentation graphique. Ces représentations sont identiques à celles des molécules indépendantes. Cependant, quelques différences les caractérisent :

- leur positionnement et leur orientation sont différents de la molécule indépendante dont elle est issue (sauf dans le cas de l'opérateur identité).

- la couleur associée à sa représentation graphique est également différente (dans le même ton mais en plus foncée) de manière à pouvoir la distinguer.

- dans l'espace de travail de $\vee\in\supset\wedge$ il est impossible d'influer directement sur le déplacement des molécules symétriques avec la souris, contrairement aux molécules indépendantes.

Pratiquement, cette dernière différence est la plus importante. Les positions et les orientations des molécules symétriques sont issues de la combinaison de leurs transformations intrinsèques (l'opérateur de symétrie) avec les variables de position de la molécule indépendante. A chaque mouvement d'une molécule indépendante, les variables de position des molécules symétriques issues de cette molécule indépendante sont recalculées en temps réel.

8. potentiellement associées aux opérations de symétrie sélectionnées une fois la procédure terminée

3.2.4 Préparation des données pour le recalage

Une fois les constellations générées, tous les ingrédients nécessaires à la construction d'un modèle pseudo-atomique de l'assemblage sont réunis. Cependant, avant de lancer la procédure d'optimisation des variables de position des molécules indépendantes, il est nécessaire de passer par une étape de préparation des données.

Comme indiquée dans la section 2.1.4, la mesure du critère d'optimisation est effectuée dans l'espace réciproque. Pour pouvoir calculer ce critère, il est donc nécessaire de disposer de F^{em}, la transformée de Fourier de la carte, ainsi que des facteurs de diffusion moléculaire f_μ obtenus par application de la transformée de Fourier[20, 21] aux densités électroniques ρ_μ calculées pour chaque modèle μ.

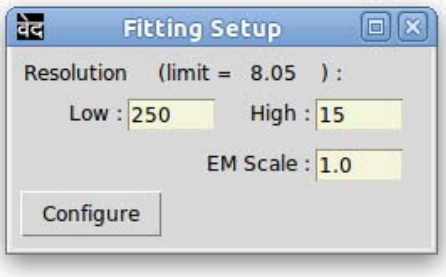


FIGURE 3.8 – "Fitting Setup"

Le "setup" des données se fait grâce à une petite fenêtre, appelée "Fitting setup", accessible via le menu dédié au recalage (voir figure 3.8). La première information donnée par cette fenêtre est la résolution maximum que l'utilisateur va pouvoir utiliser. Cette valeur, égale à deux fois le spacing, est disponible dans le menu à partir du moment où la carte est chargée.

Trois valeurs sont requises avant de pouvoir lancer la préparation des données au recalage : deux valeurs spécifiant un intervalle de résolution et un facteur de mise à l'échelle.

Les valeurs spécifiant l'intervalle de résolution vont conditionner le nombre maximum de coefficients de Fourier utilisables dans l'étape de recalage proprement dite. Sachant qu'il est possible de restreindre cet intervalle au moment de l'affinement, il est préférable, au moment du "setup", de définir

ces valeurs de résolution largement, de manière à ne pas avoir à revenir en arrière.

Le facteur de mise à l'échelle, "EM scale", est assez semblable à la magnification [9], à la différence qu'il a également une influence sur les paramètres de symétrie dans le cas hélicoïdal (le seul cas où les opérations de symétrie incluent des translations).

Le "setup" doit être relancé dans trois cas de figure :

- lorsque l'utilisateur désire augmenter l'intervalle de résolution,
- lorsque une molécule issue d'un nouveau modèle est ajoutée dans l'espace de travail de $\vee\in\supset\wedge$,
- lorsque la carte de densité est modifiée (chargement d'une nouvelle carte ou changement de magnification de la carte en cours).

Dans tous les cas la procédure d'affinement procède à une vérification avant de lancer les calculs. Si un changement nécessitant une nouvelle préparation des données est détecté, un message indiquant la marche à suivre apparaît.

3.2.5 Procédure de recalage interactif et/ou d'affinement automatique en corps rigides

Une fois la préparation des données effectuée, vient le moment de recaler les molécules. Deux procédures capables d'effectuer le recalage sont disponibles.

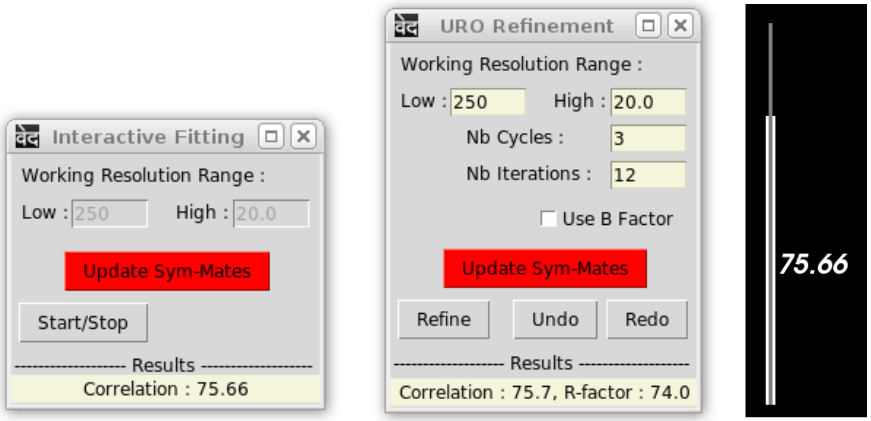
La première, appelée recalage interactif, fonctionne manuellement, utilisant les connaissances a priori des utilisateurs sur la structure des assemblages. La possibilité de recalage interactif permet de se faire une idée assez réaliste de la sensibilité des positions moléculaires par rapport au critère de recalage.

La seconde consiste en un ajustement automatique des positions moléculaires. Cet affinement est basé sur un protocole d'optimisation de type moindre carré inspiré d'une technique de remplacement moléculaire[22],

Ces deux procédures complémentaires peuvent fonctionner de concert. Une fois le recalage interactif lancé via un menu dédié (voir figure 3.9a),

9. facteur de grossissement de la reconstruction modifiable dans le "Target Map Wizard"

chaque fois que l'utilisateur modifie les variables de position d'une molécule indépendante dans l'espace de travail de $\vee\in\supset\wedge$, une corrélation est calculée en temps réel et représentée par une barre de longueur variable sur la gauche de la fenêtre principale de rendu (voir figure 3.9c). Lorsque l'utilisateur juge opportun d'affiner automatiquement les positions moléculaires, il lui suffit de presser le bouton "Refine" dans le menu "URO refinement" (voir figure 3.9b). Suite à l'affinement, les positions moléculaires vont se modifier provoquant la mise à jour de la corrélation à l'intérieur de la procédure de recalage interactif.



(a) (b) (c)

FIGURE 3.9 – Menus dédiés au recalage interactif et à l'affinement automatique

Bien sûr chacun des deux modes de recalage peut s'utiliser séparément. Dans les deux cas, il est nécessaire de spécifier l'intervalle de résolution définissant le nombre de coefficients de Fourier à prendre en compte dans les calculs. Concernant l'affinement automatique, il faut en plus définir le nombre de cycles et le nombre d'itérations propre à la procédure d'optimisation.

La procédure d'optimisation consiste, pour chaque molécule indépendante, en une recherche six-dimensionnelle [10] utilisant la valeur de Q comme contrainte d'optimisation (voir section 2.1.4). Plus précisément, les positions des molécules indépendantes sont affinées séquentiellement à l'intérieur d'un cy-

10. 3 angles d'Euler définissant la rotation et 3 coordonnées cartésiennes définissant les translations par rapport à la position du modèle

cle de minimisation.

Les positions de chacune des molécules sont affinées tandis que les autres molécules restent fixes. Les cycles se terminent lorsque les changements, en terme de RMSD [11], observés durant tout un cycle sont inférieurs à seuil donné, ou lorsque le nombre d'itérations définies pour chaque cycle est terminé.

3.2.6 Sauvegarde des molécules recalées

Une fois les positions moléculaires affinées il est possible d'enregistrer, au format PDB, les coordonnées atomiques des molécules indépendantes constituant l'assemblage résultant de l'affinement. Il est également possible de sauver l'ensemble des molécules symétriques associées à chaque molécule indépendante. Pour ce faire, il suffit de sélectionner une molécule dans la liste générale accessible dans la barre d'outils à droite de la fenêtre de rendu, de se déplacer dans le menu "File", puis dans le sous-menu "Save" et de sélectionner "Molecule" ou "Constellation".

Lors de l'enregistrement des molécules indépendantes, la procédure consiste dans l'application des variables de position aux coordonnées de référence du modèle dont est issue la molécule. Dans le cas de l'enregistrement d'une constellation, la procédure est comparable pour les coordonnées de chaque molécule symétrique. Cependant la transformation, appliquée aux coordonnées de référence du modèle, est donnée par la combinaison des opérations de symétrie avec les variables de position des molécules indépendantes dont elles sont issues.

3.2.7 Déformation élastique des modèles moléculaires

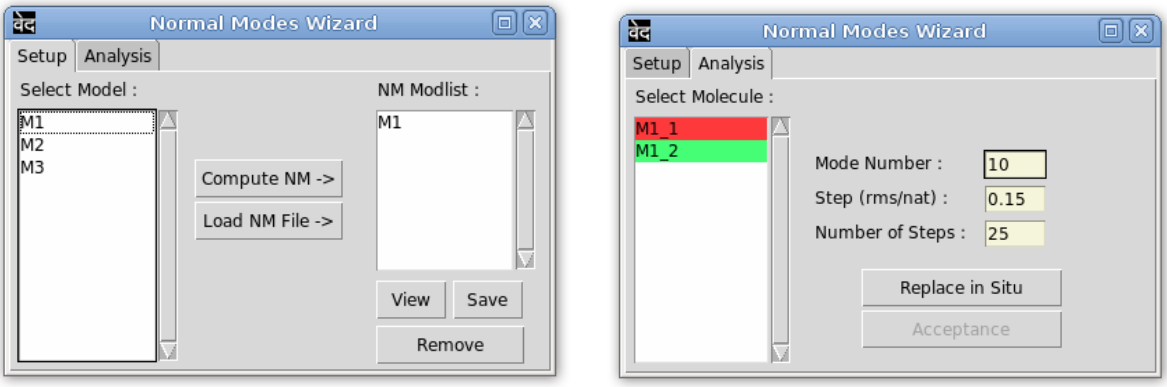
La technique permettant d'améliorer la qualité du recalage par déformations élastiques des molécules est probablement l'outil le plus avancé de $\vee\in\supset\wedge$.

Cet outil permet de simuler des mouvements de basse fréquence des molécules ressemblant beaucoup aux mouvements fonctionnels observables expérimentalement. Sur la base de cette dynamique (dite "harmonique") il est possible de construire, pour un mode normal de vibration donné, un ensemble de structures déformées, appelées conformères (voir section 2.2.3).

11. pour "Root Mean Square Deviation"

Chacune de ces différentes conformations correspond à une amplitude particulière du mouvement. Le problème est alors de déterminer quel est le conformère qui, considéré comme corps rigide, maximise, après optimisation, le coefficient de corrélation.

Un wizard dédié (le “Normal Modes Wizard”) permet de manipuler l’ensemble des fonctionnalités liées à la déformation élastique des molécules et au recalage flexible. Le premier onglet, appelé “Setup” (voir figure 3.10a), permet de calculer l’ensemble des données nécessaires à la construction de la séquence de conformères (voir section 2.2).



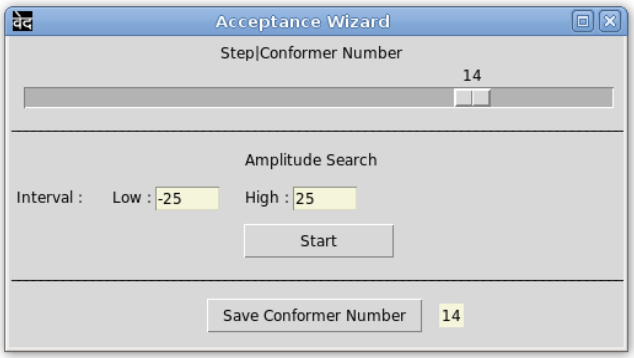
(a) Normal Modes Wizard - onglet Setup (b) Normal Modes Wizard - onglet Analyse

FIGURE 3.10 – Le “Normal Modes Wizard” permet d’étudier la déformation des molécules afin d’améliorer la qualité du recalage.

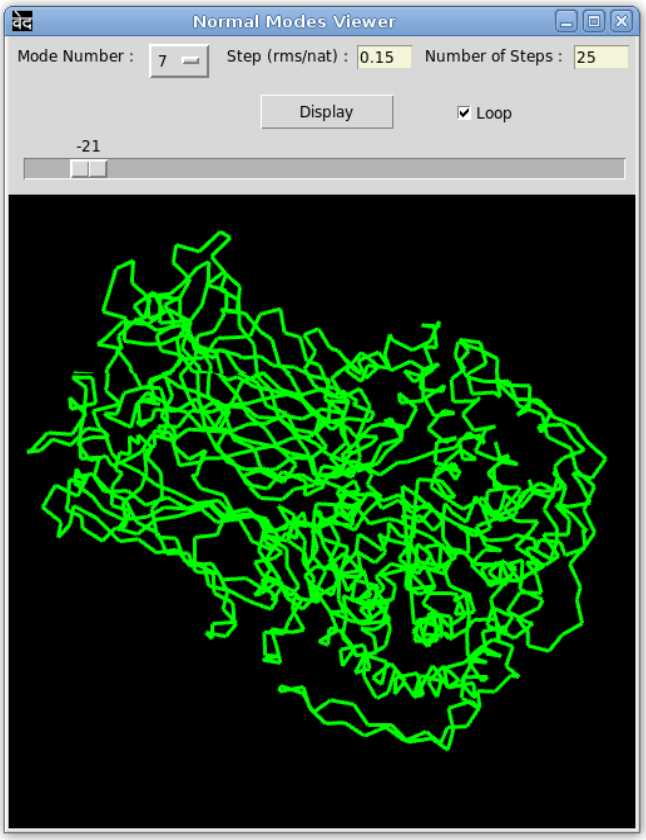
Pour ce faire, il suffit de sélectionner un modèle dans la liste de gauche du “wizard” et d’actionner le bouton “Compute NM”. Une fois le calcul terminé, il est possible de sauvegarder à l’intérieur d’un fichier, l’ensemble des vecteurs propres issus de la diagonalisation du Hessien (voir section 2.2.2). Le fichier sauvegardé pourra alors être rechargé lors d’une future session de ∨∈⊃∧, afin d’échapper à cette (longue) phase de calcul.

Une fois cette étape de calcul effectuée (ou si un fichier de vecteurs propres précédemment calculés est chargé), l’identifiant du modèle utilisé est affiché dans la liste de droite (“NM Modlist”) de l’onglet “Setup” et l’ensemble des molécules issues de ce même modèle sont affichées dans la liste de gauche de l’onglet “Analyse” (voir figure 3.10b). Cela signifie qu’il est désormais possible d’utiliser le “Normal Modes Viewer” (voir figure 3.11b) de l’onglet

"Setup" et le "Replace in Situ" de l'onglet "Analyse".



(a) Normal Modes Wizard - sous menu "Acceptance"



(b) Normal Modes Wizard - sous fenêtre "Normal Modes Viewer"

FIGURE 3.11 – "Normal Modes Wizard" : Menus supplémentaires

Dans les deux cas, la procédure débute par le calcul du déplacement de la structure le long d'un mode de vibration et se termine par la génération d'une séquence de conformères. Il est intéressant de noter ici que la normalisation en amplitude des modes normaux a été intégrée afin d'éliminer des mouvements irréalistes, trop localisés, de basse fréquence.

Que cette procédure soit lancée du "Normal Modes Viewer" externe de l'onglet "Setup" ou du "Replace in Situ" de l'onglet "Analyse", elle requiert trois paramètres :

– le numéro de mode normal de vibration allant de 7 à 26 [12].

– Le pas, donné en RMSD (Å), spécifie l'amplitude du déplacement.

– Le nombre total de pas.

Une fois que l'exécution de la procédure est terminée, la séquence de conformères résultants est chargée en mémoire.
Afin de donner l'impression que la molécule est en mouvement, tous les conformères sont alors affichés séquentiellement.

Cet affichage dynamique de la séquence de conformères peut être effectué soit de manière isolée dans le "Normal Modes Viewer", soit en situation, dans l'espace de travail de ∨∈⊃∧, dans le cas de la procédure "Replace in Situ".

Le "Normal Modes Viewer" a été conçu comme un pur outil de visualisation. Il permet d'observer la déformation de la molécule sous tous les angles de vues possibles mais également de stopper l'affichage dynamique afin de sélectionner manuellement une conformation particulière.

Le calcul du déplacement des atomes d'une molécule le long d'un mode de vibration et le chargement des conformères issus de ce calcul est assez rapide. L'utilisateur peut donc visualiser un grand nombre de modes et d'amplitudes différents. Une fois qu'une déformation lui semble intéressante, l'utilisateur peut se déplacer dans l'onglet "Analyse" afin de confirmer ou d'infirmer son impression par le calcul.

La procédure "Replace in Situ" fonctionne de manière similaire au "Normal Modes Viewer". Cependant, la déformation de la molécule est observable à l'intérieur de la carte de densité, dans le contexte du recalage. Le menu "Acceptance Wizard" (voir figure 3.11a) accessible depuis l'onglet "Analyse", une fois la procédure "Replace in Situ" lancée, permet d'aller plus loin dans l'analyse des différents conformères.

Ce dernier "wizard" est conçu pour déterminer, en terme de corrélation, lequel des conformères est le plus adapté à la carte de densité et aux molécules

12. Chaque numéro fait référence à une fréquence de vibration, les 6 premiers modes correspondent à des mouvements externes de la molécule (rotation et translation), le mode numéro 7 est donc celui de plus basse fréquence. Les modes de numéro supérieurs à 26, bien que calculés, ne sont pas utilisés car considérés comme peu ressemblants à des mouvements fonctionnels

qui l'entourent dans le cadre du recalage. La procédure de recherche est assez simple :

- chaque conformère est successivement substitué à la molécule indépendante sélectionnée ainsi qu'à l'ensemble de ses symétriques.

- La position du conformère est alors optimisée, donnant ainsi une corrélation de référence pour chacune des conformations composant la déformation moléculaire.

De manière analogue au "Normal Modes Viewer", une barre de défilement permet de sélectionner dans le "Replace in Situ" un conformère particulier, et une option supplémentaire permet d'enregistrer au format PDB le conformère sélectionné.

Contrairement à la procédure d'enregistrement standard [13], ici, seules les coordonnées de C^{α} sont sauvegardées afin de rendre compte de l'approximation effectuée lors de la construction du réseau moléculaire élastique nécessaire au calcul des modes normaux (voir section 2.2.1).

3.3 Outils supplémentaires

3.3.1 Redimensionnement des cartes de densité

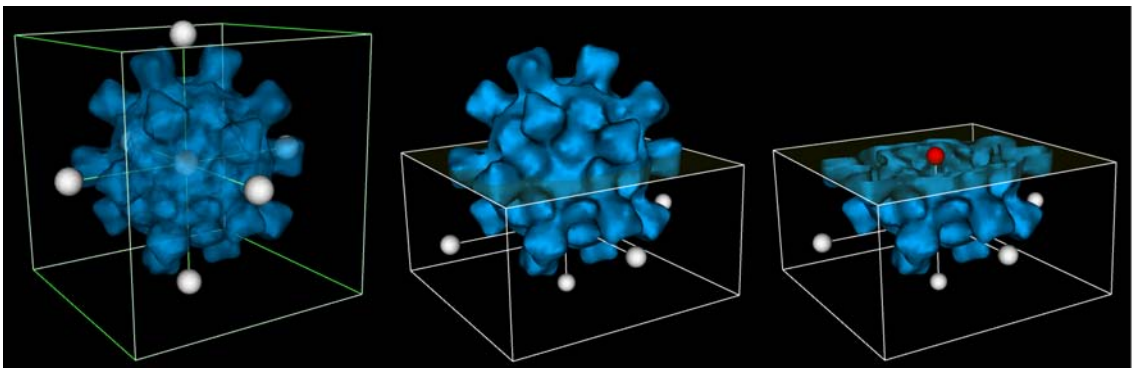


FIGURE 3.12 – Le Crop : outil visuel 3D permettant le redimensionnement des cartes de densité électroniques

Si des portions de la reconstruction ne sont pas utilisées pour le recalage ou si la carte de densité est trop grosse pour être manipulée facilement, il est

13. Procédure, qui, quelque soit la représentation des modèles, enregistre tous les atomes.

possible d'utiliser l'outil de redimensionnement des cartes intégré dans $\vee\in\supset\wedge$.

Cet outil, appelé "Crop", permet le redimensionnement manuel des cartes de densité. Il est combiné, en temps réel, avec un contrôle de compatibilité des dimensions de la carte avec les sous-routines de transformée de Fourier "Ten Eyck"[20, 21], utilisées lors de la préparation des données.

Cet outil est constitué visuellement par un parallélépipède dont les six faces, munies visuellement de poignées sphériques, peuvent être déplacées indépendamment par l'action de l'utilisateur.
Visuellement, les parties de l'isosurface se retrouvant à l'extérieur de la boite sont supprimées du rendu dans l'espace de travail de $\vee\in\supset\wedge$ (voir figure 3.12).

Le processus de "crop" est activé depuis le troisième onglet du "wizard" "Target map", celui-ci est composé, de haut en bas, par :

- trois boutons permettant de lancer, de réinitialiser et de stopper la session de redimensionnement

- six zones affichant, en temps réel, les bornes (en pixels) du volume extrait par la boîte du "crop"

- un bouton "crop map-file" qui actionne une procédure écrivant les valeurs de densité extraites de la boite dans un nouveau fichier.

Une fois le bouton "crop map-file" actionné, le fichier généré est ensuite automatiquement lu, et son contenu est chargé comme la nouvelle carte cible dans l'espace de travail de $\vee\in\supset\wedge$.

3.3.2 Recherche du grossissement optimal de la reconstruction

Le grossissement (en anglais magnification) des images de microscopie électronique peut comporter une marge d'erreur allant jusqu'à $\sim 5\%$[4], c'est pourquoi il est nécessaire de déterminer l'échelle absolue de la reconstruction.

Une fois que les molécules, suite à l'affinement, ont convergé vers des positions stables à l'intérieur de la carte de densité et, qu'ainsi une valeur de corrélation de référence a été établie, le grossissement optimal peut être

déterminé automatiquement à l'aide d'un outil appelé "Magnification Finder".

Cet outil va permettre dans un premier temps, par changements successifs des valeurs de spacing de la grille d'origine, de générer un ensemble de cartes d'échelles différentes. Une fois la transformée de Fourier de chacune de ces cartes calculée, il est possible de répéter itérativement le processus d'optimisation afin de déterminer quel facteur d'échelle donne la meilleure corrélation.

3.3.3 Statistique de convergence

La qualité des assemblages moléculaires est décrite par le coefficient de corrélation (voir section 2.1.4). Celui-ci est un excellent indice pour décrire globalement "l'accord" existant entre la reconstruction et l'ensemble des molécules qui composent l'assemblage. Pourtant, lorsqu'il s'agit d'évaluer localement la stabilité des solutions, la corrélation n'est pas informative.

