N° d'ordre 075-2003 Année 2003

Thèse

présentée devant
l'Université Claude Bernard - Lyon 1
pour l'obtention
du Diplôme de Doctorat
(arrêté du 25 avril 2002)
présentée et soutenue publiquement
le 20 juin 2003

par Martin Jambon

Un système bioinformatique de recherche de similitudes fonctionnelles dans les structures 3D de protéines

Directeur de thèse : Dr Christophe Geourjon

Jury : Pr Alain-Jean Cozzone, Président
Pr Alexander Bockmayr, Rapporteur
Pr Joël Janin, Rapporteur
M François Delfaud
Dr Christophe Geourjon
Dr Serge Pérez

Résumé

SuMo est un système bioinformatique de comparaison de structures 3D de protéines. Le but de cette approche est de faciliter l'exploration des données structurales par les biologistes et la formulation d'hypothèses fines sur l'implication biologique des protéines.

Contrairement aux approches existantes dans ce domaine, SuMo ne s'attache pas à résoudre un problème formel dérivant d'une modélisation de la notion de fonction biologique. Ceci autorise des efforts constants de développement de l'heuristique mise en oeuvre, permettant de se rapprocher des notions intuitives de fonction biologique plutôt que de coller à un modèle mathématique peu réaliste.

L'heuristique développée est basée sur la représentation des macromolécules par des groupements chimiques aux propriétés géométriques hétérogènes et associés en triplets.

L'utilisation de SuMo s'effectue directement à partir du serveur web `http://sumo-pbil.ibcp.fr`. Le criblage de la banque de sites de fixation de ligands générée par et pour SuMo permet de repérer de façon conviviale de potentielles cibles thérapeutiques.

Abstract

SuMo is a bioinformatic system for comparing 3D structures of proteins. This approach was designed to help along the exploration of structural data by biologists and the formulation of accurate hypotheses concerning the biological implication of proteins.

As opposed to existing approaches in this field, SuMo does not solve a formal problem that would derive from a model for biological function. This allows constant development of the heuristics, by getting closer to intuitive concepts for biological function rather than stick to a mathematical model with poor relevance.

The heuristics that has been developed is based on a representation of macromolecules using chemical groups that are associated with heterogeneous geometrical properties and grouped into triplets.

SuMo can be used directly from the web server `http://sumo-pbil.ibcp.fr`. Screening SuMo's database of ligand binding sites allows in a convenient way to discover potential therapeutic targets.

Table des matières

Table des figures

Liste des tableaux

Liste des algorithmes

Chapitre 1

Introduction

Environ 20000 structures tridimensionnelles de macromolécules biologiques sont actuellement connues et stockées dans la Protein Data Bank (PDB) [8], base de données publique internationale. Comme l'indique son nom, cette base de données comporte une majorité de structures de protéines (environ 90 %), assez souvent en complexe avec des partenaires de nature variée.

Les principaux mécanismes biologiques qui conduisent à la fabrication des protéines par les cellules des différents êtres vivants sont aujourd'hui bien connus des biologistes. Les techniques de la biologie moléculaire, apparues et popularisées dans les années 1980, sont basées essentiellement sur la manipulation et le détournement de ces mécanismes. De ce fait, la notion d'acide aminé, monomère constitutif des protéines, est devenue centrale dans la façon de décrire les protéines en biologie moléculaire. Néanmoins, les protéines ne sont pas les seuls composés impliqués dans les mécanismes biologiques. Les protéines elles-mêmes se retrouvent fréquemment en association stable — covalente ou non — avec des composés de nature variée. Si la notion d'acide aminé est centrale dans les approches classiques de génétique, elle ne devrait pas avoir de statut privilégié lors de l'étude des interactions entre les molécules des êtres vivants. Le choix de modélisations appropriées aux problèmes que l'on traite a donc été central dans le développement du système qui va être décrit.

Les motivations qui ont poussé à mener à bien ce projet sont les suivantes :
- mettre en place un outil d'analyse et de comparaison des structures 3D de protéines permettant de mettre en évidence des similitudes et des différences critiques pour leur activité biologique ;
- que cet outil soit pratique à utiliser ;
- que les moyens financiers et matériels mis en jeu pour son utilisation soient raisonnables.

À partir de ces seules motivations, de nombreuses stratégies peuvent être envisagées. Il n'est pas nécessaire a priori de minimiser un paramètre défini mathématiquement comme un modèle énergétique, ni d'effectuer une superposition optimale de structures, ni encore de considérer comme une entité centrale les acides aminés formant un polypeptide.

La méthode retenue est *heuristique*, c'est-à-dire que les structures de données et les algorithmes développés ont été choisis sans preuve formelle de leur pertinence. Ce choix a été motivé par l'impossibilité de définir formellement ce qu'est une fonction biologique. Des problèmes voisins tels que *Ce ligand peut-il former un complexe stable avec ma protéine de structure 3D connue ?* ou *Ce site est-il rare ?* peuvent être formulés de façon exacte, mais ne sont pas l'objet du système développé. Ce système de détection de similitudes fonctionnelles à partir des structures 3D de protéines a été baptisé *SuMo*[1].

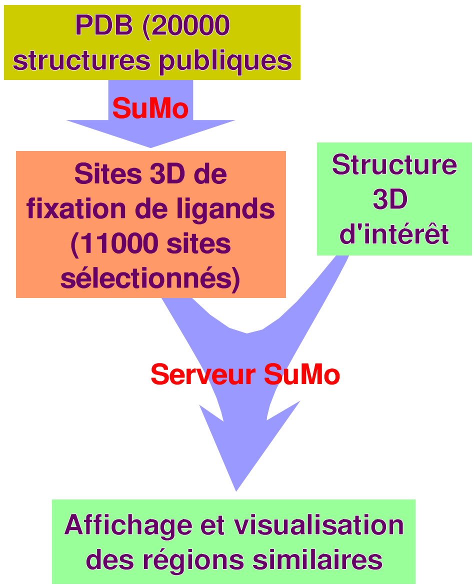


FIG. 1.1 – Recherche de sites de fixation de ligands dans une structure d'intérêt par SuMo

Le projet SuMo, initié en janvier 2000, a pour vocation d'extraire des similitudes fonctionnelles à partir de structures 3D de complexes macromoléculaires stables. Il se veut le plus générique possible, même si sa conception actuelle est centrée sur le modèle protéine rigide/ligand flexible. A ce jour, SuMo permet notamment de cribler une base de données de 11000 sites 3D de fixation de ligands contre une structure 3D d'intérêt. Les résultats four-