Un outil, appelé "RMS Threshold", a été créé afin d'étudier le rayon de convergence de la procédure d'optimisation de $\vee\in\supset\wedge$ autour de solutions particulières.

Lors de l'optimisation, les positions finales des molécules dépendent fortement des positions initiales (voir section 2.1.5), or il n'est pas possible d'explorer de manière exhaustive l'ensemble des convergences à l'intérieur d'une zone définie. C'est pourquoi, la technique utilisée consiste à échantillonner un ensemble de positions de départ à l'intérieur d'une boule définie par une distance RMS en Å entre la solution d'origine et chacune des molécules destinées à converger.

La méthode d'échantillonnage consiste, tout d'abord, dans le tirage aléatoire d'une transformation permettant de faire varier les positions de la molécule étudiée autour de sa position initiale. L'ensemble des positions générées sont ensuite optimisées itérativement. Chacune des positions résultant de l'optimisation est ensuite comparée avec la solution initialement étudiée en terme de RMSD (en Å) et de corrélation.

Déterminer correctement la stabilité des positions moléculaires nécessite un échantillonnage convenable de l'espace des variables de position autour de la solution que l'on étudie. Cet échantillonnage dépend de plusieurs paramètres que l'utilisateur doit saisir à l'intérieur d'un wizard dédié (voir figure 3.13).

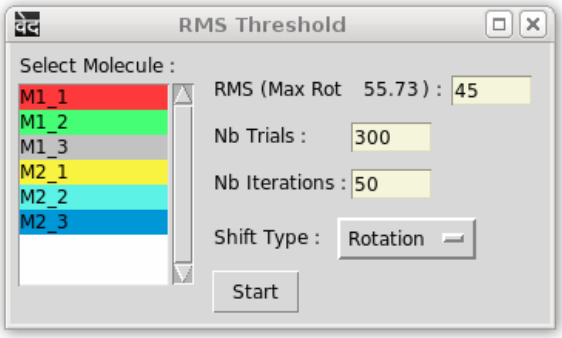


FIGURE 3.13 – RMS Threshold : outil permettant d'estimer la stabilité des solutions issues de la procédure d'optimisation

Ces paramètres sont les suivants :

- Le "RMS" est ici une distance (exprimée en Å) définissant la boule paramétrique à l'intérieur de laquelle les transformations appliquées aux variables de position de la molécule étudiée sont tirées aléatoirement.

- Le "nombre d'essais" correspond au nombre de molécules générées dans le voisinage de la solution à tester.

- Le "Mode de décalage" va indiquer au programme, le type de transformation [14] qu'il va devoir générer afin de déplacer la molécule étudiée d'une distance égale au "RMS" spécifié plus haut.

Afin de connaître plus précisément le profil de convergence de solutions particulières, il est utile d'explorer différentes valeurs pour le paramètre "RMS".

Au travers de cette exploration, l'utilisateur va prendre la mesure d'une distance limite, au delà de laquelle les molécules ne convergent statistiquement plus vers la solution étudiée. Cette connaissance, même floue, va aider l'utilisateur à juger de la confiance qu'il peut accorder à son modèle d'assemblage moléculaire.

14. Rotation uniquement, translation uniquement et rotation/translation combinées

3.3.4 Visualisation par sections

Si les frontières entre les molécules et la carte de densité sont enfouies et qu'elles ne peuvent être vues malgré l'option de transparence dans l'affichage de l'isosurface il est possible d'utiliser l'outil de "clipping slab" intégré dans $\vee\in\supset\wedge$.

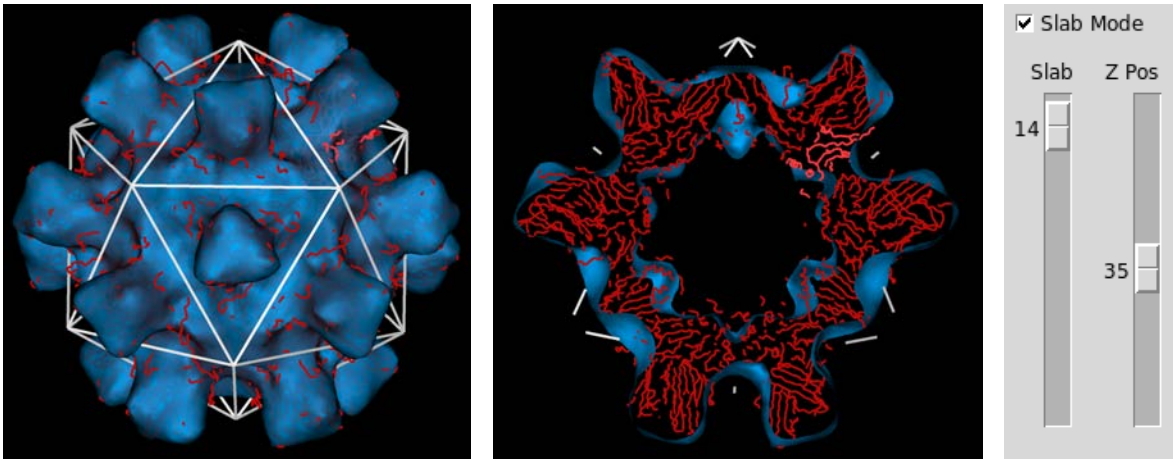
Plus simplement appelé "slab", cet outil de visualisation par section permet de limiter le rendu graphique à une zone délimitée par deux plans orthogonaux au vecteur définissant la direction de la caméra. Au chargement d'une carte de densité cible, un ensemble de contrôles apparaissent dans la barre d'outil de droite :

- Un bouton permettant d'activer et de désactiver le mode "slab".

- Une barre de défilement permettant de régler l'épaisseur de la coupe du "slab" (la distance entre les deux plans de clipping)

- Une deuxième barre permettant de faire varier la position des deux plans en profondeur (le long de la droite colinéaire au vecteur définissant la direction de la caméra)

Bien sûr, ces contrôles spécifiques au sectionnement sont compatibles avec l'ensemble des contrôles de base tels que la rotation et la translation de la camera, et le zoom. Si l'orientation des plans de "clipping" est fonction du point de vue de la caméra, le zoom, quand à lui, n'influence pas le plan de coupe, de manière à ce que l'on puisse s'approcher d'une section sans la modifier.



(a) (b) (c)

FIGURE 3.14 – Le “Slab” : outils de visualisation permettant le sectionnement de la scène 3D.

Chapitre 4

Applications

Quelques applications sont présentées dans ce chapitre afin d'illustrer le fonctionnement et les performances de $\vee\in\supset\wedge$. Les exemples, classés selon leurs groupe de symétrie, ont été choisis pour leurs propriétés (basses résolutions, grands volumes, nombre de sous-unités important, fortes symétries) afin de mettre en avant les spécificités de $\vee\in\supset\wedge$.

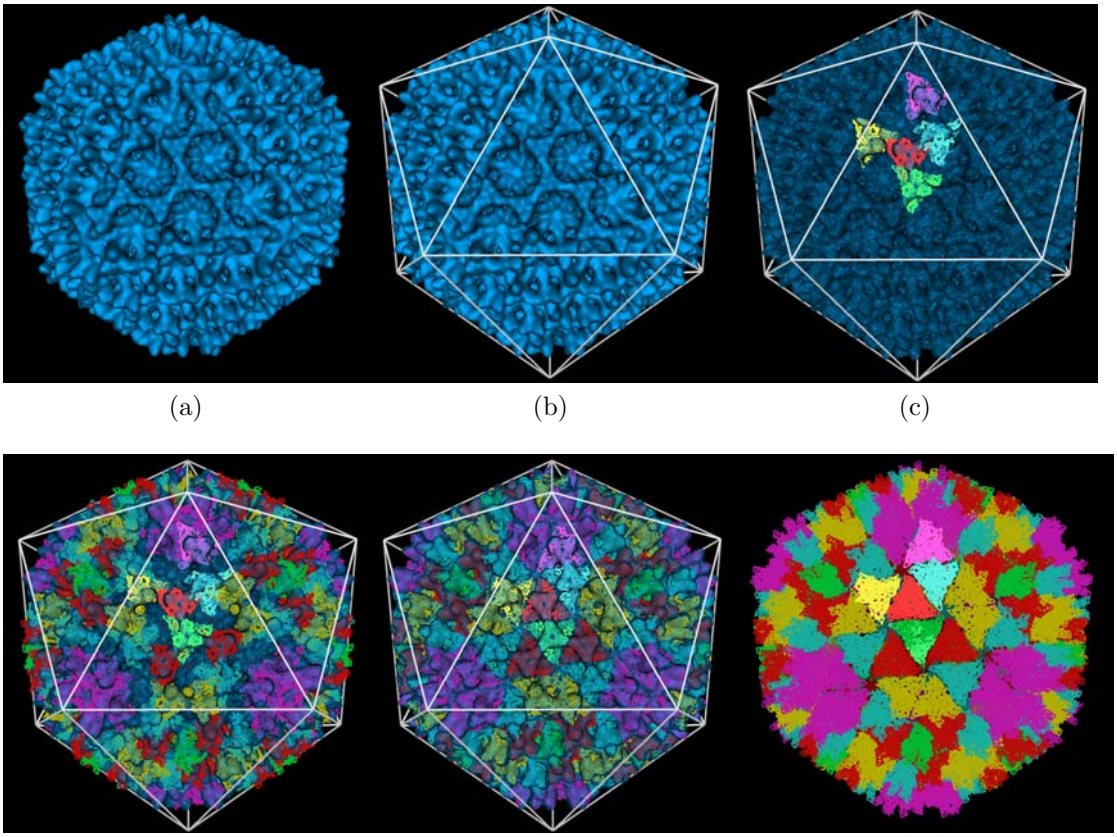
Certains des exemples présentés ici sont issus de travaux en collaboration, soit en cours de réalisation (4.3.1, 4.3.2), soit déjà publiés (4.2.2, 4.4.1). Certaines autres équipes de recherche réalisent à l'heure actuelle des travaux, utilisant $\vee\in\supset\wedge$, qu'ils ne souhaitent pas encore diffuser. C'est pourquoi, quelques exemples ont été réalisés à partir de reconstruction et de modèles atomiques issus de base de donnée afin de compléter la démonstration des possibilités de $\vee\in\supset\wedge$.

4.1 Symétries icosaédriques

4.1.1 Modèle moléculaire de la capside du virus IBDV

IBDV est un Avibirnavirus de la famille des Birnaviridae, C'est un virus, à ARN bisegmenté, très résistant. Il provoque chez les oiseaux une maladie contagieuse appelée bursite infectieuse.

Les données utilisées pour ce recalage proviennent d'une part de l'EM Data Bank[23], qui a fourni la carte de densité électronique de la capside (code EMDB : 1115), et d'autre part de la Protein Data Bank[19] qui a fourni le modèle moléculaire de la protéine VP2 sous sa forme trimèrique (code PDB : 2GSY).



(a) (b) (c)

(d) (e) (f)

FIGURE 4.1 – Différentes étapes du recalage de trimères de VP2 dans la reconstruction de la capside du virus IBDV

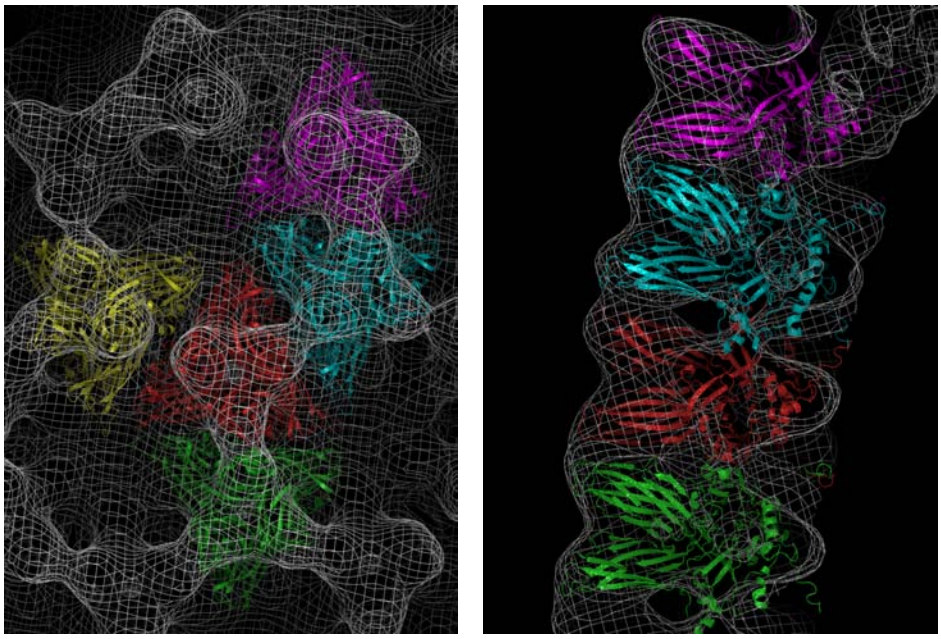
La capside du virus IBDV possède une symétrie de type icosaédriques et un nombre de triangulation [1] égale à 13. La protéine VP2 est le composant essentiel de la capside du virus. La problématique est de construire un modèle d'assemblage décrivant comment ces protéines sont architecturées.

Il faut recaler 4 trimères et un monomère pour remplir complètement la carte de densité après application des opérations de symétrie. En effet, les axes 3 de la symétrie icosaédrique sont superposés avec les axes 3 de

1. Le nombre de triangulation T est donné par la relation suivante : $T = h^2 + hk + k^2$ ou h et k sont des entiers positif définissant la position de l'axe de symétrie d'ordre 5 au sein de la maille hexagonale composant l'icosaèdre[24]

symétrie cristallographique des trimères VP2. Pourtant, dans cette application, 5 trimères indépendants ont été recalés grâce à l'utilisation du "collision filter" (voir section 3.2.3) lors de la création de la constellation.

Une fois la carte chargée (voir figure 4.1a), la symétrie configurée (voir figure 4.1b), les molécules indépendantes ajoutées à l'espace de travail et placées (voir figure 4.1c), ainsi que l'ensemble des molécules symétriques générées (voir figure 4.1d), la préparation des données effectuée, il est possible d'affiner les positions jusqu'à convergence (voir figures 4.1e et 4.1f).



(a) (b)

FIGURE 4.2 – Vues détaillées du modèle moléculaire de la capside du virus IBDV

Après sauvegarde des molécules recalées, il est facile d'utiliser n'importe quel outil dédié à la visualisation (e.g Pymol[25], Chimera[26], O[27]) afin d'étudier plus finement le modèle d'assemblage issus du recalage (voir figure 4.2).

4.1.2 Modèle moléculaire de la capside de l'Adénovirus

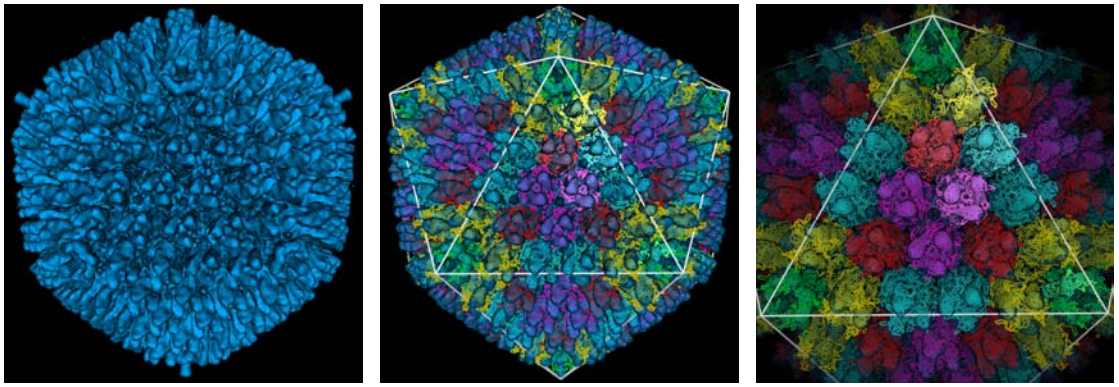
Les Adénovirus, de la famille des *Adenoviridae* sont des virus icosaédriques non enveloppés (sans couche lipidique externe) de taille relativement impor-

tante (100nm) composés d'une nucléocapside et d'un génome à ADN linéaire double brin.

La capside de ce virus, est composée de 252 capsomères (240 hexons et 12 pentons). Les capsomères situés aux sommets de l'icosaèdre sont prolongés par une fibre de longueur variable (non recalée dans cet exemple) et sont appelés pentons.

La carte utilisée (voir figure 4.3a) pour cette application, a été reconstruite par Guy Schoehn (groupe Microscopie Electronique et Méthodes, IBS/UVHCI, Grenoble). Deux modèles moléculaires ont été utilisés : une structure trimérique pour recaler les hexons [code Pdb : 1P2Z] et une structure monomèrique [code Pdb : 1X9P] pour recaler les pentons.

Un des objectifs de ce deuxième exemple est de montrer qu'il est facile de recaler plusieurs molécules indépendantes, issues de plusieurs modèles, dans une carte, haute résolution ($\sim 8\mathring{A}$), aux dimensions importantes (ici pratiquement 74 millions de pixels (419^3)).



(a) Vue générale de la reconstruction 3D de la capside de l'Adenovirus

(b) Vue générale du modèle moléculaire issue du recalage

(c) Vue détaillée du modèle le long de l'axe 3

FIGURE 4.3 – Modèle moléculaire de la capside de l'Adenovirus

Tout comme dans la construction du modèle moléculaire de la capside d'IBDV, les axes 3 de la symétrie icosaédrique sont superposés avec les axes 3 de symétrie cristallographique des hexons. Encore une fois il a été possible de recaler un modèle trimérique grâce à l'utilisation du "collision filter".

Cinq molécules indépendantes (4 trimères d'hexons et un monomère de la base du penton) ont donc été recalées afin de construire le modèle moléculaire de la capside visible sur les figures 4.3b et 4.3c. Ce modèle d'assemblage pourrait permettre, par exemple, d'étudier les contacts existants entre les différents capsomères.

4.2 Symétries hélicoïdales

4.2.1 Modèle moléculaire d'un crochet flagellaire bactérien

Le flagelle est un assemblage macromoléculaire assurant la mobilité d'une cellule. Chez les procaryotes, les flagelles sont des structures semi-rigides ancrées dans la membrane plasmique. Un flagelle de type bactérien est composé par environ 30 protéines différentes, présentes en nombre de dizaines de milliers de copies.

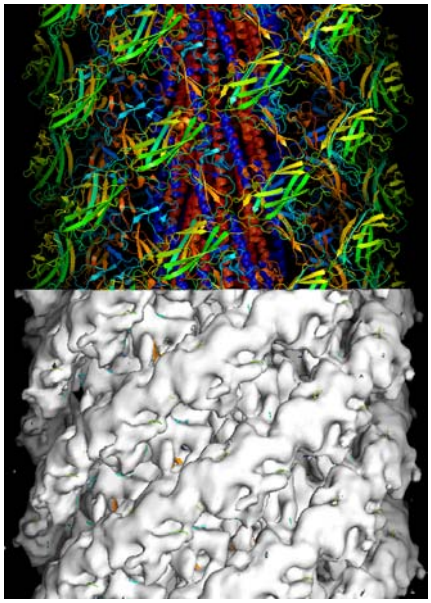
Un flagelle peut être divisé en trois parties principales :

- Le filament flagellaire est un cylindre creux et rigide qui s'étend depuis la surface cellulaire. Il est constitué de nombreux monomères d'une seule protéine appelée la flagelline. Ces monomères empruntent un canal central pour être finalement assemblés à l'extrémité distale[28].

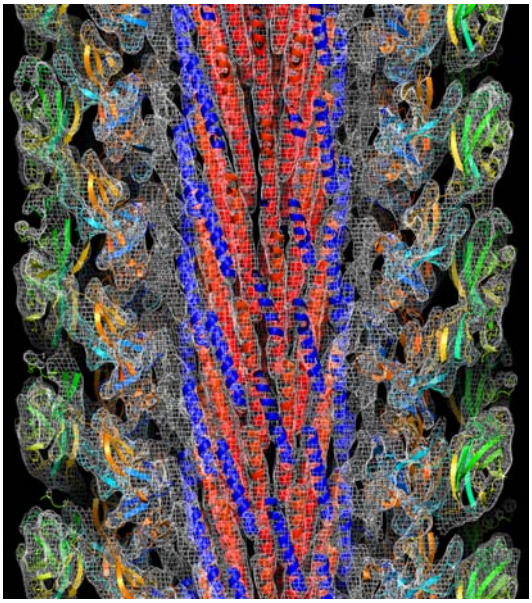
- Le crochet flagellaire, d'un diamètre supérieur au filament, est situé tout près de la surface cellulaire. Il fait la liaison entre le corps basal et le filament flagellaire. Son rôle est de transmettre le mouvement du corps basal au filament.

- Le corps basal est enfoui dans la cellule, il est constitué par un moteur moléculaire qui assure la rotation de l'ensemble du flagelle qui devient alors une hélice propulsive.

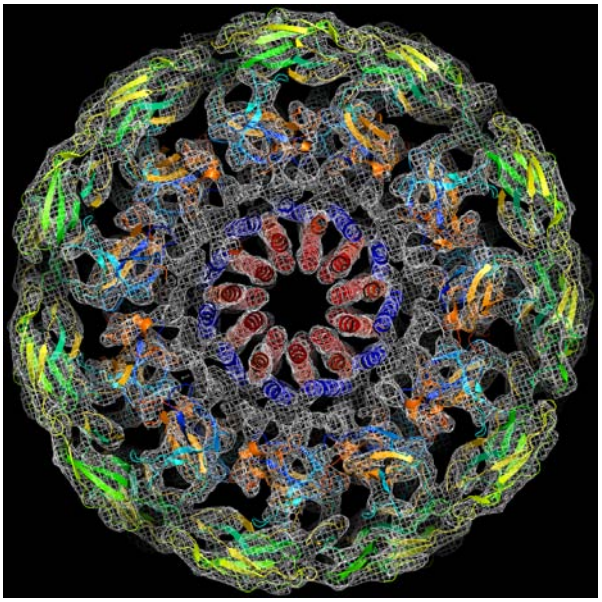
Cet exemple d'application, basé sur une carte de densité issue de Cryo-ME, a permis de construire un modèle moléculaire du crochet flagellaire. La reconstruction hélicoïdale du crochet flagellaire de la bactérie *Salmonella enterica* (Code EMDB : 1647), à haute résolution (7.1Å), a servi à recaler un modèle moléculaire contenant la structure cristallographique partielle de la protéine flgE comprenant deux hélices alpha terminales (Code PDB : 3A69).



(a) vu de coté montrant à la fois la carte de densité et le modèle macromoléculaire



(b) vu de coté d'une section longitudinale



(c) section du crochet vu le long de l'axe hélicoïdal. Le diamètre du crochet est de 180 Å et celui du canal central est de 18 Å.

FIGURE 4.4 – Modèle moléculaire d'un crochet flagellaire bactérien

Après chargement de la carte de densité et imposition de la symétrie hélicoïdale ($\Delta\Phi = 64.78^o$, $\Delta z = 4.12$Å, $\#s = 1$), la molécule indépendante et sa constellation ont été rapidement placées grâce à l'utilisation de l'"interactif fitting" (voir section 3.2.5), puis affinées automatiquement jusqu'à convergence dans une gamme de résolution comprise entre 200 et 8Å.

Une fois la convergence atteinte, la valeur de corrélation (67.9%) et le facteur R (59.0) indiquent que les sous unités de l'assemblage hélicoïdal sont bien ajustées à la densité, d'autant qu'une partie de la densité n'est pas occupée par les molécules, issues d'un modèle moléculaire incomplet. Une simple inspection visuelle (voir figure 4.4) confirme l'information donnée par la procédure d'affinement automatique.

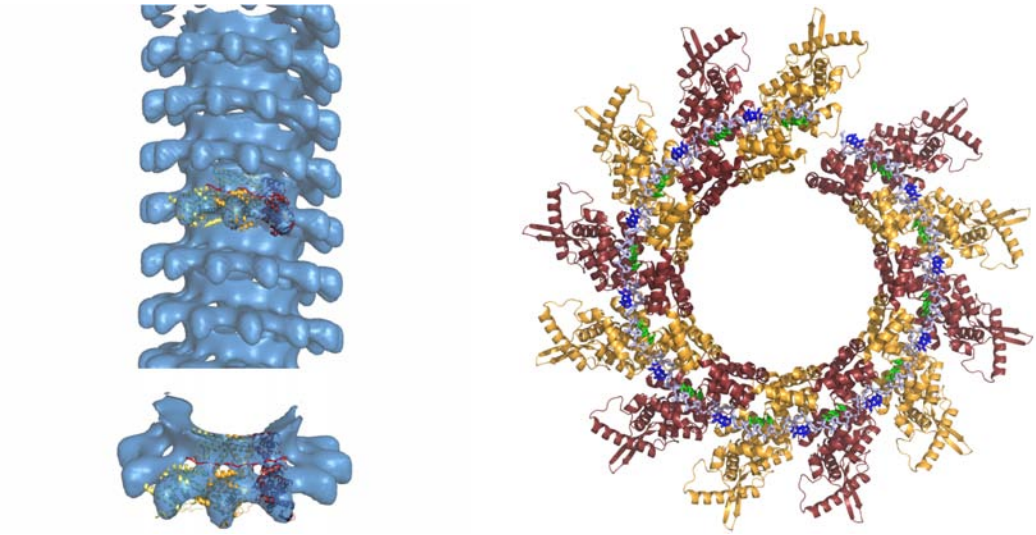
4.2.2 Orientation de la nucléoprotéine-ARN dans la nucléocapside du virus de la rougeole par microscopie électronique 3D

Les hélices de la nucléoprotéine-ARN du virus recombinant de la rougeole ont été analysées par microscopie électronique à coloration négative. Des reconstructions tridimensionnelles de nucléocapsides (digérées par la trypsine et intactes) ont été utilisées pour construire un modèle macromoléculaire (voir figure 4.5b).

Ce modèle, construit par recalage d'une structure atomique d'une sous-unité de nucléoprotéine-ARN, issue du virus respiratoire syncytial (appelé RSV pour " respiratory syncytial virus"), à l'intérieur des cartes de densité électronique (voir figure 4.5a), soutient la thèse que la place de l'ARN est à l'extérieur de l'hélice et que le domaine C-terminal désordonné se situe vers l'intérieur de l'hélice[29].

Le modèle construit suggère également la position des six nucléotides en ce qui concerne le protomère N de la rougeole.

Le travail de recalage, réalisé avec le logiciel $\vee\in\supset\wedge$ a été effectué en collaboration avec Ambroise Desfosses et Irinia Gutsche de l'UVHCI.



(a) Recalage d'une sous unité de nucléoprotéine-ARN de RSV dans une reconstruction de nucléocapside de la rougeole

(b) Détail du modèle moléculaire ayant permis d'inférer la position externe de l'ARN

FIGURE 4.5 – Modèle moléculaire de la nucléocapside du virus de la rougeole

4.3 Symétries diédriques

4.3.1 Modèle moléculaire d'un assemblage de "double roulette" constituée de transporteurs ABC

Un transporteur ABC bactériens d'antibiotiques baptisé BmrA (pour "Bacillus multidrug resistance ATP") est constitutivement exprimé chez *Bacillus subtilis*. Il a été surexprimé de façon hétérologue chez *E. coli*.

BmrA s'associe in vitro en oligomère pour former une supra-structure en double roulette contenant au total 48 monomères (voir figures 4.6A et 4.6B). Cette structure a été déterminée à basse résolution (16Å), par Cryo-ME et analyse d'images, en collaboration avec Sergio Marco et Daniel Lévy (Institut Curie, Paris).

Bien que cette structure oligomérique, extrêmement stable, soit purement artificielle, elle a permis d'étudier l'effet produit par l'addition de différents effecteurs de BmrA (nucléotides, drogues,...)[30].

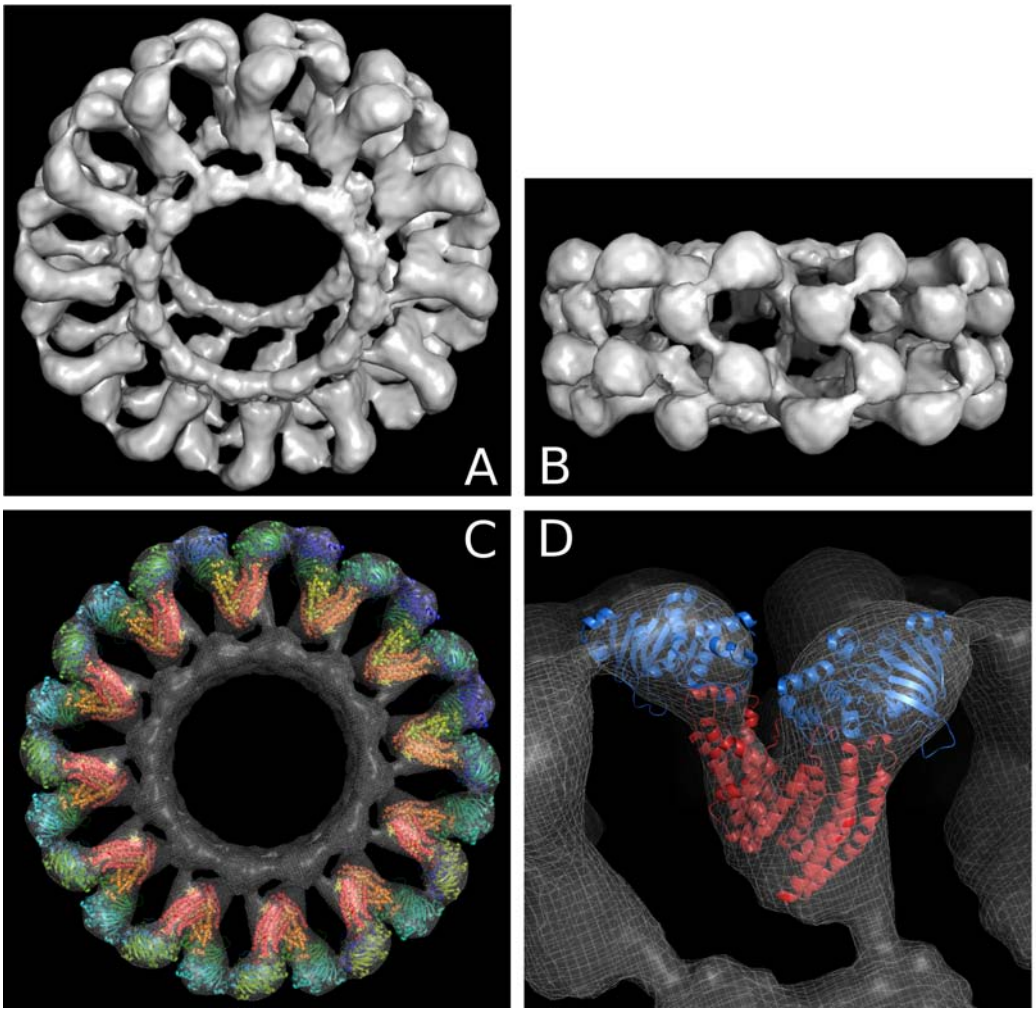


FIGURE 4.6 – Roulette de BmrA obtenue par Cryo-ME, recalage des modèles moléculaires de transporteurs ABC réalisé avec $\vee\in\supset\wedge$.
A : Roulette constituée de l'empilement de deux anneaux, chacun formé par l'association de 12 dimères de BmrA.
B : Roulette vue de dessus, chaque lobe observé à la surface de la roulette correspond au NBD.
C et D : Vue globale et en détail de la roulette montrant le recalage du NBD de Tap1 et des ICD de MDR1.

Le recalage de modèles moléculaires de transporteurs ABC (NBD [2] de Tap1 (x2) et ICD [3] du transporteur MDR1 de la souris) à l'intérieur de la

2. domaine de fixation de nucléotide (ou NBD pour "Nucleotide-Binding Domain")
3. pour "Intra-Cellular Domain"

carte de Cryo-ME (voir figures 4.6C et 4.6D) a été réalisé avec le programme $\vee\in\supset\wedge$ en collaboration avec Jean-Michel Jault (IBS, Grenoble)

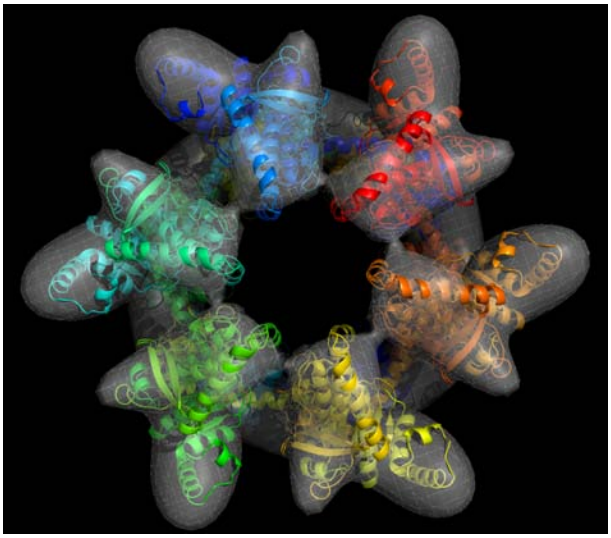
4.3.2 Modèle d'un complexe macromoléculaire impliqué dans la réplication chromosomique d'*Helicobacter pylori*

L'initiation de la réplication est l'étape consistant à la formation de la fourche de réplication en un point précis du chromosome, ainsi qu'aux chargements des enzymes nécessaires à la synthèse des copies de l'ADN. L'assemblage des protéines DnaA forme des complexes de haut poids moléculaire permettant de charger l'hélicase DnaB et de démarrer la réplication.

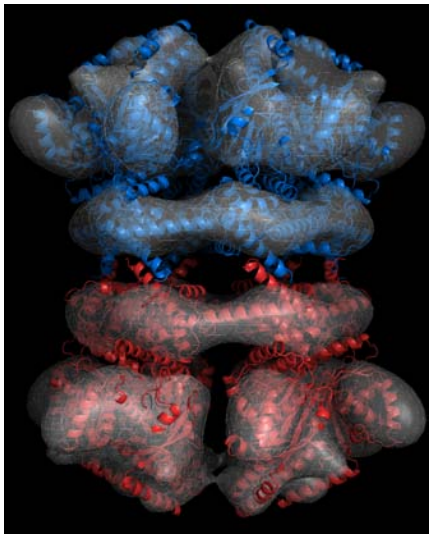
La structure de cette hélicase (HpDnaB) a été étudiée chez *Helicobacter pylori*. La structure cristalline du domaine ATPase ($HpDnaB^{C35}$) a été déterminée à 2.5Å alors qu'une structure à faible résolution de la protéine entière à 25Å a également été obtenue en collaboration avec Irina Gutsche (UVHCI, Grenoble).

Ces structures (les premières pour une hélicase réplicative de bactérie à Gram-négatif) révèlent plusieurs éléments remarquables. La structure obtenue en microscopie électronique montre qu'HpDnaB adopte une structure très différente de celle des autres bactéries décrites jusqu'ici. En effet alors que les DnaB décrites jusqu'à présent forment des hexamères, celle d'H. pylori forme des doubles hexamères tête-tête par l'intermédiaire des domaines N-terminaux. Cet arrangement ressemble remarquablement aux structures observées chez les hélicases réplicatives (appelées Mcm) des eucaryotes et des archées.

La structure du domaine $HpDnaB^{C35}$ adopte une structure similaire à celles des DnaB déjà obtenues, mais HpDnaB possède une insertion unique de 34 résidus (HPI) qui forment deux hélices qui se projettent à l'extérieur du domaine globulaire. Il a été possible de recaler la structure de $DnaB^{C35}$ dans l'enveloppe issue de la microscopie électronique avec le programme $\vee\in\supset\wedge$. Un modèle original pour la structure de HpDnaB a été proposé (voir figure 4.7). Dans ce dernier modèle, deux hélices externes forment une structure d'insertion annulaire qui pourrait permettre d'interagir avec l'ADN.



(a) Modèle d'HpDnaB vue le long de l'axe de symétrie d'ordre 6



(b) Vue de coté du modèle d'HpDnaB : un dimère d'hexamères

FIGURE 4.7 – Modèle moléculaire d'une hélicase (HpDnaB) d'*Helicobacter pylori*

L'ensemble de ces résultats permet de proposer un mode différent de chargement de DnaB sur la fourche de réplication sans nécessiter un facteur additionnel de type DnaC d'*E. coli*, absent dans le génome d'*H. pylori*. En effet la structure en double anneaux permettrait de positionner simultanément deux anneaux en directions opposées qui pourraient alors s'engager sur chacune des branches de la fourche de réplication[31].

L'ensemble de ce travail a été initié et réalisé en collaboration avec Laurent Terradot (IBCP, Lyon) et Irina Gutsche (UVHCI, Grenoble).

4.4 Symétries spéciales

4.4.1 Modèle d'une cage moléculaire modulant l'activité de l'enzyme lysine décarboxylase

La structure d'un variant d'une protéine de Régulation d'une ATPase (MoxR AAA+), présente chez *E. Coli*, révèle le principe de construction d'une cage moléculaire (voir figure 4.8) modulant l'activité de l'enzyme lysine décarboxylase.

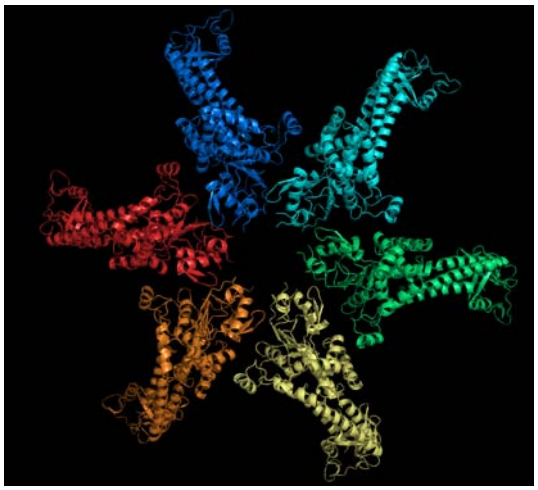
Cette enzyme, en libérant le groupement carboxyle des lysines, permet

de tamponner le milieu intracellulaire, conférant à la bactérie une résistance à différents stress[32].
Ce travail a été initié dans le département de biochimie de l'université de Toronto, en collaboration avec Irina Gutsche de l'UVHCI. Le programme $\vee\in\supset\wedge$ a été utilisé pour l'interprétation des données.

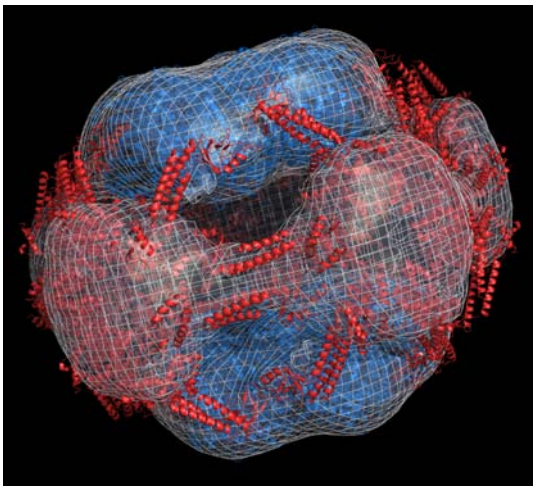
Dans un premier temps un modèle d'assemblage moléculaire hexamèrique constitué de six protéines de régulation d'ATPase de type A (MoxR AAA+) a été construit par recalage, à l'intérieur d'une carte de densité avec imposition d'une symétrie cyclique d'ordre 6 (voir figure 4.8a).

Ce même hexamère a ensuite été recalé en addition d'une structure de lysine décarboxylase dans une autre carte assez basse résolution ($\sim 25\mathring{A}$) avec imposition d'une symétrie dihédrique d'ordre 5 particulière (voir figure 4.8b).

En effet, les opérations du groupe D5 liées aux axes d'ordre 2 n'ont été appliquées qu'à la structure de lysine décarboxylase, alors que les opérations liées à l'axe 5 n'ont été appliquées qu'à l'hexamère. Cette adaptation de symétrie, s'est faite automatiquement, par application de la procédure de "collision filter" (voir section 3.2.3).



(a) Modèle moléculaire d'hexamère d'un variant d'une protéine de régulation d'ATPase (RavA MoxR AAA+)



(b) Cage moléculaire formée d'un pentamère d'hexamère autour de deux lysines décarboxylases

FIGURE 4.8 – Modèle d'une cage moléculaire formée d'un pentamère d'hexamère d'un variant d'une protéine de régulation d'ATPase

Chapitre 5

Conclusion et Perspective

Le logiciel $\vee\in\supset\wedge$ est l'aboutissement de trois ans de conception, de développement et de tests. Initialement, l'ensemble des développements devait s'appuyer sur le programme graphique UROX[11], développé précédemment au laboratoire, qui intégrait le logiciel URO[4] pour la partie calculatoire.

Le projet initial du travail de thèse consistait en l'ajout de nouveaux modules à l'intérieur d'UROX. Ce dernier se révéla par la suite inadapté pour supporter les nouvelles spécifications. La décision de développer un environnement graphique "de novo" fut alors prise.

La plupart des objectif fixés ont été atteints. Malheureusement, le temps utilisé pour la conception et la réalisation des fonctionnalités de base de $\vee\in\supset\wedge$ ne laissa pas l'opportunité d'aller au bout de tous les développements envisagés. En particulier, l'amélioration des reconstructions tomographiques par recalage de différents tomogrammes n'a pas pu être abordée. Il s'agissait d'élaborer un critère pour tenir compte du coin manquant dans les reconstructions tomographiques. Néanmoins certains pré-requis à ce développement sont aujourd'hui en place, par exemple, le recalage d'une carte de densité sur une autre carte de densité est possible.

Certaines limitations de $\vee\in\supset\wedge$ sont dues à des choix de développements :

- Le choix de la bibliothèque de visualisation 3D VTK a été bénéfique pour la rapidité de développements, mais sa caractéristique "haut niveau" a limité la liberté d'action. Par exemple, l'absence de représentations graphiques "cartoon" pour les structures secondaires dans le rendu 3D des modèles moléculaires, et le type de maille représentant l'iso-surface ne permet pas toujours de voir correctement les positions des molécules

à l'intérieur des cartes.

Néanmoins pour effectuer le recalage, l'environnement graphique de $\vee\in\supset\wedge$ est suffisant, d'autant que les résultats de $\vee\in\supset\wedge$ peuvent être facilement exportés vers des programmes dédiés à la visualisation (e.g Pymol[25], Chimera[26]).

– Le développement de $\vee\in\supset\wedge$ a été effectué sous Linux. Pourtant, la facilité d'installation du logiciel n'est pas homogène sur toutes les distributions de ce système d'exploitation. En effet, certaines bibliothèques (notamment encore VTK !) requises pour le fonctionnement du programme sont difficiles à installer lorsqu'aucun paquet pré-compilé n'est disponible, via le gestionnaire d'installation standard de la distribution. Dans ce cas, il faut alors compiler les sources à la main à travers une procédure un peu délicate.

  Un paquet d'installation auto-consistant de $\vee\in\supset\wedge$ a été créé pour Mac OSX par un conseiller technique du projet. Une fois les améliorations développées récemment incluses, ce paquet sera diffusé sur le site officiel de $\vee\in\supset\wedge$ (http ://mem.ibs.fr/VEDA).

Cette dernière année a été riche en collaboration et en interactions avec les utilisateurs. $\vee\in\supset\wedge$ a été présenté à de multiples reprises lors de différents cours et travaux pratiques. Il s'est révélé être un bon outil pédagogique pour présenter les différents aspects de la construction d'assemblages macromoléculaires.

Durant cette même période, le nombre d'utilisateurs a considérablement augmenté, entraînant de nombreux retours positifs sous bien des aspects.

Comme il est difficile de créer des tests unitaires contrôlant les fonctionnalités d'un environnement graphique, les testeurs de $\vee\in\supset\wedge$ ont pu rapporter un certain nombre de "bugs" difficiles à reproduire. Cela a permis ainsi au programme d'atteindre son degré actuel de stabilité.

En outre, les interactions avec ces utilisateurs ne se sont pas arrêtées à la simple résolution des problèmes. En effet, bon nombre d'outils et d'améliorations ont été suggérés dans le cadre d'un projet spécifique ou, plus généralement, pour pallier des besoins récurrents (e.g. "Crop", "Slab", optimisation du grossissement des reconstructions, ...).

Début 2011, le consortium américain à but non lucratif SBGrid [1], spécialisé dans l'administration à distance de serveurs applicatifs dédiés aux programmes de biologie structurale, a demandé l'autorisation d'inclure $\vee\in\supset\wedge$ dans ses services. Ce consortium, basé à l'université d'Harvard, va rendre $\vee\in\supset\wedge$ disponible, à partir d'août 2011, dans près de 150 laboratoires dans le monde entier.

L'intégration du logiciel à l'intérieur du consortium SBGrid est la marque de l'intérêt porté par la communauté scientifique à $\vee\in\supset\wedge$, et par la même de la réussite de ce travail.

1. www.sbgrid.org

Bibliographie

[1] M. Rossmann and G. Michael, "Fitting atomic models into electron-microscopy maps," *Acta Crystallographica Section D*, vol. 56, pp. 1341–1349, Oct 2000.

[2] M. M. Tirion, "Large amplitude elastic motions in proteins from a single-parameter, atomic analysis," *Phys. Rev. Lett.*, vol. 77, pp. 1905–1908, Aug 1996.

[3] K. Suhre, J. Navaza, and Y. Sanejouand, "*NORMA* : a tool for flexible fitting of high-resolution protein structures into low-resolution electron-microscopy-derived density maps,"

[4] J. Navaza, J. Lepault, F. A. Rey, C. Álvarez-Rúa, and J. Borge, "On the fitting of model electron densities into EM reconstructions : a reciprocal-space formulation," *Acta Crystallographica Section D*, vol. 58, pp. 1820–1825, Oct 2002.

[5] W. Wriggers, R. Milligan, and J. McCammon, "Situs : A Package for Docking Crystal Structures into Low-Resolution Maps from Electron Microscopy," *Journal of Structural Biology*, vol. 125, no. 2-3, pp. 185–195, 1999.

[6] F. Tama, O. Miyashita, and C. Brooks, "Normal mode based flexible fitting of high-resolution structure into low-resolution experimental data from cryo-EM," *Journal of Structural Biology*, vol. 147, no. 3, pp. 315–326, 2004.