[1]signifiait initialement *Surfing the Molecules*

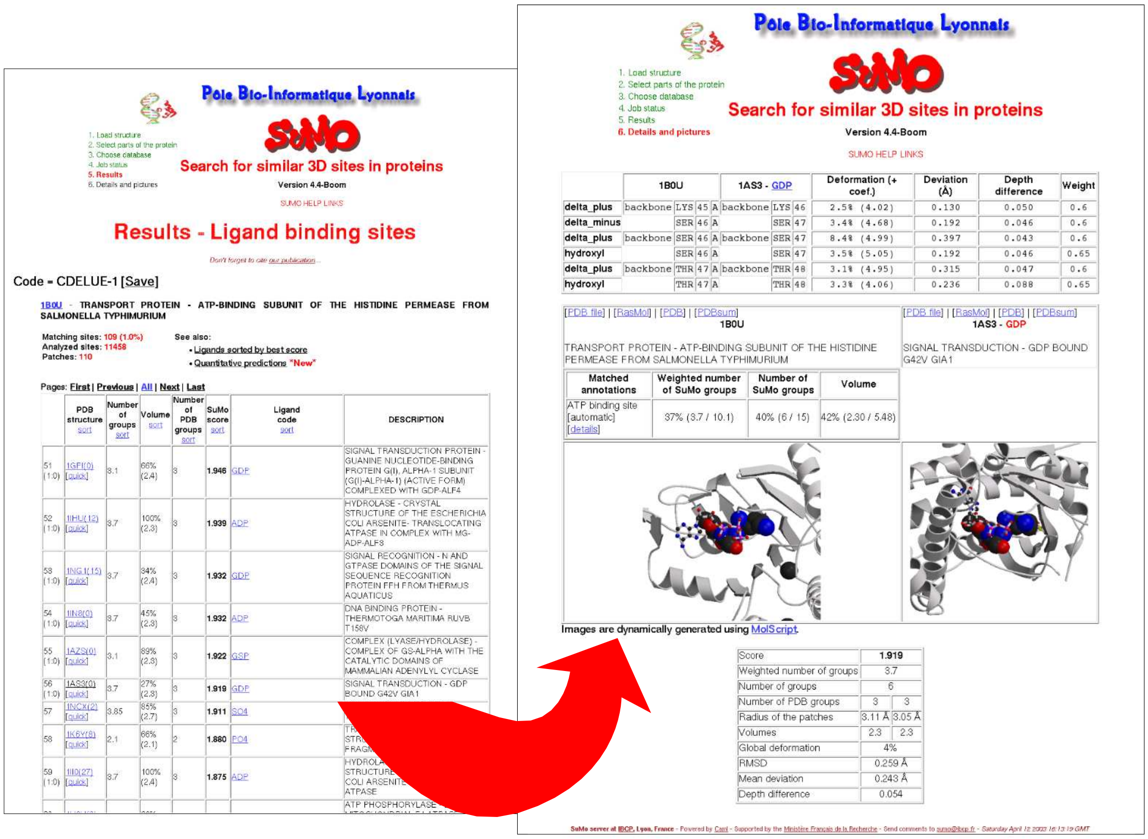


Fig. 1.2 – Présentation de résultats obtenus par SuMo. Exemple du criblage de la base de données de sites de fixation de ligands par la structure d'une protéine fixatrice d'ATP.

nis proposent des sites potentiels de fixation de ligands ou de fragments de ligands. Cette démarche prédictive est schématisée au niveau de la figure 1.1 page 20 et un aperçu de la présentation des résultats est présenté figure 1.2. Elle s'inscrit dans le cadre de l'identification de cibles thérapeutiques potentielles. D'une manière générale elle doit favoriser la compréhension d'éventuels rôles biologiques de protéines mal connues, comme les protéines issues de la *génomique structurale*.

Avant de commencer la lecture de ce rapport, il est essentiel d'essayer le système SuMo, a partir de l'adresse suivante :

```
http://sumo-pbil.ibcp.fr
```

Pour utiliser SuMo, il suffit d'être connecté à Internet et de connaître une structure 3D de protéine. Le parcours rapide des fonctionnalités proposées au niveau du serveur web est considéré comme une partie intégrante des illustrations de ce rapport de thèse.

Ce rapport décrit les fondements conceptuels et techniques du système SuMo. L'utilisation de SuMo dans le cadre de problématiques biologiques particulières n'est pas l'objet de notre propos.

La version décrite est SuMo 4.4, première version publique officielle, datant du mois de mars 2003. SuMo n'est pas une méthodologie figée. Ainsi, des modifications à tous les niveaux sont susceptibles d'intervenir dans les prochaines versions.

Chapitre 2

Contexte scientifique et méthodologique

2.1 Bioinformatique moléculaire et cellulaire

La figure 2.1 page 24 présente un schéma très simple des principaux types de données couramment utilisées en biologie moléculaire et cellulaire et facilement modélisables et stockables dans des bases des données informatiques. Les différentes étapes illustrées sont centrées sur les protéines : l'information génétique stockée au niveau des acides nucléiques (ADN, ARN) peut être stockée sous forme de séquences, de la même façon que pour les protéines. Cette modélisation très pratique et provenant directement des résultats expérimentaux de séquençage n'est pas actuellement suffisante pour comprendre les mécanismes biologiques, que ce soit dans le cas des protéines ou dans celui du matériel génétique.

Les outils de bioinformatique actuels ne permettent pas de prédire une cascade biochimique à partir du génome d'un organisme. Ceci est lié en particulier au caractère auto-contrôlé des êtres vivants et à la non-connaissance des conditions initiales nécessaires pour amorcer la simulation d'un être vivant.

Actuellement, la bioinformatique permet de proposer des moyens d'exploiter des données expérimentales de grande taille ou complexes selon des approches prédéfinies et reproductibles. Ces approches permettent de fournir des prédictions et des hypothèses permettant d'éclairer le biologiste dans ses recherches.