[7] M. Orzechowski and F. Tama, "Flexible Fitting of High-Resolution X-Ray Structures into Cryoelectron Microscopy Maps Using Biased Molecular Dynamics Simulations," *Biophysical Journal*, vol. 95, no. 12, pp. 5692 – 5705, 2008.

[8] L. Trabuco, E. Villa, K. Mitra, J. Frank, and K. Schulten, "Flexible Fitting of Atomic Structures into Electron Microscopy Maps Using Molecular Dynamics," *Structure*, vol. 16, no. 5, pp. 673–683, 2008.

[9] T. Goddard, C. Huang, and T. Ferrin, "Visualizing density maps with UCSF Chimera," *Journal of Structural Biology*, vol. 157, no. 1, pp. 281–287, 2007.

[10] S. Birmanns, M. Rusu, and W. Wriggers, "Using Sculptor and Situs for simultaneous assembly of atomic components into low-resolution shapes," *Journal of Structural Biology*, vol. 173, no. 3, pp. 428–435, 2011. Combining computational modeling with sparse and low-resolution data.

[11] X. Siebert and J. Navaza, "*UROX* 2.0 : an interactive tool for fitting atomic models into electron-microscopy reconstructions," *Acta Crystallographica Section D*, vol. 65, pp. 651–658, Jul 2009.

[12] F. Tama and Y. Sanejouand, "Conformational change of proteins arising from normal mode calculations," *Protein Engineering*, no. 14, p. 1–6, 2000.

[13] H. Goldstein, C. Poole, J. Safko, and S. Addison, *Classical Mechanics, 3rd ed.* Am. J. Phys, 2002.

[14] I. Bahar and A. Rader, "Coarse-grained normal mode analysis in structural biology," *Current Opinion in Structural Biology*, vol. 15, no. 5, pp. 586–592, 2005. Carbohydrates and glycoconjugates/Biophysical methods.

[15] W. J. Schroeder, K. M. Martin, and W. E. Lorensen, "The design and implementation of an object-oriented toolkit for 3d graphics and visualization," *Visualization Conference, IEEE*, vol. 0, p. 93, 1996.

[16] J. E. Grayson, *Python and Tkinter programming.* Greenwich, CT, USA : Manning Publications Co., 2000.

[17] B. Meyer, "Conception et programmation par objets," *Seconde édition. Informatique Intelligence Artificielle. InterÉditions, Paris*, 1991.

[18] G. Kleywegt and T. Jones, "xdlMAPMAN and xdlDATAMAN - programs for reformatting, analysis and manipulation of biomacromolecular electron-density maps and reflection data sets," *Acta Crystallographica Section D*, vol. 52, pp. 826–828, 1996.

[19] H. Berman, K. Henrick, and H. Nakamura, "Announcing the worldwide Protein Data Bank," *Nat Struct Biol*, no. 10, p. 980, 2003.

[20] J. Navaza, "On the computation of structure factors by FFT techniques," *Acta Crystallographica Section A*, vol. 58, pp. 568–573, Nov 2002.

[21] L. F. Ten Eyck, "Efficient structure-factor calculation for large molecules by the fast Fourier transform," *Acta Crystallographica Section A*, vol. 33, pp. 486–492, May 1977.

[22] E. E. Castellano, G. Oliva, and J. Navaza, "Fast rigid-body refinement for molecular-replacement techniques," *Journal of Applied Crystallography*, vol. 25, pp. 281–284, Apr 1992.

[23] C. L. Lawson, M. L. Baker, C. Best, C. Bi, M. Dougherty, P. Feng, G. van Ginkel, B. Devkota, I. Lagerstedt, S. J. Ludtke, R. H. Newman, T. J. Oldfield, I. Rees, G. Sahni, R. Sala, S. Velankar, J. Warren, J. D. Westbrook, K. Henrick, G. J. Kleywegt, H. M. Berman, and W. Chiu, "EMDataBank.org : unified data resource for CryoEM," *NUCLEIC ACIDS RESEARCH*, vol. 39, pp. D456–D464, JAN 2011.

[24] T. Baker, N. Olson, and S. Fuller, "Adding the Third Dimension to Virus Life Cycles : Three-Dimensional Reconstruction of Icosahedral Viruses from Cryo-Electron Micrographs," *Microbiol. Mol. Biol. Rev.*, vol. 63, 1999.

[25] W. Delano, "The PyMOL Molecular Graphics System," 2002.

[26] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin, "UCSF Chimera : a visualization system for exploratory research and analysis," *Journal of computational chemistry*, vol. 25, pp. 1605–1612, oct 2004.

[27] T. Jones, M. Bergdoll, and M. Kjeldgaard, "O : A macromolecule modeling environment," *Crystallographic and Modeling Methods in Molecular Design*, p. 189–199, 1990.

[28] T. Fujii, T. Kato, and K. Namba, "Specific arrangement of [alpha]-helical coiled coils in the core domain of the bacterial flagellar hook for the universal joint function," *Structure*, vol. 17, no. 11, pp. 1485 – 1493, 2009.

[29] A. Desfosses, G. Goret, L. F. Estrozi, R. Ruigrok, and I. Gutsche, "Nucleoprotein-RNA Orientation in the Measles Virus Nucleocapsid by Three-Dimensional Electron Microscopy," *J. Virol.*, vol. 85, pp. 1391–1395, 2011.

[30] S. Marco, D. Lévy, G. Goret, et al, and J. Jault, "Manuscrit en préparation,"

[31] M. Stelter, I. Gutsche, U. Kapp, G. Bajic, G. Goret, M. Jamin, J. Timmins, et al, and L. Terradot, "Architecture of a dodecameric bacterial replicative helicase," *Soumis à Proc Natl Acad Sci USA*.

[32] M. E. Bakkouri, I. Gutsche, U. Kanjee, B. Zhao, M. Yu, G. Goret, G. Schoehn, W. Burmeister, and W. Houry, "Structure of RavA MoxR AAA+ protein reveals the design principles of a molecular cage modulating the inducible lysine decarboxylase activity," *Proc Natl Acad Sci U S A*, no. 107(52), pp. 22499–504, 2010.

Chapitre 6

Annexes

Complément d'information annexe concernant l'architecture de $\vee\in\supset\wedge$

Table des matières

1 Généralités

$\vee\in\supset\wedge$ est un programme de taille conséquente. Toute les sources cumulées représentent plus de 25000 lignes de code. Tout au long du développement de $\vee\in\supset\wedge$ l'architecture a été un guide permettant de maîtriser la complexité croissante du logiciel.

Une première solution mise en œuvre pour éviter que le code ne grossisse de manière monolithique est de décomposer le logiciel en des sous-ensembles plus faciles à maîtriser. Cette découpe repose sur un paradigme architectural dit orienté objet. Ce dernier consiste dans l'élaboration de briques logicielles appelées objets, qui intègrent les données et les opérations de traitement de ces données (méthodes). Dans le paradigme orientée objet, une catégorie d'objet particulière appelée "classe" déclare des propriétés communes à un ensemble d'objets. Une classe déclare des attributs représentant l'état des objets ainsi que les méthodes représentant leur comportement.

Les 7 modules python de $\vee\in\supset\wedge$ sont formés d'une vingtaine de classes cumulant entre elles environ 300 méthodes différentes. Afin de préciser le fonctionnement général du programme, les classes les plus importantes seront détaillées sous la forme de schémas UML partiels. Ces représentations seront ensuite intégrées dans un diagramme de classe plus général, au cours d'une discussions concernant les patrons de conception utilisés dans le développement de $\vee\in\supset\wedge$.

2 Schémas partiels de classes

Dans le paradigme orienté objet, un programme consiste dans l'interaction et l'évolution des objets qui le composent. La description dynamique de cet ensemble de composants est très complexe, c'est pourquoi, une description statique des classes est réalisée dans un premier temps sans dévoiler les patrons de conception qui seront décrits dans la section suivante. Les schémas partiels des classes les plus importantes sont situés à l'intérieur d'un cadre général qui impose une séparation entre l'interface homme-machine (la présentation), les traitements (services), les entités (données), et enfin le contrôleur qui orchestre le tout.

2.1 IHM

Dans le développement de $\vee\in\supset\wedge$ l'interface homme-machine (IHM) est entièrement décrite par une classe occupant l'intégralité d'un module dédié (module Itf) assez consistant (plus de 2000 lignes de code). Ce module et cette classe portent le même nom. La classe Itf utilise une collection de composants d'interface graphique ("widgets") issus de la bibliothèque Tk pour Python. La classe Itf contient une méthode associée à chaque fenêtre ("wizard") de plus haut niveau ("toplevel"), ainsi qu'un ensemble d'attributs (variables et pointeurs) permettant la communication d'informations partagées par ces "wizards".

À la différence d'un programme en ligne de commande, où l'interaction avec l'utilisateur est séquentielle, dans une autre logique, une interface graphique fait

FIGURE 1 – Classe supportant l'interface homme-machine

intervenir la notion de programmation événementielle. À tout moment, chaque "widget" est susceptible d'être affecté par l'action de l'utilisateur (l'événement). Il existe des événements simples (clic de souris sur un bouton, saisie au clavier dans un champ) et des événements plus complexes (navigation dans un menu ou une liste déroulante).

À chaque "widget" est attaché, par défaut, un certain nombre de réponses automatiques à des événements. Celles-ci correspondent à une gestion des événements de bas niveau nécessitant très peu d'intervention du programmeur. Une boucle événementielle les prend en charge et les répartit. Dans tout les autres cas, il est possible de lier un widget ou un événement à un appel de procédure (binding) ou à une commande extérieure (callback).

La plupart des "wizards" de l'IHM contiennent une multitude de composants dont la gestion est effectuée directement dans la classe Itf (ouverture d'un panel de couleur, ajout d'un item dans une liste ...). D'autres en revanche impliquent l'utilisation d'entités et des classes de traitements.

2.2 Entités

\/∈⊃/\ manipule trois principaux types de données (voir figure 2) : une carte de densités électronique représentée par une instance de la classe Map, ainsi que des modèles moléculaires et des molécules indépendantes représentées respectivement par les instances des classes Mod et Mol. Il est intéressant de noter que ces trois classes ne sont pas à proprement parler des données purement métier. En effet, ces classes hybrides intègrent un ensemble de méthodes et d'attributs qui leurs permettent notamment d'assurer elles même leurs représentations polygonales.

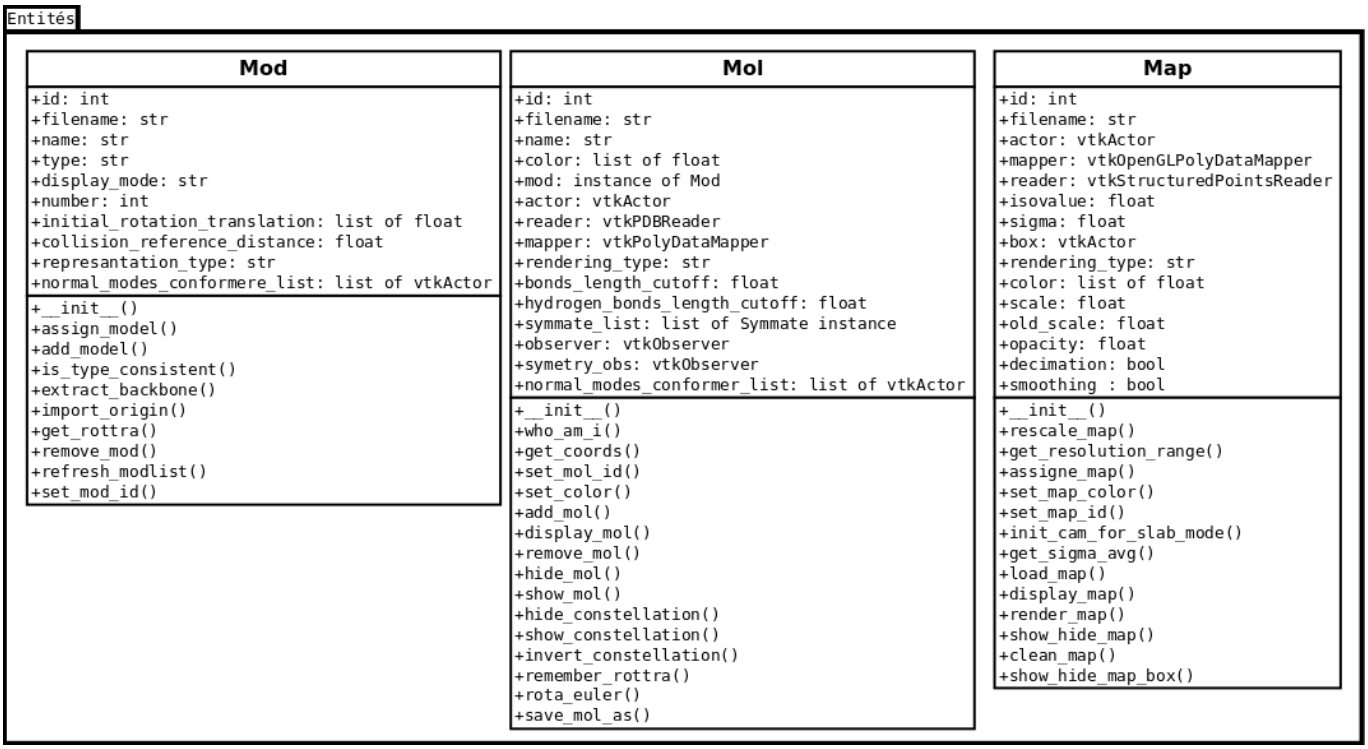


FIGURE 2 – Différentes classes représentant les entités

2.3 Traitements

Les traitements à l'intérieur de $\vee\in\supset\wedge$ sont localisés dans une dizaine de classes spécialisées qui ont parfois recours, au sein de leurs méthodes, à des appels externes à des exécutables binaires Fortran. Les quatre exemples de classes représentés sur la figure 3 sont assez représentatifs du fonctionnement général du noyaux fonctionnel. L'ensemble des méthodes implémentées dans ces classes sont déclenchées (directement ou par conséquence) depuis l'IHM en réponse à des actions de l'utilisateur. Au niveau du code, cela signifie que la classe Itf (voir section 2.1) doit pouvoir accéder assez largement aux méthodes incluses dans les objets instanciés par les classes de traitements. En retour ces derniers doivent pouvoir accéder à l'intégralité des données afin d'effectuer les traitements requis. Cependant afin de limiter le degré de couplage inter-modulaire et la dépendance entre les classes, un protocole d'échange est établi (voir section 3.1).

2.4 Contrôleur

Le contrôleur est un objet qui se charge globalement de la gestion des événements (rediriger les requêtes provenant de l'IHM vers les modules de traitements) et de la synchronisation entre les données et l'affichage.

En exception au cadre général spécifiant la séparation de la couche de présentation

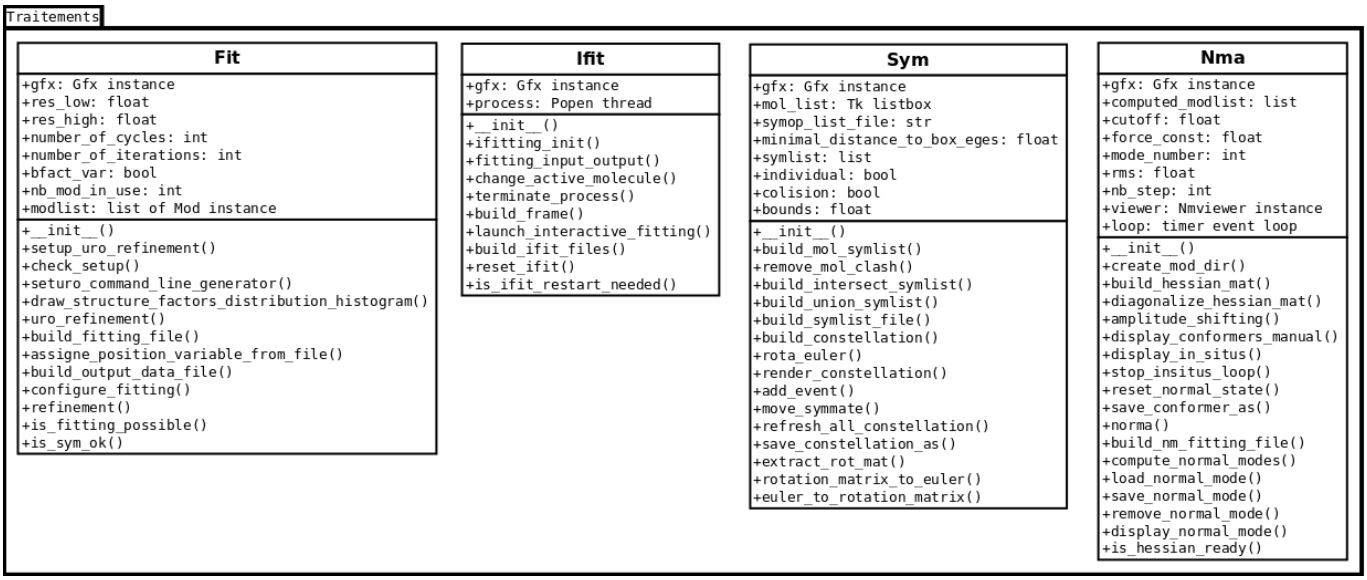


FIGURE 3 – Traitements

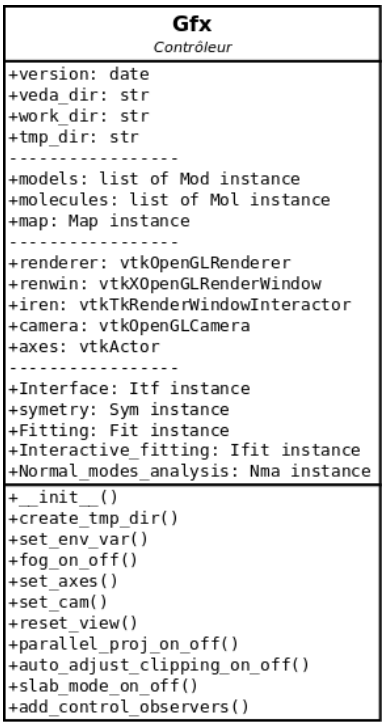


FIGURE 4 – Classe Contrôleur

et de traitements évoqué plus haut, le contrôleur cumule également la fonction de classes de gestion du graphisme (d'où son nom Gfx). Cependant, même si la classe Gfx effectue certains traitements (mapping, rendu, ...), elle ne modi-

fie aucunement les entités. Elle reçoit les requêtes de l'IHM et se contente de transmettre aux classes de traitement.

3 Patrons de conception

Tout comme l'algorithmique permet de structurer le corps des méthodes (à l'intérieur des classes), une technique de génie logiciel permet de structurer l'organisation des classes entre elle. Cette technique d'organisation en patron de conception (en anglais "design pattern") permet de minimiser les interactions qu'il peut y avoir entre les différentes classes (ou plus généralement modules) d'un même programme. Elle permet d'appliquer des solutions déjà existantes à des problèmes courants de conception (couplage, code circulaire, complexité ...) permettant ainsi d'éviter au développement d'aboutir à des anti-patron (plats de spaghettis, objet omniscient ...).

L'architecture de la couche haut niveau de $\vee\in\supset\wedge$ inclue différents patrons de conception "académiques" décrit ci-dessous.

3.1 MVC2 (Modèle, Vue, Contrôleur)

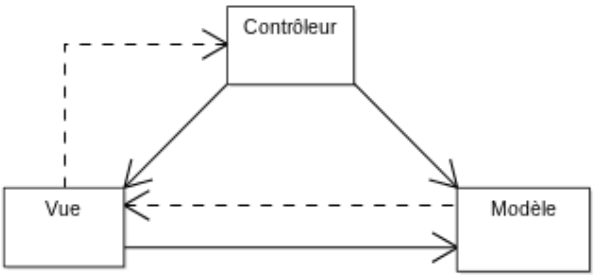


FIGURE 5 – Schéma conceptuel du patron Modèle/Vue/Contrôleur

Le MVC (Modèle, Vue, Contrôleur) est un patron de conception décrivant un type d'organisation pour les IHM. Ce patron stipule qu'un logiciel interactif doit être divisé selon trois principaux composants (voir figure 5) :

- **Le modèle** représente le comportement de l'application : traitements des données, interactions avec les entités, ... Il décrit et contient les données manipulées par l'application et assure leur bonne gestion.

- **La vue** correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, ...). Ces différents événements sont envoyés au contrôleur. La vue n'effectue aucun traitement, elle affiche les résultats des traitements effectués par le modèle et interagit avec l'utilisateur.

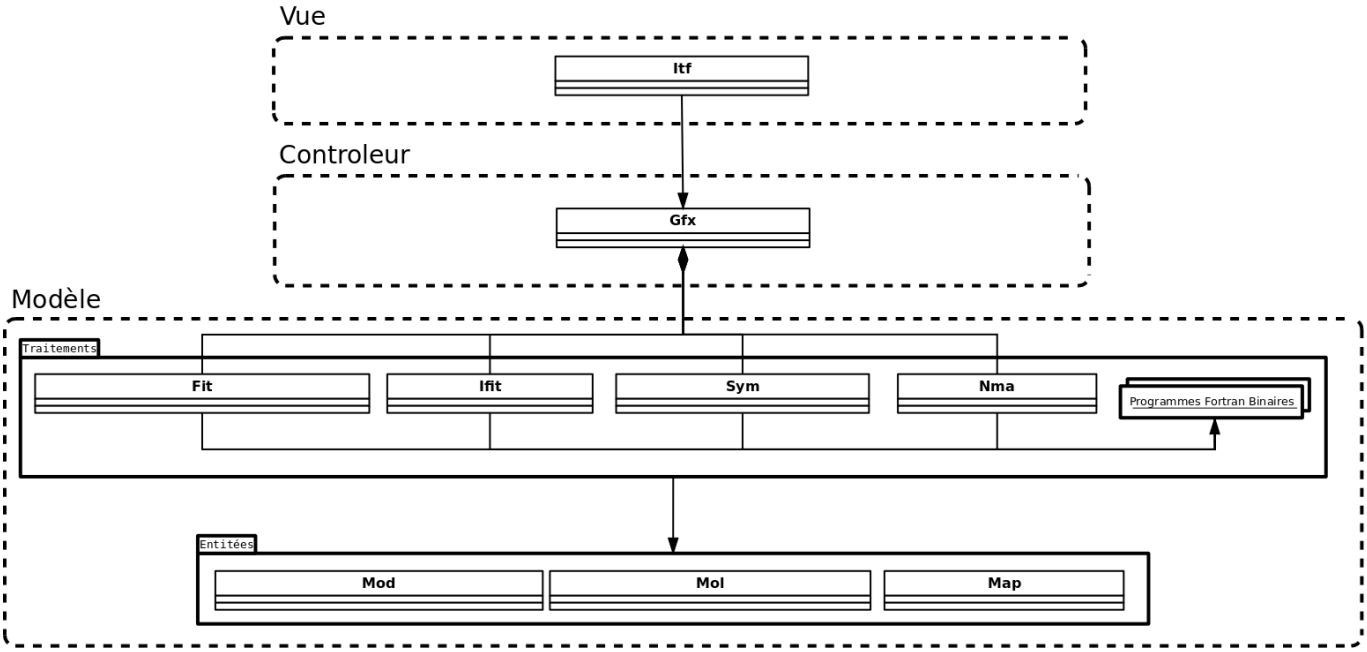


FIGURE 6 – Diagramme de classe du patron de conception Modèle/Vue/Contrôleur dans le contexte de l'implémentation

– **Le contrôleur** prend en charge la gestion des événements, il s'assure de la cohérence des représentations avec les données internes et convertit les actions de l'utilisateur en opérations fonctionnelles. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer.