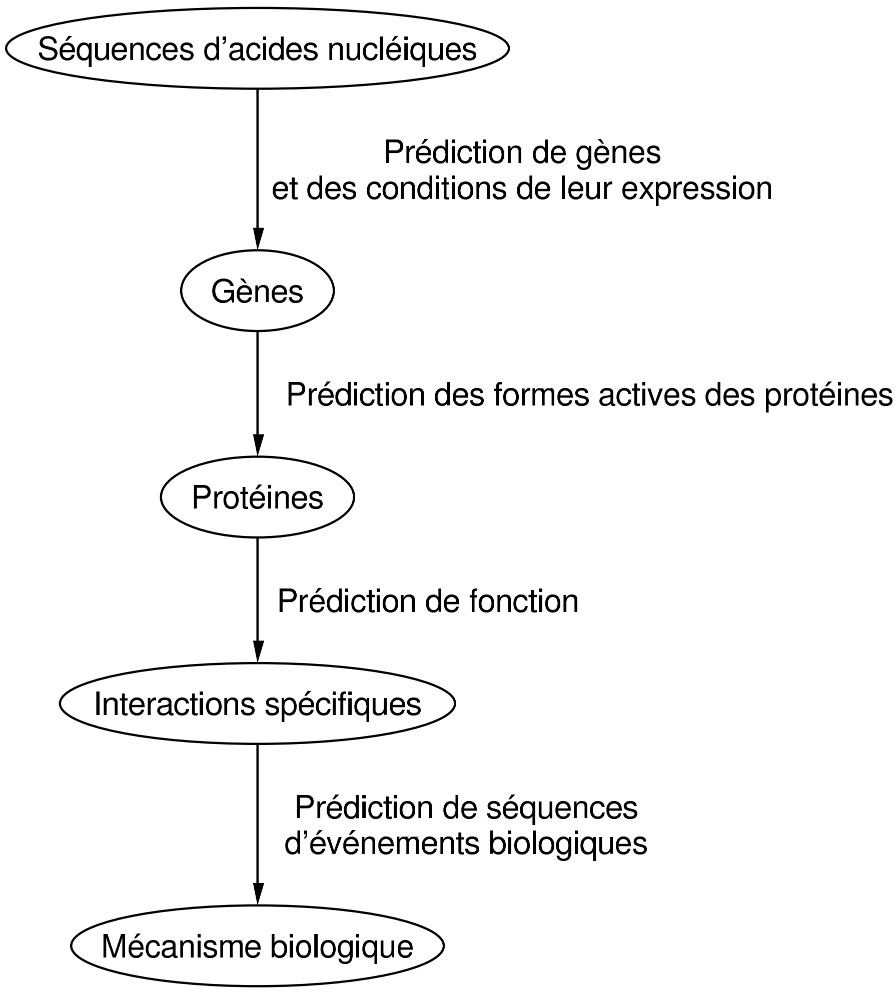


FIG. 2.1 – Données biologiques couramment exploitées en bioinformatique moléculaire et cellulaire

2.2 Relation structure-fonction dans les protéines

2.2.1 Généralités

2.2.1.1 Notion de site

Les protéines sont des hétéropolymères linéaires naturels, présents dans tous les êtres vivants connus et dont les monomères sont généralement les 20 acides aminés classiques. Dans de nombreux cas, les protéines peuvent subir des modifications covalentes, par greffage d'autres molécules, clivage ou pontages internes par des ponts disulfures entre cystéines. Beaucoup de protéines adoptent, dans des conditions données, une conformation stable, c'est-à-dire un pelotonnement particulier dont les fluctuations sont faibles devant la taille globale de la protéine. Nous appellerons structure 3D la conformation stable d'une protéine observée expérimentalement, où les coordonnées des atomes la constituant sont connues avec une incertitude plus faible que la distance entre ces atomes. Bien entendu, toutes les régions d'une protéine donnée n'ont pas la même stabilité, en particulier les chaînes d'atomes en contact avec le milieu extérieur (solvant, ligands) et ayant peu de contraintes stériques ont plus de chances d'être mobiles. Les principales techniques permettant de connaître la structure 3D des protéines sont actuellement la cristallographie aux rayons X et la résonance magnétique nucléaire (RMN).

La plupart des protéines ont une influence sur la vie de l'organisme qui les produit. Nous appellerons *fonction d'une protéine* la description de cette influence. Les fonctions des protéines peuvent être variées, et leur description n'a rien de formel. Cependant, de nombreuses protéines interagissent avec des partenaires spécifiques. Ces interactions n'impliquent en général qu'une partie de la protéine. De telles régions sont appelées *sites fonctionnels* et plus précisément sites d'interaction lorsqu'ils correspondent uniquement à la zone de contact entre les 2 partenaires. L'étude de familles de protéines ayant des séquences similaires et des fonctions semblables permet de révéler les acides aminés ou les groupements chimiques particulièrement conservés au cours de l'évolution. Ces régions peuvent être considérées comme impliquées dans des sites fonctionnels, que ce soit dans des sites d'interaction directe ou dans des interactions permettant de stabiliser la conformation de la protéine.

2.2.1.2 Compétition pour les interactions

Les protéines des êtres vivants sont localisées dans des milieux denses (liquides, membranes lipidiques, cristaux, ...). Elles sont donc en interaction permanente avec d'autres molécules, et peuvent former des complexes de durée de vie hautement variable. Si une protéine fixe de façon assez stable deux molécules différentes en mettant en jeu des sites chevauchants, alors il y aura *compétition*. Une molécule qui interagit de façon plus stable que la moyenne avec certaines protéines est appelée *ligand* de ces protéines. Une définition plus précise de cette notion sera donnée section 3.2.1.1 page 49. Certains ligands induisent des modifications de conformation des protéines sur lesquelles ils se fixent, pouvant entraîner un changement d'affinité de la même protéine pour un ligand donné au niveau d'un autre site.