Le MVC, peut cependant se révéler lourd à mettre en place à cause de la multitude de contrôleurs à implémenter. Afin de simplifier l'implémentation d'un modèle de ce type, une nouvelle version a été introduite. Cette version appelé MVC2 est très similaire au MVC à la différence qu'il n'y a plus qu'un seul contrôleur qui se charge de rediriger les requêtes vers le bon traitement.

Le MVC est un modèle abstrait qui ne décrit pas sous quelle forme sont réalisées et connectées les différentes interfaces, par contre, il offre un cadre structurant à l'application en imposant une séparation entre les données, la présentation et les traitements. Durant le développement de \/∈⊃/\ la spécification de l'ensemble des interactions existant entre ces trois agents à était implémentée en s'inspirant d'un patron de conception appelé médiateur.

3.2 Médiateur

Le patron de conception médiateur est ce qui à permis à \/∈⊃/\ d'atteindre une forte cohésion. Il a contraint les méthodes à manipuler le même ensemble de données. Dans le programme, sauf exceptions, les méthodes sont appelées dans un ordre spécifique et sont dédiées à une seule et unique tâche.

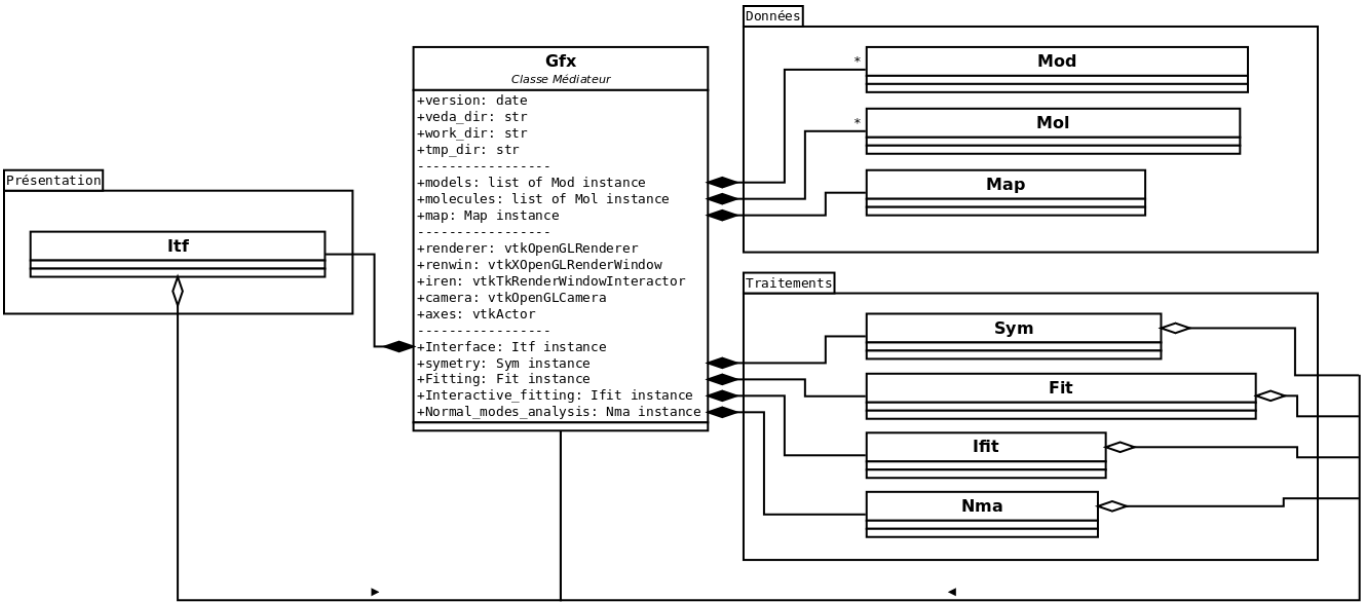


FIGURE 7 – Diagramme de classe du patron de conception Médiateur dans le contexte de l'implémentation

Pratiquement pour réaliser l'implémentation de ce patron, une classe centrale (le Contrôleur Gfx) est utilisée comme médiateur. Elle pointe sur l'ensemble des objets (de présentation, de données et de traitements) et est capable de gérer l'ensemble de leurs interfaces. A l'inverse l'ensemble des classes de traitements possède une référence de l'objet gfx leur permettant d'accéder aux attributs et au méthodes des entités (voir figure 7).

3.3 Observateurs

Une autre méthode permet de coupler les classes de façon à réduire leurs dépendances entre elles. Elle consiste dans la notion d'observation événementielle. Lorsque il est nécessaire de gérer des événements ou quand une classe déclenche l'exécution d'une ou plusieurs autres méthodes, il est utile d'utiliser le patron de conception observateur. Ce dernier permet de détecter un signal provenant d'un objet (qui joue le rôle d'observé), cela entraîne alors une action adéquate de l'observateur en fonction des informations qui lui parviennent depuis les objets qu'il observe.

Par exemple, dans $\vee\in\supset\wedge$, les objets molécules indépendante (instanciées par la classe Mol) encapsulent à leurs créations un objet observateur (symetry_obs :vtkObserver). Une fois activé lors de la création d'une constellation, il permet de mettre à jour automatiquement la position des molécules symétriques à chaque changement des variables de position. Ainsi chaque fois qu'un mouvement de molécule indépendante est détecté, une méthode d'un des objets instanciés par la classe Mol (méthode Mol.move_symmates()) va assigner de nouvelles variables de position à chaque molécule symétrique. Pour ce faire, cette méthode tient compte

de la matrice de position et d'orientation observée, et la combine à la transformation associée à chaque molécule symétrique.

Un deuxième observateur de molécules indépendantes (Mol.ifit_obs) est activé lors du lancement de la procédure de recalage interactif (Ifit.launch_interactive_fitting()). Tout comme le symetry_obs, le ifit_obs va détecter si un déplacement d'une des molécules est effectué par l'utilisateur. Si c'est le cas, il va lancer la méthode Ifit.fitting_input_output() qui va, à sont tour, encoder les nouvelles positions moléculaires et va les envoyer sous la forme d'une frame au sous-processus de calcul de la classe Ifit (Ifit.process). Ce processus va alors renvoyer de manière synchrone une corrélation associée à ce changement, elle même assignée aux différents composants de l'interface requérant cette information.

4 Interfaçage Python/Fortran

4.1 Etat des lieux

Le package de binaire URO, écrit en Fortran, préexistait largement au développement de \∈∧\. Ce package, toujours maintenu par Jorge Navaza, est toujours disponible sous sa forme originale sur le site web du groupe Microscopie Electronique et Méthodes. L'utilisation d'URO dans sa forme première est décrite précisément dans un "writeup" sous la forme d'un protocole décrivant chaque étape d'un recalage :

- La configuration préliminaire : la définition des variables d'environnement et des alias nécessaires au fonctionnement des différents programmes composant le package.

- La préparation des fichiers de données d'entrées : mise au format de la carte de densité, génération des opérations du groupe de symétrie, création des liens symboliques pointant vers les fichiers PDB ...

- L'édition des différents fichiers plats de configurations : limites de résolution du recalage, sélection des opérations de symétrie, listes des modèles et des molécules...

- La séquence d'exécution des binaires : calcul des facteurs de structure de la carte, calcul des facteurs de diffusion moléculaire, conversion des coordonnées, recalage ...

Dans l'ancienne version d'URO, l'ensemble de ces opérations étaient effectuées à l'aide du terminal sans aucune possibilité de visualisation. Seul le résultat du recalage était visible à l'aide du logiciel de visualisation graphique O.

4.2 Intégration

Ainsi la partie Python de ∨∈⊃∧ est venue s'intégrer comme un encapsuleur (en anglais "wrapper"), gérant les entrées et les sorties de tout les binaires issus du Fortran. De son coté la partie constituée originellement du pakage URO a également bien évoluée.

Pour simplifier le passage de paramètres Python/Fortran, la plupart des binaires Fortran ont reçu une interface écrite en langage cshell afin d'éviter l'utilisation de la syntaxe "<<ENDOF p_1 p_2 [...] p_n ENDOF" dans les appels système [1] à l'intérieur du code Python.

L'intégration des graphismes à souvent nécessité l'emploi de références communes. Par exemple, afin d'être certains que les positions et l'orientation des molécules comme corps rigides soient identiques dans les deux couches logicielles, un outils de mise en position de référence a été développé spécifiquement pour ∨∈⊃∧.

Cet outils place les molécules comme corps rigides dans une position particulière (centre de masse à l'origine, axes d'inertie principaux alignés avec les (X, Y, Z) du repère orthonormé) et renvoie la transformation effectuée à la partie graphique de ∨∈⊃∧, qui va alors assigner cette même référence sans que cela apparaisse visuellement.

Un autre exemple de nouveau développement à l'intérieur de la couche Fortran concerne le recalage interactif. L'idée était de brancher le plus directement possible un exécutable Fortran chargé de calculer la corrélation pour un ensemble de carte, symétrie, modèles, molécules et coordonnées moléculaires donnés. La partie graphique envoie en temps réel les modifications de position et d'orientation des différentes molécules.

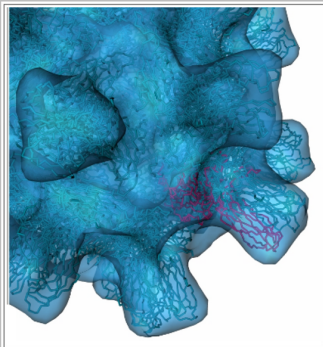
Après différents essais infructueux un système de type thread gérant le binaire de calcul comme un processus [2] a été choisi. L'exécutable Fortran écoute sur son entrée standard et reçois un ensemble de données envoyé depuis Python (molécule active, rotation et translation associées). En retour une autre méthode Python lit sur la sortie standard de l'exécutable Fortran la corrélation et le facteur R associés à chaque envoi de manière synchrone.

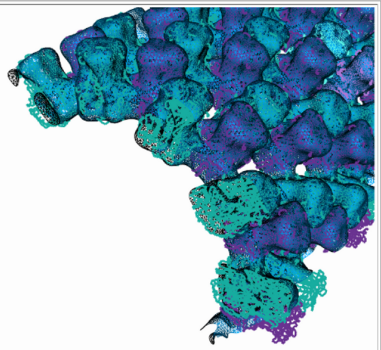
Ainsi, l'interfaçage du Fortran s'est faite de différentes manières : passage de paramètres à l'exécution des binaires, fichiers plats (configuration, données, résultats), communication directe via les Entrée/Sortie.

Bien que l'essentiel (en terme de calcul) d'URO était déjà présent avant l'existence de ∨∈⊃∧, le code Fortran a également bien évolué tout au long du développement de l'environnement graphique. Au final ∨∈⊃∧ est le fruit d'une intense collaboration entre Jorge Navaza (contributeur majoritaire du code Fortran) et moi même (contributeur exclusif du code Python). Cette collaboration a abouti à un environnement robuste et efficace combinant la rapidité du code Fortran issus d'URO (et de Norma pour les modes normaux), et la facilité d'utilisation et l'interactivité apportées par l'aspect graphique développé en Python.

1. Les appels système sont passés grâce à la librairie (built-in) OS
2. la classe Popen du module Subprocess utilisé pour géré le processus

6.2 Site Web





Last updated : July 11, 2011
Comments/Questions : contact us
developer : Gael Goret

The problem

How to interpret low-resolution reconstructions of macromolecular assemblies (typically Transmission EM reconstructions) at atomic scale ?

The problem can be solved in the particular but frequent case where subunits are known at atomic scale (typically MX or NMR molecular models).

Our solution

The main idea is to fit the available atomic models (considered as rigid bodies) into the EM reconstruction.

The method is to compare, the EM map with a calculated model-based electron density within a selected volume.

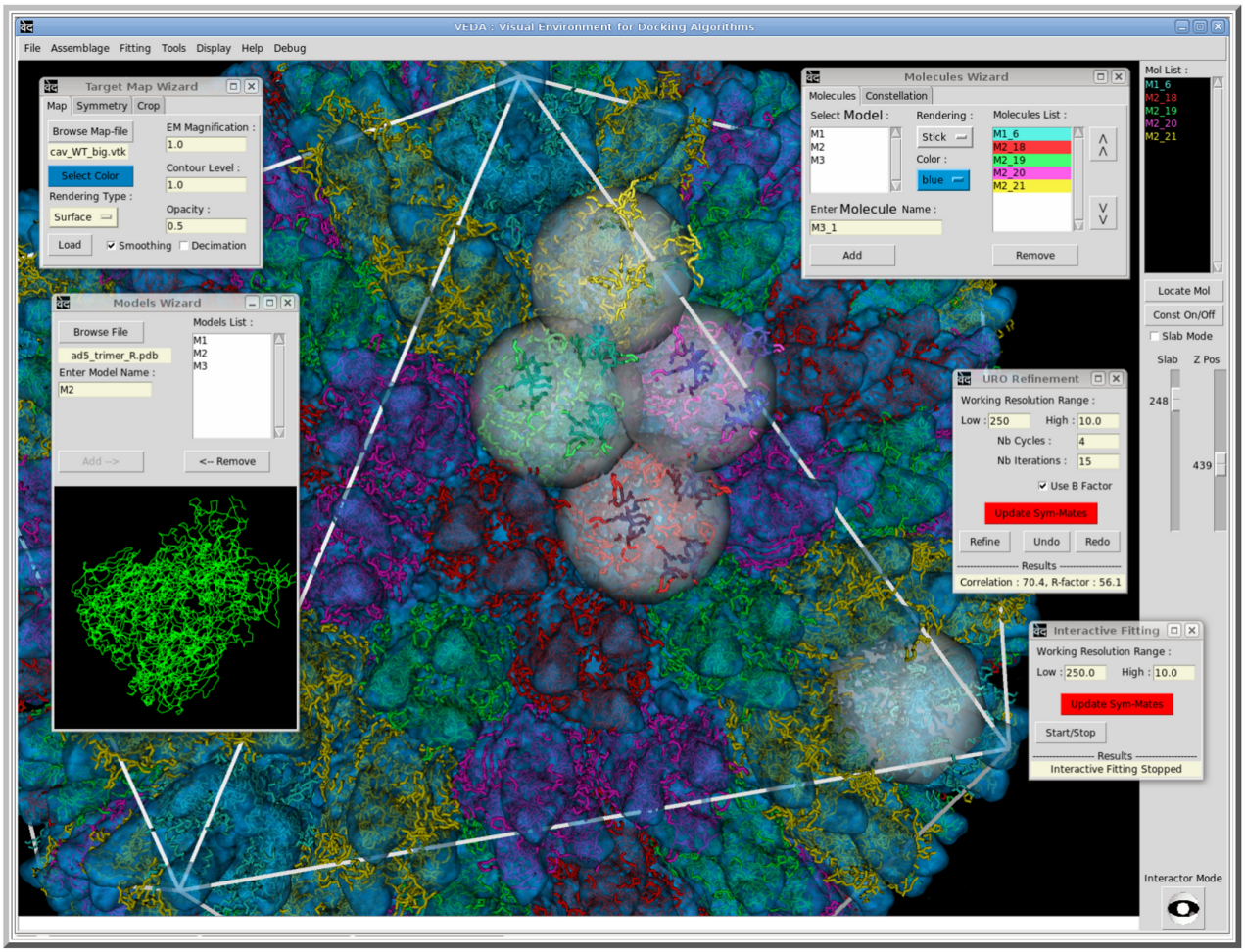
The numerical problem consists in determining and optimizing the variables that specify the positions of the rigid models within the macromolecular assembly.

What is VEDA ?

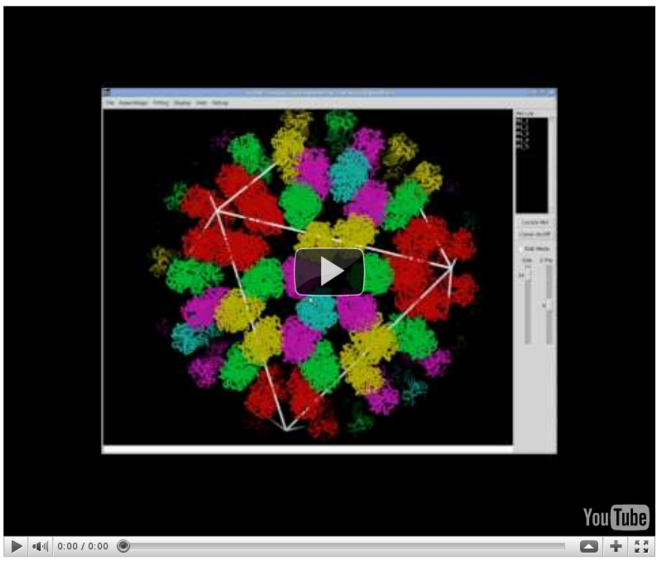
VEDA is a visual environment developed to fit interactively atomic models into 3D reconstructions.

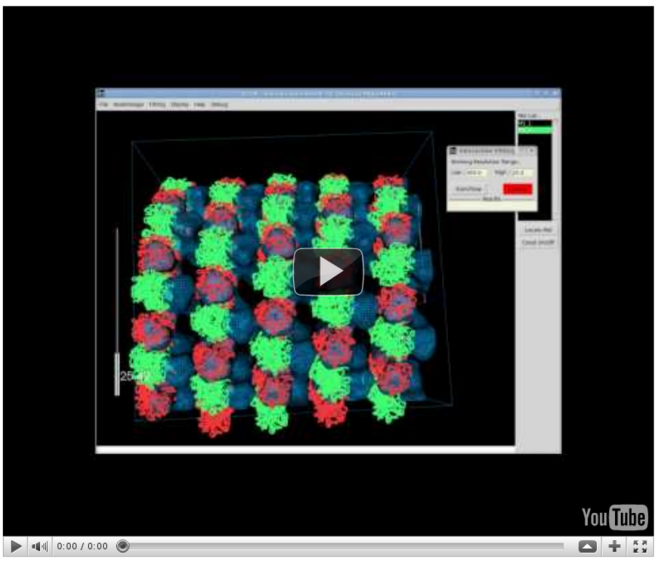
Calculations are performed in reciprocal-space and the symmetry of the reconstruction is taken into account.

The computations are fast and an entire EM reconstruction can be used.



VEDA working !





Download and Install VEDA

VEDA_lin_64_Linux version 22-12-2010 (compiled with gfortran)

VEDA_lin_32_Linux version 22-12-2010 (compiled with gfortran)

The OSX version and stand-alone packages are comming soon ...

Installation Guide (for Linux Ubuntu)
User Guide (Beta Version)
Examples (small tubes and dlp Rotavirus, contain density files and PDBs)

6.3 Publications

Structure of RavA MoxR AAA+ protein reveals the design principles of a molecular cage modulating the inducible lysine decarboxylase activity

Majida El Bakkouri[a], Irina Gutsche[b], Usheer Kanjee[a], Boyu Zhao[a], Miao Yu[a], Gael Goret[c], Guy Schoehn[b,c], Wim P. Burmeister[b,d], and Walid A. Houry[a,1]

[a]Department of Biochemistry, University of Toronto, Toronto, ON, Canada M5S 1A8; [b]Unit for Virus Host-Cell Interactions (UVHCI), Unité Mixte de Internationale (UMI) 3265, Université Joseph Fourier (UJF)-European Molecular Biology Laboratory (EMBL) Grenoble Outstation-Centre National de la Recherche Scientifique (CNRS), 6 rue Jules Horowitz, 38042 Grenoble, France; [c]Institut de Biologie Structurale (IBS) Jean-Pierre Ebel, Unité Mixte de Recherche (UMR) 5075, Commissariat à l'Énergie Atomique (CEA)-CNRS-UJF, 41 rue Jules Horowitz, 38027 Grenoble, France; and [d]Institut Universitaire de France (IUF), 103 Boulevard St. Michel, 75005 Paris, France

Edited by Gregory A. Petsko, Brandeis University, Waltham, MA, and approved November 5, 2010 (received for review June 24, 2010)

The MoxR family of AAA+ ATPases is widespread throughout bacteria and archaea but remains poorly characterized. We recently found that the *Escherichia coli* MoxR protein, RavA (Regulatory ATPase variant A), tightly interacts with the inducible lysine decarboxylase, LdcI/CadA, to form a unique cage-like structure. Here, we present the X-ray structure of RavA and show that the αβα and all-α subdomains in the RavA AAA+ module are arranged as in magnesium chelatases rather than as in classical AAA+ proteins. RavA structure also contains a discontinuous triple-helical domain as well as a β-barrel-like domain forming a unique fold, which we termed the LARA domain. The LARA domain was found to mediate the interaction between RavA and LdcI. The RavA structure provides insights into how five RavA hexamers interact with two LdcI decamers to form the RavA-LdcI cage-like structure.

acid stress | alarmone

Proteins of the AAA+ superfamily (ATPases Associated with diverse cellular Activities) are highly ubiquitous and found in all kingdoms of life. These proteins are characterized by the structural conservation of a central ATPase domain of about 250 amino acids called the AAA+ module (1, 2). AAA+ ATPases employ the energy derived from ATP hydrolysis to remodel proteins, DNA, or RNA. Typically, the AAA+ domain can be divided into two structural subdomains, an N-terminal P-loop NTPase αβα subdomain that is connected to a smaller C-terminal all-α subdomain. The αβα subdomain adopts a Rossman fold and contains several motifs involved in ATP binding and hydrolysis, including Walker A, Walker B, and Sensor 1 signature sequences (3–6). The all-α subdomain, which contains the Sensor 2 motif (7), is much less conserved across AAA+ proteins.

AAA+ proteins form oligomers, usually hexameric rings, in the presence of nucleotides (8). The ATP-binding pocket is located at the interface between two neighboring subunits. A highly conserved arginine from one subunit, called an "arginine finger," contacts the γ-phosphate of bound ATP of the neighboring subunit (9). AAA+ proteins typically go through a cycle of ATP binding, hydrolysis, and release of products. This reaction cycle results in a series of conformational changes and mechanical movements that allow these proteins to exert their activity either directly or through domains attached to the AAA+ domain (3, 10).

The RavA protein (Regulatory ATPase Variant A) belongs to the MoxR AAA+ family (11). Limited experimental data suggest a function of MoxR AAA+ proteins as chaperones in the assembly of multimeric complexes and a possible role in small molecule cofactor insertion/removal (11). However, how these proteins act is not clear. In *Escherichia coli*, the *ravA* gene is in an operon with another gene of unknown function, which we termed *viaA*, and the operon is under the control of σ^S promoter, suggesting a function of RavA and ViaA under stress conditions (12). This is further substantiated by our discovery that RavA physically interacts with the inducible lysine decarboxylase enzyme, LdcI/CadA, a key enzyme in the bacterial acid stress response. We have visualized the LdcI-RavA complex by negative staining electron microscopy and found it to form a large, about 3.3 MDa, unusual cage-like structure consisting of two LdcI decamers that are linked by up to five RavA hexamers (12). Because LdcI is fivefold symmetric whereas RavA is sixfold symmetric, understanding the construction of this complex is important to understanding how the symmetry mismatch was used in forming the cage-like structure.

We recently solved the X-ray crystal structure of LdcI decamer and unexpectedly found that LdcI activity is strongly inhibited by the binding of the alarmone, ppGpp (further details on the LdcI structure will be described elsewhere). Here, we have determined the three-dimensional structure of RavA full-length protein as a monomer in complex with ADP by X-ray crystallography. Insights into the intersubunit organization of the hexameric RavA were obtained from electron microscopy. These structures provided important insights into how nature solved the symmetry mismatch problem between the fivefold symmetric LdcI decamer and sixfold symmetric RavA hexamer to allow for the construction of the RavA-LdcI molecular cage-like structure. We show that the RavA-LdcI interaction reduces the inhibition of LdcI activity by the alarmone ppGpp in vitro as well as in vivo. The biological implications of this interaction are discussed.

Results

The Overall Structure of RavA Protomer. RavA full-length recombinant protein was expressed and purified to homogeneity as previously described (12). Purity was tested by mass spectrometry and light scattering. RavA crystals were obtained several years ago but failed to diffract to better than 7 Å resolution. High-quality crystals were finally obtained by employing a dehydration protocol as described in *Materials and Methods*. The protein crystallized in the space group $P6_5$ with one molecule in the asymmetric unit. The crystal structure was solved to 2.9 Å resolution. The model includes 475 of the 498 residues of RavA and one bound ADP molecule, but no electron density corresponding to a Mg^{2+} ion was found (Table S1). Two segments, 88–97 and

Author contributions: M.E.B., I.G., U.K., G.S., W.P.B., and W.A.H. designed research; M.E.B., I.G., U.K., B.Z., M.Y., and W.A.H. performed research; M.E.B., I.G., U.K., B.Z., G.G., and W.A.H. analyzed data; and M.E.B., I.G., U.K., B.Z., and W.A.H. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

Data deposition: The crystallography, atomic coordinates, and structure factors have been deposited in the Protein Data Bank, www.pdb.org (PDB ID code 3NBX).

[1]To whom correspondence should be addressed. E-mail: walid.houry@utoronto.ca.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1009092107/-/DCSupplemental.

438–441 (Fig. 1*A*), are missing in the final electron density map and are indicated by dotted lines in Fig. 1*B*.

The RavA monomer has a complex elongated overall structure consisting of three distinct domains (Fig. 1 *A* and *B*). The N-terminal domain of RavA is the AAA+ module, which is composed of two subdomains: the αβα subdomain (residues 1–192, brown) and the all-α subdomain (residues 226–306, wheat). The αβα subdomain exhibits a Rossmann-type fold commonly found in nucleotide binding proteins. It consists of a central β-sheet with five parallel β-strands, ordered as 51432, sandwiched between seven α-helices. The all-α subdomain consists of four antiparallel α-helices. The αβα subdomain and the all-α subdomain are linked by a 32-residue helical segment (residues 193–225, green). The relative arrangement of the subdomains is similar to that found in Mg chelatases (discussed below).

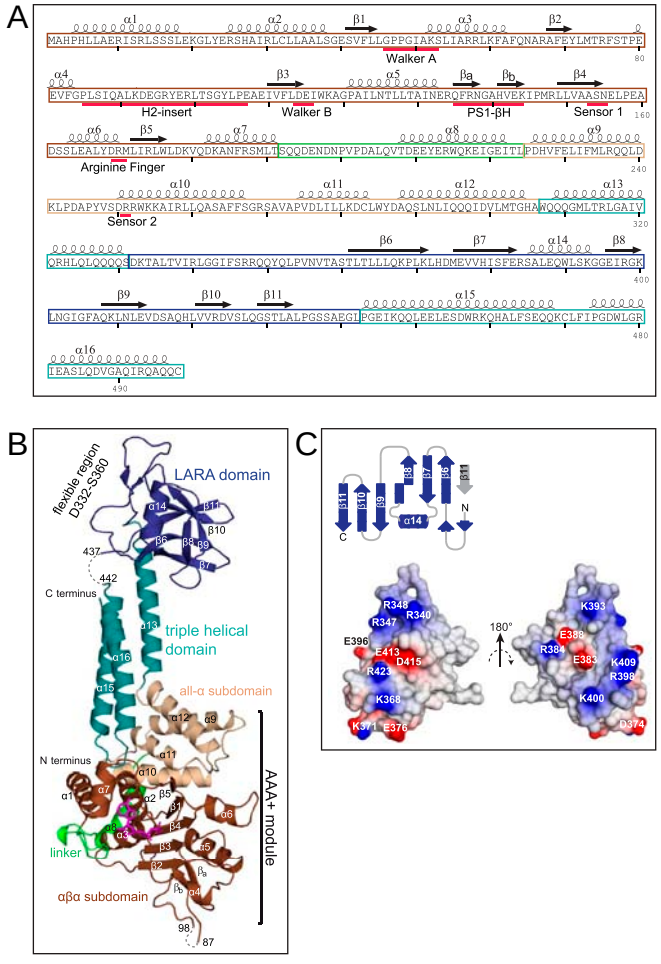


Fig. 1. Overall view of RavA protomer structure. (*A*) Sequence of *E. coli* RavA showing secondary structure and conserved motifs. (*B*) X-ray structure of RavA protomer. αβα subdomain is shown in brown, all-α subdomain is shown in wheat, the linker between the two subdomains is shown in green, triple-helical bundle domain is shown in blue, the LARA domain is shown in dark blue, and bound ADP is shown in violet. The α-helices and β-strands are labeled sequentially except for β_a and β_b of the Pre-Sensor 1 β-Hairpin insertion. Residues 88–97 and 438–441 were not observed in the X-ray structure and are indicated by a dashed line. The figure was generated using PYMOL. (*C*) Shown is a topological diagram of the LARA domain drawn using TopDraw (25) and its electrostatic surface potential calculated using Delphi (26). Colors are according to the calculated electrostatic surface potential and range from red (potential of −5 kT) to blue (+5 kT). The hydrophobic core of the domain is made by the side chains of hydrophobic residues from each of the β-strands (β1: L362, L364, L366, L372; β2: V377, I380, F382; β3: I397, L401; β4: L410, L412; β5: L420, V422; β6: L432) as well as residues L387, W390 and L391 from the α14 helix.

The second domain is a discontinuous triple-helical domain formed by helices α13, α15, and α16 (residues 307–330 and 442–497, light blue). This domain has a rigid structure stabilized by hydrophobic interactions localized at the interface between the three helices. The third domain (residues 331–437, dark blue), which we have named the LARA domain (for reasons described below), is a protuberance between helices α13 and α15 of the triple-helical domain. As shown in Fig. 1 *B* and *C*, the LARA domain forms a compact antiparallel β-barrel-like structure consisting of six β-strands (β6–β11) and one α-helix (α14). The LARA domain also includes an N-terminal flexible region (residues D332–S360). The domain is very basic (pI of 9.6) resulting from a highly positively charged surface formed by residues R340, R347, R348, R398, K400, K409, and R423 (Fig. 1*C*). We performed an extensive search for structures similar to that of the LARA domain in the Protein Data Bank using secondary-structure matching (SSM) (13) and Dali (14), but no such structures were found. Hence, we conclude that the *E. coli* RavA LARA domain adopts a unique fold.

The sequence conservation at the C terminus of RavA spanning the triple-helical and LARA domains diverges quite quickly, although, according to secondary-structure prediction, all organisms containing RavA protein with a LARA domain also have a triple-helical domain. It was surprising to find that a phenylalanine (F472), located at the turn between helices α15 and α16 of the triple-helical domain, is absolutely conserved (Fig. S1*A*). This phenylalanine makes hydrophobic contacts with the AAA+ module and, hence, anchors the triple-helical bundle to the AAA+ module (Fig. S1 *B* and *C*). F472 might serve to transmit the nucleotide-dependent conformational changes in the AAA+ domain to the C-terminal triple-helical and LARA domains of RavA.

RavA Hexameric Assembly. Although RavA and many other AAA+ ATPases crystallize as monomers, their functional form is well known to be an oligomeric ring structure. Previous work in our laboratory provided first evidence for a hexameric assembly of RavA induced by ATP, ADP, or 5′-adenylyl-β,γ-imidodiphosphate (AMPPNP) binding (12). Here we present a 3D structure of the RavA hexamer formed in the presence of ADP obtained by negative staining electron microscopy (EM) and image analysis.

Similar to other AAA+ protein structures, hexameric RavA-ADP is characterized by a ring-shaped core surrounding a central pore. Some representative class averages, as well as corresponding projections of the 3D structure at similar orientations, are shown in Fig. 2*A*. The distinctive feature of the class averages is the relatively weak density of the LARA domain, which necessitated a good alignment in order to be properly visualized (see *Materials and Methods* for details). The RavA hexameric ring forms a rather unique flower-like structure and is found to be about 220 Å in diameter and of 80-Å thickness, whereas the diameter of the central channel is about 25 Å. The 3D reconstruction possesses a prominent handedness, visible in the core AAA+ part and notably accentuated by the protrusions. An atomic model of the hexamer was then generated by docking the crystal structure of the monomer into the EM density of the hexamer and adjustment of the resulting intersubunit contacts based on a homology model generated from the hexameric crystal structure of HslU [PDB ID code 1DO0 (15)] (see *Materials and Methods* and Fig. S2 for details). The final EM reconstruction and the resulting atomic model of the RavA-ADP hexamer are shown in Fig. 2*B*.

The Organization of the AAA+ Motor Subdomains. In their classification of AAA+ proteins, Aravind and coworkers grouped RavA within the helix 2 insert clade (8). Members of this family are found to have (*i*) an insert within helix 2 of the conserved ASCE (refers to Additional Strand, Catalytic E) division P-loop ATPase core, (*ii*) a β-hairpin N terminal to Sensor 1, as well as, (*iii*) a long helical seg-

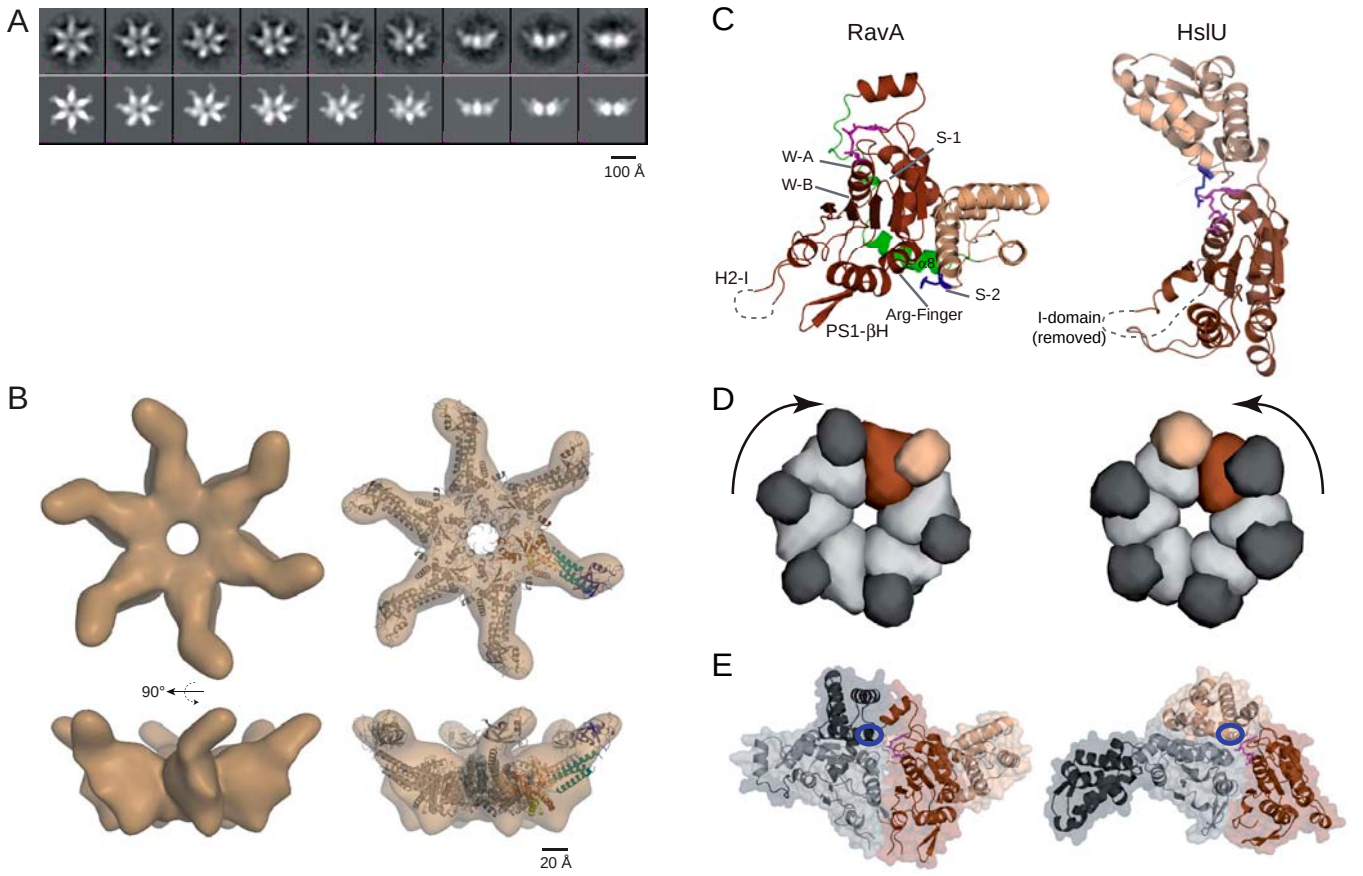


Fig. 2. RavA-ADP hexamer structure (*A*, *Top*) Characteristic class averages of the negatively stained RavA-ADP hexamer; (*Bottom*) projections of the final 3D reconstruction at similar orientations. (*B*) Top and side views of the EM 3D reconstruction of the RavA-ADP hexamer. An atomic model of RavA hexamer was generated from the X-ray structure of the RavA protomer by docking into the EM envelope of the hexamer and comparison with the X-ray structure of the HslU hexamer (PDB ID code 1DO0). (*C*) Ribbon representation of RavA AAA+ module (*Left*) and HslU AAA+ module (*Right*, PDB ID code 1DO0). Different subdomains are colored as in Fig. 1 and conserved motifs are shown; the Sensor 2 motif is colored in dark blue and the nucleotide is shown in violet. The I-domain of HslU has been omitted for clarity. (*D*, *Left*) Schematic representation of RavA AAA+ domain from the hexameric model viewed along the sixfold axis. (*Right*) X-ray structure of HslU hexamer AAA+ domain. For each structure, the αβα and all-α subdomains of one protomer are colored in brown and wheat, respectively. The other protomers are colored in light and dark gray for the αβα and all-α subdomains, respectively. (*E*) Space-filling and ribbon models of a representative dimer of each hexamer in *D*. Nucleotide is shown in violet, whereas the blue circle indicates the location of the Sensor 2 motif.

ment separating the C terminus of the αβα subdomain and the N terminus of the all-α subdomain. The X-ray structure of the RavA AAA+ module agrees with this classification. The αβα subdomain contains two characteristic β-hairpin insertions (Figs. 1*A* and 2*C*). The helix-2 insert (H2-I, residues Pro85–Pro106) is incorporated within helix α4, but is partially unstructured in our model (shown as a dotted line in Figs. 1*B* and 2*C*). In the case of the Mg chelatase subunit BchI, this insert consists of two β-strands flanking a small helix (16). The Pre-Sensor 1 β-Hairpin insertion (PS1-βH, residues Gln135–Pro146) is incorporated between the Sensor 1 strand (β4) and the preceding helix (α5). It has been shown that PS1-βH and H2-I are usually important for substrate interaction (17–20). The long helical segment between the αβα and all-α subdomains is shown in green in Figs. 1 *A* and *B* and 2*C*; this segment consists of two helices that wrap around the surface of the αβα subdomain making several contacts with it.

SSM and Dali structural similarity searches using the whole RavA AAA+ module indicated only four proteins to have similar structures as the RavA AAA+ domain: the putative ATPase from *Cytophaga hutchinsonii*, BchI subunit of *Rhodobacter capsulatus* Mg chelatase, archaeal minichromosome maintenance protein MCM from *Sulfolobus solfataricus*, and an archaeal MCM homolog from *Methanopyrus kandleri* (Fig. S3). In all of these proteins, the spatial localization of the all-α subdomain relative to the αβα subdomain is similar to that of RavA and is significantly different from its typical position found in most other AAA+ members (8, 21).

Fig. 2*C* illustrates the differences between the AAA+ module organization of HslU, a canonical model of the typical AAA+ module fold (15), and RavA AAA+ domain. In HslU, the all-α subdomain is located on "top" of the αβα subdomain. In the all-α subdomain, the Sensor 2 motif (shown in dark blue in Fig. 2*C*), a required motif for nucleotide binding and hydrolysis, is oriented toward the ATP-binding site of the same protomer. In the case of RavA, the all-α subdomain is on the "right" of the αβα subdomain; consequently, the Sensor 2 motif cannot contribute to ATP hydrolysis on the same protomer as in HslU. However, when the X-ray structure of the HslU hexamer is compared to the RavA hexameric model (Fig. 2*D*), then it is clear that HslU and RavA share similar organization of the subdomains in the oligomer although the orientation of the protomers is reversed. In the RavA hexamer, the all-α subdomain of one protomer is located on top of the αβα subdomain of the protomer on its right (Fig. 2 *D* and *E*), and the Arg residue of Sensor 2 faces the ATP-binding site of a neighboring protomer (Fig. 2*E* and Fig. S4). In the HslU hexamer, the all-α subdomain of one protomer is located on top of the αβα subdomain of the protomer on its left (Fig. 2 *D* and *E*), and the Arg residue of Sensor 2 is facing the ATP-binding site of the same protomer (Fig. 2*E* and Fig. S4). Hence, when viewed from the top, the subunits in RavA are organized clockwise,

whereas the subunits in HslU are organized counterclockwise (Fig. 2*D*).

It should be noted that, in the RavA structure, the all-α and αβα subdomains from the same monomer make extensive interactions. The buried surface area between these two subdomains is much larger in the case of RavA (2,797 Å^2) than for HslU monomer (1,084 Å^2) (Fig. 2*E*). As a result, in the HslU hexamer, the all-α subdomain makes more extensive interactions with the αβα subdomain of the neighboring protomer (2,930 Å^2) than in the RavA hexamer (754 Å^2).

Even with these major differences in the assembly of the subunits, the overall structures of the RavA and HslU AAA+ hexamers remain similar with a high conservation in the organization of the ATP-binding site. Fig. S4 shows the localization of the nucleotide between two subunits of the RavA and HslU hexamers. In both cases, the nucleotide makes contact with three subdomains. For HslU, the nucleotide is sandwiched between the αβα and the all-α subdomains of the same subunit and faces the αβα subdomain of the left neighboring subunit, whereas the nucleotide in RavA contacts the αβα subdomain of one subunit and faces the all-α and the αβα subdomains of the left neighboring subunit.

The LARA Domain Mediates RavA-LdcI Interactions. Previous work in our laboratory has shown that RavA interacts with LdcI, an inducible lysine decarboxylase enzyme, forming an unusual cage-like complex of about 3.3 MDa consisting of two LdcI (81 kDa) decamers and up to five RavA (56 kDa) hexamers (Fig. 3*A*) (12). We further confirmed this interaction in this study. The pull-down of RavA from an *E. coli* strain in which a Sequential Peptide Affinity (SPA) tag (22) was fused at the 3′ end of the endogenous *ravA* gene, also pulled down LdcI as previously observed (12). Analysis of the complex by size exclusion chromatography showed that the majority of RavA was part of a 3.3-MDa complex with LdcI (Fig. S5*A*), which corresponds to the mass of the complex shown in Fig. 3*A*. LdcI migrated as a complex with RavA but also as uncomplexed decamers as well. In another experiment, analysis of the interaction between purified RavA and LdcI proteins by sedimentation velocity analytical ultracentrifugation revealed the presence of 0.8-, 2.7-, and 6.0-MDa complexes (Fig. S5*B*). These complexes would correspond to LdcI decamer alone, the RavA-LdcI cage-like complex of Fig. 3*A*, and a dimer of the cage-like complex, respectively.

Docking of RavA hexameric model of Fig. 2*B* and LdcI decameric crystal structure that we recently determined into the EM envelope suggested that the LARA domain of RavA might interact with LdcI (Fig. 3*A* and Fig. S6*A*). To determine the validity of the docking model, a RavA construct was made in which the LARA domain was deleted and was termed RavAΔLARA (consisting of residues Met1–Ala335 and Leu434–Cys498). An isolated LARA domain construct was also generated (residues Gln329–Glu440). Circular dichroism analysis showed that both proteins have the expected secondary-structure content. Furthermore, RavAΔLARA formed a hexameric complex in the presence of ATP (Fig. S6*B*), and its ATPase activity was similar to that of WT RavA (Fig. S6*B*, *Inset*). The interaction of LdcI with RavA and its different constructs was then assessed by surface plasmon resonance (SPR) using the Biacore system. In these experiments, LdcI was immobilized on the chip. The SPR experiments clearly showed that, although WT RavA and LARA domain do interact with LdcI (Fig. 3*B*), RavAΔLARA does not bind to LdcI (Fig. 3*C*). In the absence of nucleotide, WT RavA bound LdcI with an apparent binding constant of 0.56 μM (Fig. 3*B*). In the presence of ATP, the binding curve was best fit using two independent binding sites with apparent binding constants of 0.018 μM and 1.22 μM (Fig. 3*B*). This might indicate that the proper hexamerization of RavA, which is attained in the presence of ATP, allows for two RavA"legs" to bind the LdcI decamer at two different sites as suggested by the docking analysis of Fig. 3*A* and Fig. S6*A*. However, it should be noted that, because of the experimental setup, the full complex shown in Fig. 3*A*, in which RavA hexamers can bridge two LdcI decamers, is unlikely to form under the conditions of the SPR experiments because LdcI is cross-linked to the chip. The isolated LARA domain is also able to bind LdcI albeit with a lower apparent K_d of 54 μM (Fig. 3*B*). Hence, these observations strongly suggest that the LARA domain is the RavA domain required for LdcI interaction, hence the acronym LARA: LdcI associating domain of RavA.

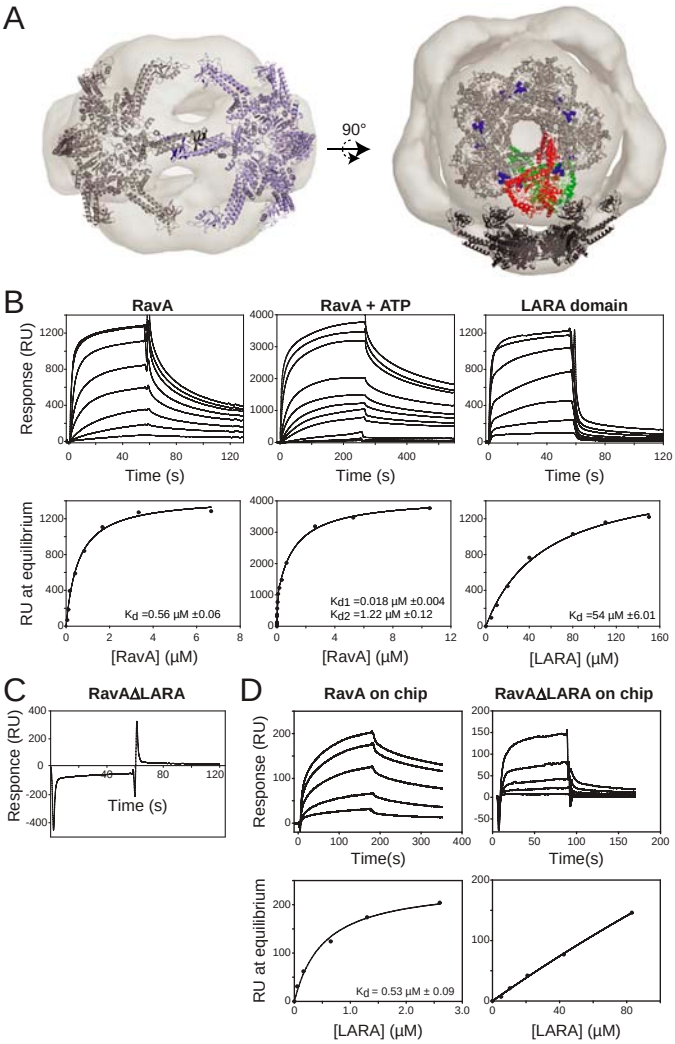


Fig. 3. The LARA domain mediates RavA-LdcI and RavA-RavA interactions. (*A*) Fit of the RavA hexameric model and LdcI decamer into the EM envelope of the LdcI-RavA complex (12) viewed from the side (*Left*) and the top (*Right*). One LdcI dimer is colored in red (the upper monomer) and green (the lower monomer). ppGpp bound to LdcI is drawn as blue spheres. (*B*) Biacore sensorgrams and equilibrium binding curves showing the interaction between LdcI (on chip) and WT RavA in the absence of nucleotide (*Left*), or WT RavA in the presence of ATP (*Middle*), or the LARA domain (*Right*). (*C*) Biacore sensorgram showing the lack of interaction between LdcI (on chip) and RavAΔLARA at 15 μM. (*D*) Biacore sensorgrams and equilibrium binding curves showing the interaction between WT RavA and LARA domain (*Left*) and the very weak interaction between RavAΔLARA and LARA domain (*Right*).

Bioinformatic analysis provided further support for the finding that the LARA domain mediates the interaction of RavA with LdcI. In 47 representative bacterial strains that contained RavA based on BLAST searches (23), we asked whether these strains also contain LdcI. As mentioned earlier, the C-terminal fragment of RavA including the triple-helical bundle and LARA domain is not well conserved. Hence, the presence of the LARA domain in RavA across the different strains was assessed using JPred sec-

ondary structure prediction program (24). It was interesting to find that RavA in all strains containing LdcI has a LARA domain, whereas strains that do not have LdcI contain RavA that has or does not have the LARA domain (Fig. S6*C*). The LARA domain in these strains might have other functions or the domain will eventually degenerate.

The LARA Domain Mediates RavA-RavA Interactions. The docking shown in Fig. 3*A* suggests that the LARA domain might also mediate RavA-RavA interactions within the RavA-LdcI cage-like complex by interacting with the triple-helical domain and/or the LARA domain of the neighboring RavA (Fig. S6*A*). SPR experiments were carried out in which RavA or RavAΔLARA is immobilized on the chip and the LARA domain is titrated. The results indicate that the LARA domain can bind to RavA with an apparent K_d of about 0.5 μM (Fig. 3*D*). This interaction is drastically reduced when the binding experiment is performed between RavAΔLARA and the LARA domain (Fig. 3*D*), suggesting that the LARA domain plays an important role in RavA-RavA interactions within the RavA-LdcI cage-like complex.

RavA Antagonizes the Inhibitory Effect of ppGpp on LdcI Activity. The LdcI crystal structure revealed the presence of a binding site for the bacterial alarmone guanosine tetraphosphate (ppGpp) at the interface between two protomers in the pentameric ring (Fig. 3*A*). Biochemical assays showed that ppGpp binding to LdcI results in drastic inhibition of the LdcI activity of approximately 10-fold at pH values greater than 5. The docking shown in Fig. 3*A* suggests that the LARA domain of RavA might bind at a site in LdcI that could affect ppGpp binding to the decarboxylase.

The activity of the decarboxylase was measured in the presence and absence of RavA and ppGpp using an isothermal titration calorimetry (ITC) approach. Initially, we ensured that the presence of GTP, GDP, ppGpp, and pppGpp does not affect RavA ATPase activity (Fig. 4*A*). When the RavA-LdcI complex is preformed, LdcI activity is not significantly changed consistent with our prior observations (12). However, when ppGpp is added to the preformed RavA-LdcI complex, the presence of RavA reduces the inhibitory effect of ppGpp on LdcI by about 40% under the conditions of the experiment (Fig. 4*B*). Moreover, the RavA-ΔLARA truncation mutant is not able to reduce the inhibitory effect of ppGpp on LdcI activity, which is consistent with our results showing that the LARA domain is responsible for the RavA-LdcI interaction. When LdcI is preincubated with ppGpp and then RavA is added, RavA is not able to reduce the inhibition of LdcI by the alarmone (Fig. 4*C*). Hence, RavA either blocks the access to the ppGpp binding site in LdcI or RavA induces a local conformational change in LdcI that reduces its ppGpp binding affinity. Alternatively, ppGpp might cause a conformational change in LdcI to reduce RavA binding to the decarboxylase.

It should be pointed out that the effect of RavA on LdcI inhibition by ppGpp is probably underestimated because these experiments were done at low concentrations of $MgCl_2$. Under such conditions, RavA has low ATPase activity; addition of higher concentrations of $MgCl_2$ lead to the precipitation of ppGpp by Mg^{2+}.

To further validate our in vitro results and to determine if the modulation of ppGpp binding to LdcI by RavA can be observed in vivo, the activity of LdcI was tested in different strains undergoing a stringent response. Four strains were used: Δ*cadBA*, WT + *rav*A, WT + *ravA*Δ*LARA*, and WT + *ravA(K52Q)*. The last three strains overexpress RavA, RavAΔLARA, and RavA (K52Q) proteins by IPTG induction (refer to *Materials and Methods*). RavA(K52Q) is ATPase deficient because the conserved Walker A K52 is mutated to Q (Fig. 1*A*). Endogenous RavA is expressed at low levels and is induced only in the stationary phase (12). Cells were grown to log phase in defined rich media buffered at pH 5, and, when the OD_{600} of each strain was approximately 0.2, proteins were induced with 1 mM IPTG for 1 h. Cells were then shifted to minimal media weakly buffered at pH 5 containing no amino acids to induce ppGpp production, and supplemented with 30 mM lysine to follow the LdcI activity by monitoring pH change of the media; no cell growth occurs during

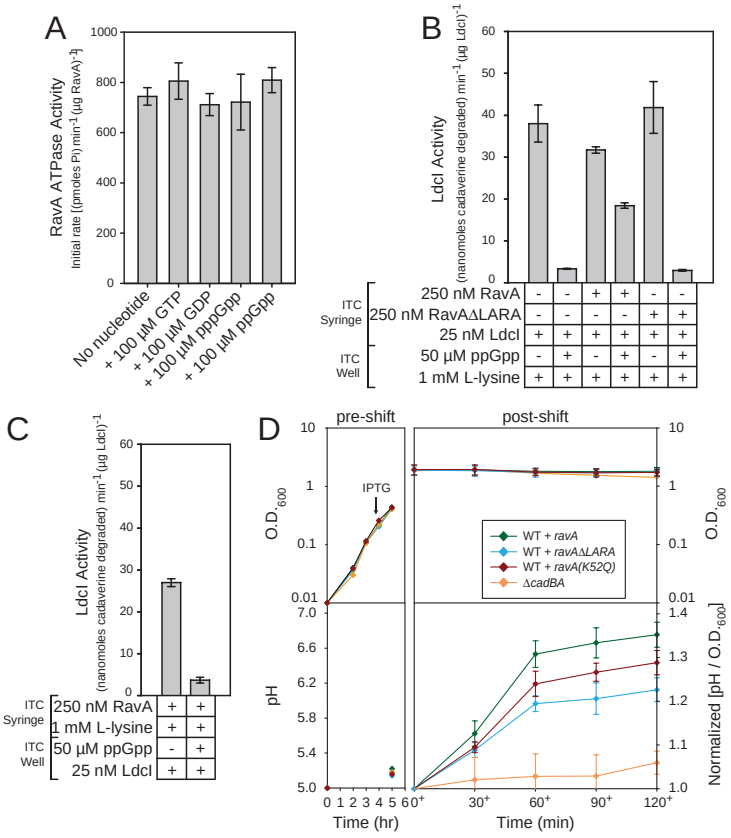


Fig. 4. RavA binding to LdcI antagonizes the inhibitory effect of ppGpp on LdcI activity. (*A*) The ATPase activity of RavA measured by ITC in the presence of different nucleotides. Error bars represent the standard deviation of the average of three experiments. (*B*) LdcI activity measured by ITC in the presence of RavA or RavAΔLARA and/or ppGpp. Note that the concentrations of proteins, substrate, and inhibitor are the final concentrations after mixing. In this panel, the RavA-LdcI complex is preformed in the syringe before adding ppGpp. (*C*) LdcI activity measured by ITC. In this experiment, LdcI-ppGpp complex is preformed in the well before adding RavA. (*D*) The effect of RavA overexpression on LdcI activity in the cell. Δ*cad*BA knockout strains and WT cells overexpressing RavA, RavAΔLARA, or RavA(K52Q) were grown to log phase in defined rich media buffered at pH 5. RavA, RavAΔLARA, or RavA(K52Q) were induced and cells were then shifted to minimal media weakly buffered at pH 5 containing no amino acids to induce ppGpp production and supplemented with 30 mM lysine. The OD_{600} of the cells is shown (*Top*); the pH of the culture media is shown (*Bottom Left*). (*Bottom Right*) The increase in pH/OD_{600} normalized to the value at 0+ (right after shift). Each time point is the result of at least three replicates. Error bars represent the standard deviations of the measurements.

this time (Fig. 4*D* and Fig. S7). Consistent with the in vitro results, WT + *ravA* strain increased the pH of the media at a higher rate than the WT + *ravAΔLARA* strain, whereas no significant pH change was observed for the Δ*cadBA* cells (Fig. 4*D*). Hence, the formation of the RavA-LdcI complex reduced the inhibitory effect of the alarmone on LdcI, allowing the cells to better respond to low acidity. On the other hand, RavAΔLARA cannot form a complex with LdcI (Fig. 3*D*) and, hence, LdcI should still be inhibited by ppGpp resulting in a lower rate of pH increase, as observed (Fig. 4*D*). The strain overexpressing RavA(K52Q) mutant increased pH faster than the strain overexpressing RavA-ΔLARA, but not as well as the strain overexpressing WT RavA (Fig. 4*D*), indicating that the binding of RavA to LdcI is not enough to modulate alarmone binding to the decarboxylase, but that the ATPase activity of RavA is also needed.

Discussion

The organization of the AAA+ module of RavA as revealed by the X-ray structure of the protein (Figs. 1*B* and 2*C*) explicitly demonstrates that the protein is closely related to the family of Mg chelatases. We had previously found that RavA and LdcI interact tightly to form an unusual cage-like structure (12). Having the X-ray structure and the EM reconstruction of RavA (this study), as well as the X-ray structure of LdcI and the negative staining EM reconstruction of the RavA-LdcI complex (12), allowed us to gain important insights into the design principles of this cage that is formed by the interaction of a fivefold symmetric oligomer of LdcI with a sixfold symmetric oligomer of RavA (Fig. 3*A*). The RavA hexamer displays six "legs," which are spanning the triple-helical domain and the LARA domain. Two of the legs interact with an LdcI dimer at the top of the complex, and two other legs show the same set of interactions with an LdcI dimer at the bottom of the complex. These interactions seem to be mainly mediated by the LARA domain. The two remaining legs of RavA are interacting with a neighboring RavA leg on the left and on the right (Fig. 3*A* and Fig. S6*A*). The RavA-RavA leg–leg interactions seem to involve the triple-helical domain, as well as the LARA domain (Fig. 3*A*). Hence, the construction of the RavA leg makes all these interactions possible. The LARA domain exhibits a unique fold and, based on the bioinformatic analysis of Fig. S6*C*, seems to be optimally evolved to mediate the interaction of RavA with LdcI.

The RavA-LdcI cage-like structure might have multiple functions in the cell yet to be elucidated; however, one consequence we found here for the formation of the RavA-LdcI complex is the reduction of the inhibitory effect of ppGpp on LdcI activity. When cells are undergoing acid stress and LdcI is induced to about 2,000 decamers per cell, nutrient limitation will result in the production of the alarmone. Because we estimate that there are about 50–100 RavA hexamers per cell in the stationary phase (12), only a small population of LdcI molecules is expected to be in complex with RavA. This population of LdcI will not be strongly inhibited by ppGpp, allowing the cells to continue to respond to acid stress at the risk of depleting lysine amounts. It is interesting to note that RavA and ppGpp have similar binding constants to LdcI: K_d of 0.02—1 μM for RavA-LdcI interaction (Fig. 3*B*) and K_d of 0.01–0.7 μM for LdcI-ppGpp interaction. The binding of RavA to apo-LdcI does not affect LdcI activity to any significant extent (Fig. 4*B* and ref. 12). Hence, there is a fine-tuning of LdcI activity by ppGpp and RavA, which is required for the cells to respond to acid stress, as well as to prevent the depletion of their amino acids. This fine-tuning probably involves other factors and proteins and also occurs for other amino acid decarboxylases involved in the bacterial acid stress response.

Materials and Methods

Details of cloning, protein expression and purification, RavA ATPase assay, SPR measurements, LdcI enzyme kinetics measurements using ITC, sedimentation velocity analytical ultracentrifugation, pull-down experiments, media shift assays, X-ray crystallography, and electron microscopy are provided in *SI Text*.

ACKNOWLEDGMENTS. We thank Dr. Elisabeth Tillier (University of Toronto) for her help with the bioinformatic analysis shown in Fig. S6*C*. We also thank several members of UVHCI: Dr. Rob Ruigrok for his support, Dr. Hassan Belrhali for help with data collection and phasing, Dr. Monika Spano, Dr. Nicolas Tarbouriech, and Dr. Thibaut Crépin for help with protein crystallization. M.E.B. is the recipient of a fellowship from the Canadian Institutes of Health Research (CIHR) Strategic Training Program in Protein Folding and Interaction Dynamics: Principles and Diseases. U.K. is the recipient of a National Sciences and Engineering Research Council of Canada (NSERC) Postgraduate Scholarship, a Canadian Institutes of Health Research Strategic Training Program in the Structural Biology of Membrane Proteins Linked to Disease, and a University of Toronto Open Fellowship. B.Z. is the recipient of NSERC Alexander Graham Bell Canada Graduate Scholarship. M.Y. is the recipient of Life Sciences Award from the University of Toronto. This work was supported by a grant from CIHR (MOP-67210) (to W.A.H.).

1. Kunau WH, et al. (1993) Two complementary approaches to study peroxisome biogenesis in Saccharomyces cerevisiae: Forward and reversed genetics. *Biochimie* 75:209–224.
2. Confalonieri F, Duguet M (1995) A 200-amino acid ATPase module in search of a basic function. *Bioessays* 17:639–650.
3. Ogura T, Wilkinson AJ (2001) AAA+ superfamily ATPases: Common structure—diverse function. *Genes Cells* 6:575–597.
4. Leipe DD, Koonin EV, Aravind L (2003) Evolution and classification of P-loop kinases and related proteins. *J Mol Biol* 333:781–815.
5. Ammelburg M, Frickey T, Lupas AN (2006) Classification of AAA+ proteins. *J Struct Biol* 156(1):2–11.
6. Snider J, Thibault G, Houry WA (2008) The AAA+ superfamily of functionally diverse proteins. *Genome Biol* 9:216.
7. Neuwald AF, Aravind L, Spouge JL, Koonin EV (1999) AAA+: A class of chaperone-like ATPases associated with the assembly, operation, and disassembly of protein complexes. *Genome Res* 9:27–43.
8. Iyer LM, Leipe DD, Koonin EV, Aravind L (2004) Evolutionary history and higher order classification of AAA+ ATPases. *J Struct Biol* 146:11–31.
9. Ogura T, Whiteheart SW, Wilkinson AJ (2004) Conserved arginine residues implicated in ATP hydrolysis, nucleotide-sensing, and inter-subunit interactions in AAA and AAA+ ATPases. *J Struct Biol* 146:106–112.
10. Hanson PI, Whiteheart SW (2005) AAA+ proteins: Have engine, will work. *Nat Rev Mol Cell Biol* 6:519–529.
11. Snider J, Houry WA (2006) MoxR AAA+ ATPases: A novel family of molecular chaperones? *J Struct Biol* 156:200–209.
12. Snider J, et al. (2006) Formation of a distinctive complex between the inducible bacterial lysine decarboxylase and a novel AAA+ ATPase. *J Biol Chem* 281:1532–1546.
13. Krissinel E, Henrick K (2004) Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallogr D* 60:2256–2268.
14. Holm L, Kaariainen S, Rosenstrom P, Schenkel A (2008) Searching protein structure databases with DaliLite v.3. *Bioinformatics* 24:2780–2781.
15. Bochtler M, et al. (2000) The structures of HslU and the ATP-dependent protease HslU-HslV. *Nature* 403:800–805.
16. Fodje MN, et al. (2001) Interplay between an AAA module and an integrin I domain may regulate the function of magnesium chelatase. *J Mol Biol* 311:111–122.
17. Han YW, et al. (2001) A unique beta-hairpin protruding from AAA+ ATPase domain of RuvB motor protein is involved in the interaction with RuvA DNA recognition protein for branch migration of Holliday junctions. *J Biol Chem* 276:35024–35028.
18. Lee SY, et al. (2003) Regulation of the transcriptional activator NtrC1: Structural studies of the regulatory and AAA+ ATPase domains. *Genes Dev* 17:2552–2563.
19. Shen J, Gai D, Patrick A, Greenleaf WB, Chen XS (2005) The roles of the residues on the channel beta-hairpin and loop structures of simian virus 40 hexameric helicase. *Proc Natl Acad Sci USA* 102:11248–11253.
20. Jenkinson ER, Chong JP (2006) Minichromosome maintenance helicase activity is controlled by N- and C-terminal motifs and requires the ATPase domain helix-2 insert. *Proc Natl Acad Sci USA* 103:7613–7618.
21. Erzberger JP, Berger JM (2006) Evolutionary relationships and structural mechanisms of AAA+ proteins. *Annu Rev Biophys Biomol Struct* 35:93–114.
22. Babu M, et al. (2009) Sequential peptide affinity purification system for the systematic isolation and identification of protein complexes from Escherichia coli. *Methods Mol Biol* 564:373–400.
23. Altschul SF, et al. (1997) Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res* 25:3389–3402.
24. Cole C, Barber JD, Barton GJ (2008) The Jpred 3 secondary structure prediction server. *Nucleic Acids Res* 36:W197–201.
25. Bond CS (2003) TopDraw: A sketchpad for protein structure topology cartoons. *Bioinformatics* 19:311–312.
26. Rocchia W, et al. (2002) Rapid grid-based construction of the molecular surface and the use of induced surface charge to calculate reaction field energies: Applications to the molecular systems and geometric objects. *J Comput Chem* 23:128–137.

JOURNAL OF VIROLOGY, Feb. 2011, p. 1391–1395 Vol. 85, No. 3
0022-538X/11/$12.00 doi:10.1128/JVI.01459-10

Nucleoprotein-RNA Orientation in the Measles Virus Nucleocapsid by Three-Dimensional Electron Microscopy▽†

Ambroise Desfosses,[1] Gaël Goret,[2] Leandro Farias Estrozi,[1] Rob W. H. Ruigrok,[1] and Irina Gutsche[1]*

UMI 3265 UJF-EMBL-CNRS, Unit for Virus Host Cell Interactions, Grenoble, France,[1] and UMR 5075 CEA-CNRS-UJF, Institut de Biologie Structurale Jean-Pierre Ebel, Grenoble, France[2]

Received 13 July 2010/Accepted 15 November 2010

Recombinant measles virus nucleoprotein-RNA (N-RNA) helices were analyzed by negative-stain electron microscopy. Three-dimensional reconstructions of trypsin-digested and intact nucleocapsids coupled to the docking of the atomic structure of the respiratory syncytial virus (RSV) N-RNA subunit into the electron microscopy density map support a model that places the RNA at the exterior of the helix and the disordered C-terminal domain toward the helix interior, and they suggest the position of the six nucleotides with respect to the measles N protomer.

The RNA genome of nonsegmented negative-strand RNA viruses is tightly and regularly encapsidated by the viral nucleoprotein N, providing flexible helical templates for viral transcription and replication. Upon heterologous expression, nucleoproteins associate not only with long cellular RNAs to form helical nucleocapsids undistinguishable from the viral ones but also with short cellular RNAs that noncovalently close up into N-RNA rings. In the rings, N-RNA is sterically constrained in a biologically inactive form, but the rings have an advantage of being rigid enough for X-ray crystallography. Conversely, the helical assemblies are challenging for electron microscopy (EM) analysis because of their flexibility but are the biologically relevant ones.

The atomic structures of N-RNA rings of rabies virus and vesicular stomatitis virus (both rhabdoviruses) (1, 10) reveal the shielding of RNA between two domains of N in a positively charged cleft situated inside the rings. Extended N- and C-terminal domains reach out to neighboring N protomers in order to stabilize and rigidify the structure. Recently, the structure of N-RNA rings of respiratory syncytial virus (RSV; a paramyxovirus) was determined (24). The global architecture of the nucleoprotein is very similar to that of the rhabdoviruses, although there are 7 ribonucleotides (nt) instead of 9 bound to each N protomer. However, the lateral contacts between adjacent N subunits of the ring confer to it an opposite curvature, which results in an outward RNA groove location. RSV N has an N-terminal exchange domain similar to that of rhabdovirus N, but the C-terminal domain is slightly different, as it is not clearly involved in contacts between subsequent N protomers. Is this inversion of the subunit orientation due simply to steric constraints in the ring, or does it also take place in a helical nucleocapsid? Tawar and coworkers modeled an RSV N-RNA helix but could not directly dock the atomic structure of RSV N into their helical EM reconstruction (24).

A sequence alignment between RSV N and measles virus N (MeV N), both paramyxovirus nucleoproteins, is difficult to interpret because of the lack of amino acid identity. However, a comparison of the secondary structure elements observed in the RSV N structure, with a secondary structure prediction for MeV N (6) (Fig. 1), shows even more similarity than that between rhabdovirus and RSV N. This comparison also shows that the β-hairpin projecting from the distal end of the RSV N protomer (24) is conserved between these two paramyxovirus nucleoproteins. One important difference lies in the length of the highly disordered C-terminal domain, the N tail, that is 31 residues long (360 to 391) for RSV N (24) but 126 residues long (400 to 525) for MeV N (16). A short sequence in the MeV N tail (residues 489 to 506) folds into a dynamic helical structure that is stabilized by binding of the viral phosphoprotein that carries the viral RNA-dependent RNA polymerase

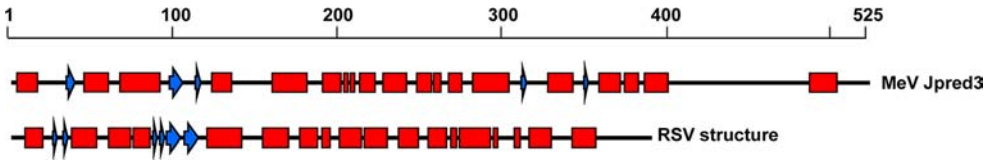


FIG. 1. Predicted secondary structure of MeV N compared to secondary structure elements in the atomic structure of RSV N. α-Helices are represented as red boxes, β-strands as blue arrows.

* Corresponding author. Mailing address: UVHCI, UMI 3265 UJF-EMBL-CNRS, BP 181, 38042 Grenoble, Cedex 9, France. Phone: 33476209463. Fax: 33476209400. E-mail: gutsche@embl.fr.

† Supplemental material for this article may be found at http://jvi.asm.org/.

▽ Published ahead of print on 24 November 2010.

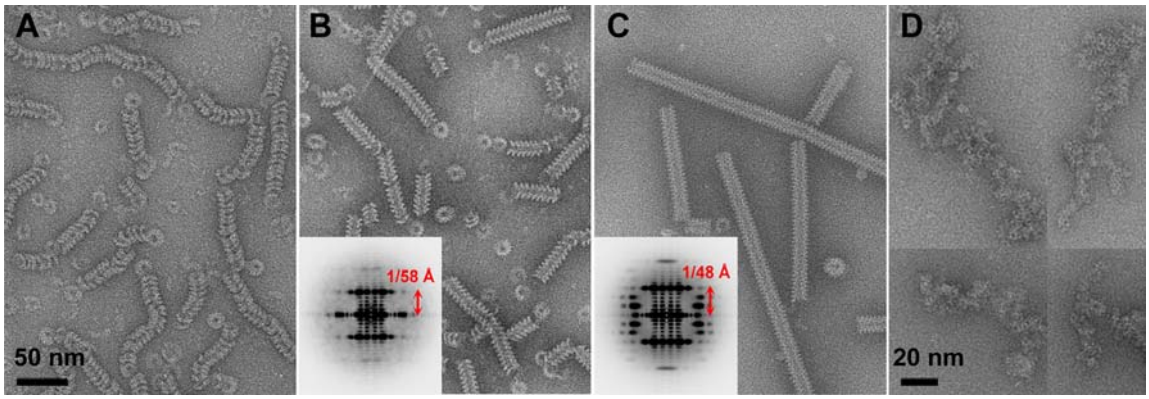


FIG. 2. Fields of view of negatively stained MeV nucleocapsids. (A) Intact nucleocapsids with 2% uranyl acetate and a single carbon layer. (B and C) Intact (B) or digested (C) nucleocapsids with NanoW in a double carbon layer and a representative class average of power spectra. (D) Recombinant C-terminally His-tagged nucleocapsids bound to anti-His-tagged antibody.

(12, 13, 16). The N tail is also involved in binding host proteins, such as hsp70 (5, 26) and interferon regulatory factor 3 (14, 15). So far, the location of the N tail on the helix is not known, although it is usually shown on the outside in cartoons that illustrate transcription and replication of paramyxoviruses (see Fig. 9 in reference 3). The helical model derived from the recombinant N-RNA ring structure of RSV, however, would place the N tail toward the helix interior, which would have consequences for the contacts between subsequent helical turns.

The helical structure of the intact measles virus N-RNA under cryoelectron microscopy (cryo-EM) conditions is highly flexible and difficult to determine by Fourier-Bessel image analysis or even by single-particle-based approaches (2, 21). However, once the N tail is removed by proteolysis, the structure becomes more regular and rigid and thus amenable to helical reconstruction by cryo-EM (21). Here, we show that the nondigested nucleocapsid structure can be addressed in negative-stain electron microscopy by trapping the sample between two layers of carbon film and by using NanoW stain (from Nanoprobes) instead of the more traditional uranyl acetate (Fig. 2). This preparation technique enables to image intact measles virus nucleocapsids as well as their trypsin-digested counterparts and has the advantage

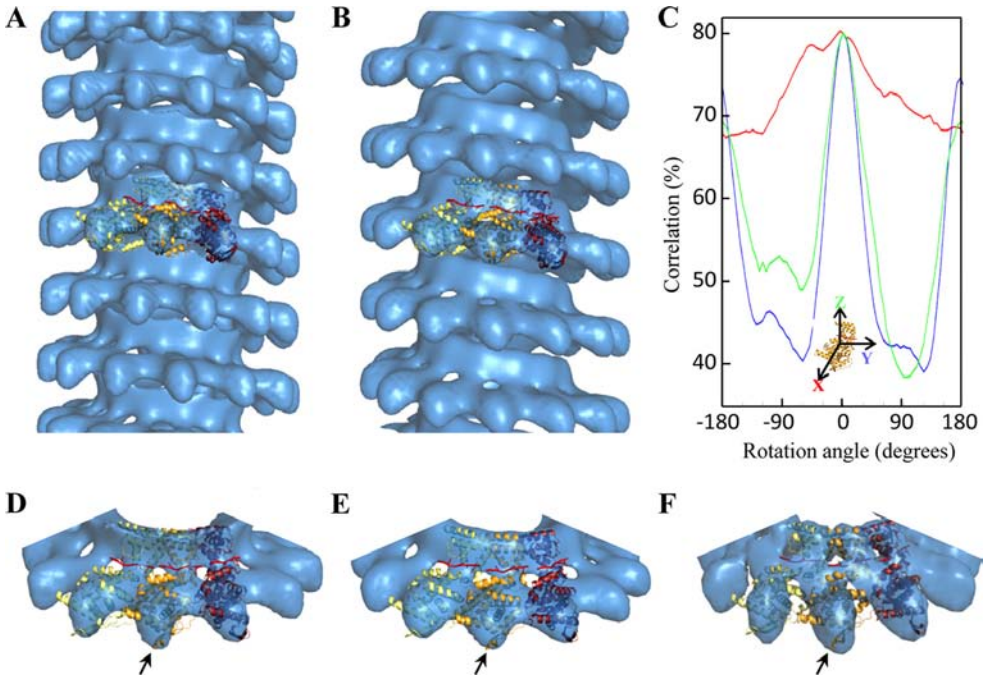


FIG. 3. Three-dimensional reconstructions of MeV nucleocapsids. (A and D) Digested nucleocapsid, (B and E) intact nucleocapsid, (F) cryo-EM reconstruction of digested MeV N (21). The fit of the RSV N-RNA atomic structure is shown as an overlay. The N-terminal β-hairpin fits nicely into the density (arrow). The RNA is shown as a red ribbon. (C) Docking precision for panel A. Shown is the correlation upon rotation of the monomer (see the supplemental material).

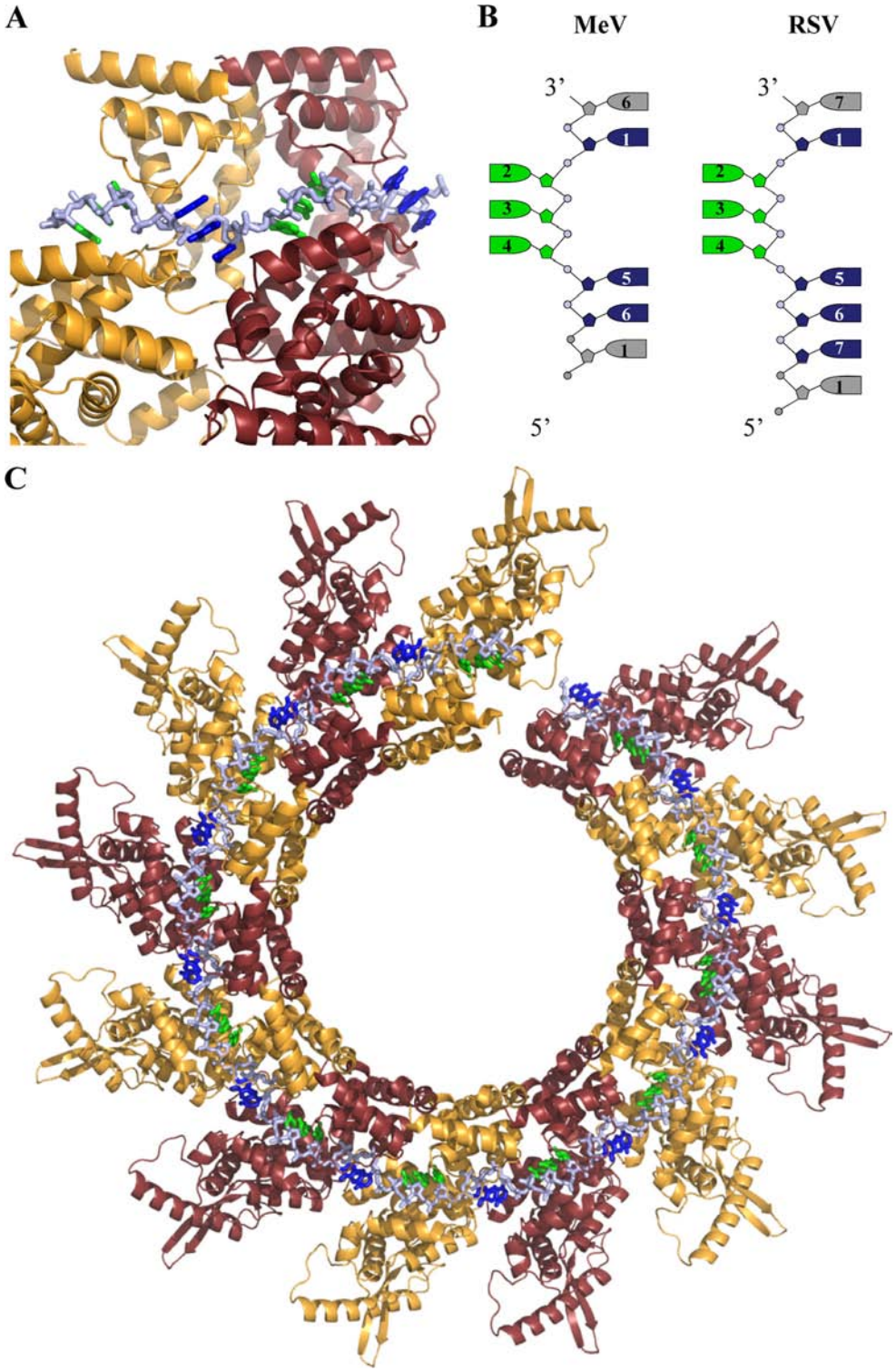


FIG. 4. RNA binding to MeV N based on RSV N-protomer fitting and energy minimization for solvent-exposed bases. Protein-oriented bases are in green, solvent-oriented bases are in blue, and the backbone is in light blue. (A) Enlarged view of RNA binding. (B) Schematic diagram for a comparison of RNA interaction with MeV N and RSV N. The numbering is as in reference 24. The gray nucleotides are on the neighboring N protomers. (C) Top view of the helical fit.

of maintaining the helix in a more rigid state. For this analysis, recombinant MeV N was produced, and a fraction of it was trypsinated as described previously (21) and imaged with a transmission electron microscope. Overlapping segments of the visually most rigid helices were selected with Boxer (17), contrast transfer function (CTF) corrected with CTFFIND3 (18) and Bsoft (11), and aligned and classified with Imagic (25). An additional classification of power

spectra of individual image frames and a sorting based on artificial smooth helical volumes improved the homogeneity of different subsets separated according to diameter and helical parameters. The major subsets were used for angular assignment and three-dimensional (3D) reconstruction in an iterative projection-matching procedure similar to that for IHRSR (7, 8) with the SPIDER package (9, 22), starting from a smooth helix of a chosen pitch as the initial model (for details, see the supplemental material).

Final three-dimensional reconstructions of trypsin-digested and intact measles nucleocapsids at a resolution of 25 Å are shown in Fig. 3. Removal of the N tail leads to a compaction of the helix, with the pitch shortening from 57.2 Å to 48.7 Å and a diameter constriction from 200 Å to 190 Å, in line with the previous cryo-negative-stain EM work (2). The number of subunits per turn in the digested nucleocapsid was found to be 12.33, the same as that previously obtained for such species under cryo-EM conditions by Fourier-Bessel analysis of the most regular helix coupled to IHRSR (21). Thus, in this case, the double-carbon layer negative-stain microscopy and the NanoW stain seem to maintain the helical structure without modifying the helical parameters. The intact nucleocapsid helix accommodates a nearly integer number of 12.92 subunits per turn, which agrees with the 5% increase in diameter. The overall shape of the nucleoprotein subunit is nevertheless very similar in both reconstructions, corroborating previous arguments in favor of the intrinsic N-tail disorder (2, 16).

Given the predicted structural similarity between RSV and MeV N, the atomic model of the RSV nucleoprotein monomer (Protein Data Bank [PDB]accession number 2WJ8) was used for fitting into the obtained 3D volumes with VEDA (http://mem.ibs.fr/VEDA), a new graphical version of URO (19) (Fig. 3). For fitting, MeV nucleoprotein helices were considered to be left-handed based on previously published metal shadowing results (21), and a modified PDB file of the RSV N protomer with only 5 nt corresponding to nt 2 to 6 was used, given that MeV N-RNA contains 6 nt per N protomer (4, 23) and not 7. Interestingly, without any constraints imposed during fitting, the fit ensures the continuity of RNA bound to measles virus N. Atomic coordinates of two RNA segments bound to consecutive subunits were extracted from the thus-obtained MeV nucleocapsid model, an additional ribonucleotide (corresponding to number 7 in Fig. 1D in reference 24) was inserted, and energy minimization was performed with VEGA software (20) to obtain a continuous, physically realistic RNA molecule. Since bases 2, 3, and 4 bind in a cavity on the RSV-N protein, their coordinates were kept fixed, while those of the solvent-facing ribonucleotides, 5, 6, and 1, were optimized. Figure 4 illustrates the possibility of easily constructing a 6-nt RNA with three bases facing the protein, as in the RSV N-RNA rings, and three bases stacked and pointing away from the protein into the solvent.

This fit of the atomic structure of RSV N into the negative-stain EM reconstructions is also consistent with the previously published cryo-EM structure of the digested MeV nucleocapsid (21) (Fig. 3F) and the RNA position predicted therein by *cis*-platinum RNA labeling. It suggests that the RNA is indeed localized at the exterior face of the helix, as in the RSV N-RNA rings, and not as in rhabdoviral N-RNA rings. Although the disordered C-terminal domain could not be resolved in the reconstruction of the intact nucleocapsid, the fit suggests that this crucial domain would point toward the helix interior. In addition, binding of anti-His-tagged antibody to C-terminally His-tagged nucleocapsids prevents correct helix formation (Fig. 2D) (see the supplemental material), indicating that the N tail domain may come out at a site where it interferes with contacts between two subsequent turns of the N-RNA helix, contributing to its flexibility.

We thank Francine Gérard and Euripedes Ribeiro de Almeida from the group of Marc Jamin (UVHCI) for help with initial production of MeV N and Guy Schoehn from UVHCI-IBS for discussions and advice.

A.D. was financed by a Ph.D. fellowship from the Rhône-Alpes region. This work was financed in part by the FINOVI Foundation.

REFERENCES

1. **Albertini, A. A., et al.** 2006. Crystal structure of the rabies virus nucleoprotein-RNA complex. Science **313:**360–363.
2. **Bhella, D., A. Ralph, and R. P. Yeo.** 2004. Conformational flexibility in recombinant measles virus nucleocapsids visualised by cryo-negative stain electron microscopy and real-space helical reconstruction. J. Mol. Biol. **340:** 319–331.
3. **Bourhis, J. M., B. Canard, and S. Longhi.** 2006. Structural disorder within the replicative complex of measles virus: functional implications. Virology **344:**94–110.
4. **Calain, P., and L. Roux.** 1993. The rule of six, a basic feature for efficient replication of Sendai virus defective interfering RNA. J. Virol. **67:**4822–4830.
5. **Couturier, M., et al.** 2010. High affinity binding between Hsp70 and the C-terminal domain of the measles virus nucleoprotein requires an Hsp40 co-chaperone. J. Mol. Recognit. **23:**301–315.
6. **Cuff, J. A., and G. J. Barton.** 2000. Application of multiple sequence alignment profiles to improve protein secondary structure prediction. Proteins **40:**502–511.
7. **Egelman, E. H.** 2007. The iterative helical real space reconstruction method: surmounting the problems posed by real polymers. J. Struct. Biol. **157:**83–94.
8. **Egelman, E. H.** 2000. A robust algorithm for the reconstruction of helical filaments using single-particle methods. Ultramicroscopy **85:**225–234.
9. **Frank, J., et al.** 1996. SPIDER and WEB: processing and visualization of images in 3D electron microscopy and related fields. J. Struct. Biol. **116:**190–199.
10. **Green, T. J., X. Zhang, G. W. Wertz, and M. Luo.** 2006. Structure of the vesicular stomatitis virus nucleoprotein-RNA complex. Science **313:**357–360.
11. **Heymann, J. B., G. Cardone, D. C. Winkler, and A. C. Steven.** 2008. Computational resources for cryo-electron tomography in Bsoft. J. Struct. Biol. **161:**232–242.
12. **Jensen, M. R., et al.** 2008. Quantitative conformational analysis of partially folded proteins from residual dipolar couplings: application to the molecular recognition element of Sendai virus nucleoprotein. J. Am. Chem. Soc. **130:** 8055–8061.
13. **Kingston, R. L., W. A. Baase, and L. S. Gay.** 2004. Characterization of nucleocapsid binding by the measles virus and mumps virus phosphoproteins. J. Virol. **78:**8630–8640.
14. **Laine, D., et al.** 2005. Measles virus nucleoprotein induces cell-proliferation arrest and apoptosis through NTAIL-NR and NCORE-FcgammaRIIB1 interactions, respectively. J. Gen. Virol. **86:**1771–1784.
15. **Laine, D., et al.** 2003. Measles virus (MV) nucleoprotein binds to a novel cell surface receptor distinct from FcγRII via its C-terminal domain: role in MV-induced immunosuppression. J. Virol. **77:**11332–11346.
16. **Longhi, S., et al.** 2003. The C-terminal domain of the measles virus nucleoprotein is intrinsically disordered and folds upon binding to the C-terminal moiety of the phosphoprotein. J. Biol. Chem. **278:**18638–18648.
17. **Ludtke, S. J., P. R. Baldwin, and W. Chiu.** 1999. EMAN: semiautomated software for high-resolution single-particle reconstructions. J. Struct. Biol. **128:**82–97.
18. **Mindell, J. A., and N. Grigorieff.** 2003. Accurate determination of local defocus and specimen tilt in electron microscopy. J. Struct. Biol. **142:**334–347.
19. **Navaza, J., J. Lepault, F. A. Rey, C. Alvarez-Rua, and J. Borge.** 2002. On the fitting of model electron densities into EM reconstructions: a reciprocal-space formulation. Acta Crystallogr. D Biol. Crystallogr. **58:**1820–1825.
20. **Pedretti, A., L. Villa, and G. Vistoli.** 2002. VEGA: a versatile program to convert, handle and visualize molecular structure on Windows-based PCs. J. Mol. Graph. Model. **21:**47–49.

21. **Schoehn, G., et al.** 2004. The 12 Å structure of trypsin-treated measles virus N-RNA. J. Mol. Biol. **339:**301–312.
22. **Shaikh, T. R., et al.** 2008. SPIDER image processing for single-particle reconstruction of biological macromolecules from electron micrographs. Nat. Protoc. **3:**1941–1974.
23. **Sidhu, M. S., et al.** 1995. Rescue of synthetic measles virus minireplicons: measles genomic termini direct efficient expression and propagation of a reporter gene. Virology **208:**800–807.
24. **Tawar, R. G., et al.** 2009. Crystal structure of a nucleocapsid-like nucleoprotein-RNA complex of respiratory syncytial virus. Science **326:**1279–1283.
25. **van Heel, M., G. Harauz, E. V. Orlova, R. Schmidt, and M. Schatz.** 1996. A new generation of the IMAGIC image processing system. J. Struct. Biol. **116:**17–24.
26. **Zhang, X., et al.** 2005. Hsp72 recognizes a P binding motif in the measles virus N protein C-terminus. Virology **337:**162–174.

Résumé

Aujourd'hui, la cristallographie de macromolécules produit couramment des modèles moléculaires à résolution atomique. Cependant, cette technique est particulièrement difficile à mettre en œuvre dans le cas de complexes de taille importante. La microscopie électronique permet, elle, de visualiser des particules de grande taille dans des conditions proches de celles *in vivo*. Cependant, la résolution des reconstructions tridimensionnelles obtenues exclut, en général, leur interprétation directe en termes de structures moléculaires, étape nécessaire à la compréhension des problèmes biologiques. Il est donc naturel d'essayer de combiner les informations fournies par ces deux techniques pour caractériser la structure des assemblages macromoléculaires. L'idée est de positionner les modèles moléculaires déterminés par cristallographie à l'intérieur de reconstructions 3D issues de la microscopie électronique, et de comparer la densité électronique associée à la reconstruction 3D avec une densité électronique calculée à partir des modèles. Le problème numérique réside dans la détermination et l'optimisation des variables qui spécifient les positions des modèles, considérés comme des corps rigides, à l'intérieur de l'assemblage. Cette idée simple a donné lieu au développement d'une méthode appelée recalage. Ce travail de thèse a eu pour but de fournir aux biologistes un outil, basé sur la méthode du recalage, qui leur permette de construire des modèles pseudo-moléculaires associés aux assemblages produits par microscopie électronique. Le logiciel issu de ce travail, nommé ∨∈⊃∧ est un environnement graphique convivial, intégrant la possibilité de recalage flexible, et un moteur de calcul performant (calcul rapide, traitement de symétries complexes, utilisation de grands volumes, ...). Testé sur des dizaines de cas réels, ∨∈⊃∧ est aujourd'hui pleinement fonctionnel et est utilisé par un nombre croissant de chercheurs, en France et à l'étranger, qui lui reconnaissent tous facilité d'utilisation, stabilité, rapidité et qualité des résultats.

Mots Clés : Recalage, Microscopie Électronique, Cristallographie, Assemblage Macromoléculaire, Modèle moléculaire, Environnement Graphique, Visualisation, Logiciel

Abstract

Macromolecular crystallography commonly produces molecular models at atomic resolution. However, this technique is particularly difficult to implement in the cases of large complexes. On the other hand, electron microscopy allows for the visualization of large particles under conditions similar to those *in vivo*. However, the resolution of three-dimensional reconstructions obtained by electron microscopy does not allow, in general, the direct interpretation of molecular structures, a necessary step in understanding biological problems. Therefore, it is natural to try to combine information provided by these two techniques to characterize the structure of macromolecular assemblies. The idea is to place the molecular models determined by crystallography inside electron microscopy 3D reconstructions, and compare the electron density associated to the 3D reconstruction with an electron density calculated from the models. The numerical problem is the determination and the optimization of the variables that specify the positions of the models, considered as rigid body, inside the assembly. This simple idea has led to the development of a method called fitting. This thesis aims to provide biologists with a tool, based on the method of fitting, that allows them to build pseudo-molecular models associated to the assemblies produced by electron microscopy. The software resulting from this work, named ∨∈⊃∧, has a user-friendly graphical environment, includes the possibility of flexible fitting, and implements a powerful calculation core (fast computation, treatment of complex symmetry, use of large volumes, etc.). Tested on dozens of real cases with experimental data sets, ∨∈⊃∧ is now fully functional and is used by a growing number of researchers in France and abroad. All of the users have recognized ease of use, stability, speed, and quality of results of ∨∈⊃∧.

Key words : Fitting, Docking, Electron Microscopy, Crystallography, Macromolecular Assembly, Molecular Model, Graphical Environment, Visualization, Software