Ces remarques nous conduisent à la constatation que la composition du milieu dans lequel se trouve une protéine va déterminer la probabilité que cette protéine soit fixée à un ligand donné à un instant t donné. Il est particulièrement important de prendre ceci en compte lors de la conception de médicaments censés jouer un rôle d'inhibiteurs compétitifs au niveau d'un site donné dans un certain contexte biologique.

2.2.2 Recherche de sites fonctionnels par comparaison

Il est souvent intéressant de connaître les sites fonctionnels d'une protéine donnée. Pour cela, toutes les approches sont autorisées. Nous pourrons distinguer les approches expérimentales, c'est-à-dire celles mettant en jeu la manipulation physique de la protéine étudiée et les approches informatiques.

Les approches informatiques consistent à transformer des données expérimentales par un système plus ou moins automatisé de façon à faire apparaître des particularités intéressantes dans les données étudiées, particularités qui ne sont pas triviales si l'on se contente de regarder les données expérimentales brutes. Ce sont ces approches qui vont nous intéresser ici, bien qu'elles soient complémentaires et indissociables des approches expérimentales.

Les protéines peuvent être modélisées de façon simple à partir de certaines données expérimentales :

- la séquence en acides aminés de la protéine étudiée,
- la structure 3D de la protéine sous la forme des coordonnées de la totalité ou d'une partie des atomes la constituant.

Les séquences des formes actives des protéines que l'on étudie peuvent être obtenues selon différentes méthodes avec plus ou moins de pertinence. Actuellement, la base de données publique généraliste de séquences de protéines non-redondantes Swiss-Prot [10] contient environ 125000 séquences de protéines. Cette base de données est annotée manuellement en utilisant un maximum d'informations expérimentales et des références vers d'autres sources de documentation. D'autres bases de données comprennent un plus grand nombre de séquences, comme PIR-NREF [5] qui regroupe environ 1200000 séquences non redondantes. Les structures 3D de protéines, stockées dans la base de données publique PDB sont au nombre de 19000 environ, ceci incluant différentes structures de protéines identiques.

Différents types d'approches bioinformatiques permettent, à partir des séquences ou des structures, de prédire les fonctions et de localiser les sites fonctionnels dans les protéines avec une qualité variable. À ce niveau, nous pouvons distinguer deux types de stratégies :

- les stratégies basées sur la comparaison avec des sites fonctionnels avérés,
- les stratégies *ab initio*, c'est-à-dire qui n'utilisent pas des informations expérimentales du même type que celles que l'on souhaite prédire comme conditions initiales ou contraintes du système.

Les stratégies *ab initio* ne seront pas examinées ici. De telles approches auraient l'avantage de faire apparaître des fonctions biologiques entièrement nouvelles. À ce titre, nous pouvons citer les simulations de *docking* pur, sans connaissance préalable de régions potentielles de fixation du ligand considéré.

La figure 2.2 page 28 présente les différents types d'approches bioinformatiques pouvant être utilisées pour mettre en évidence des sites conservés dans les protéines. Ces approches sont classifiées selon que ce sont des techniques d'alignement global de protéines ou bien des techniques d'identification de sites similaires, la notion de site étant ici très large.

2.2.2.1 À partir de la séquence en acides aminés

La séquence en acides aminés est une information moins riche que la structure 3D, au moins parce qu'elle peut être déduite de la structure 3D de façon triviale. Cette information est cependant plus facile à obtenir expérimentalement.

Alignement de séquences homologues Lorsque l'on dispose d'un ensemble de protéines de séquences connues et de fonction identique, par exemple provenant de différents organismes, on peut analyser quels sont les acides aminés qui ont été le plus conservés au cours de l'évolution. On pourra alors considérer que ceux qui sont les plus conservés ont été soumis à une pression de sélection parce que leur présence est importante. Pour cela, on peut effectuer des alignements multiples de séquences homologues en utilisant des outils comme ClustalW [47] ou Multalin [14].

Recherche de motifs Un motif, lorsque l'on travaille sur les séquences de protéines, peut être une expression régulière. Celle-ci peut être obtenue de diverses façons, éventuellement en s'aidant d'alignement multiple de séquences

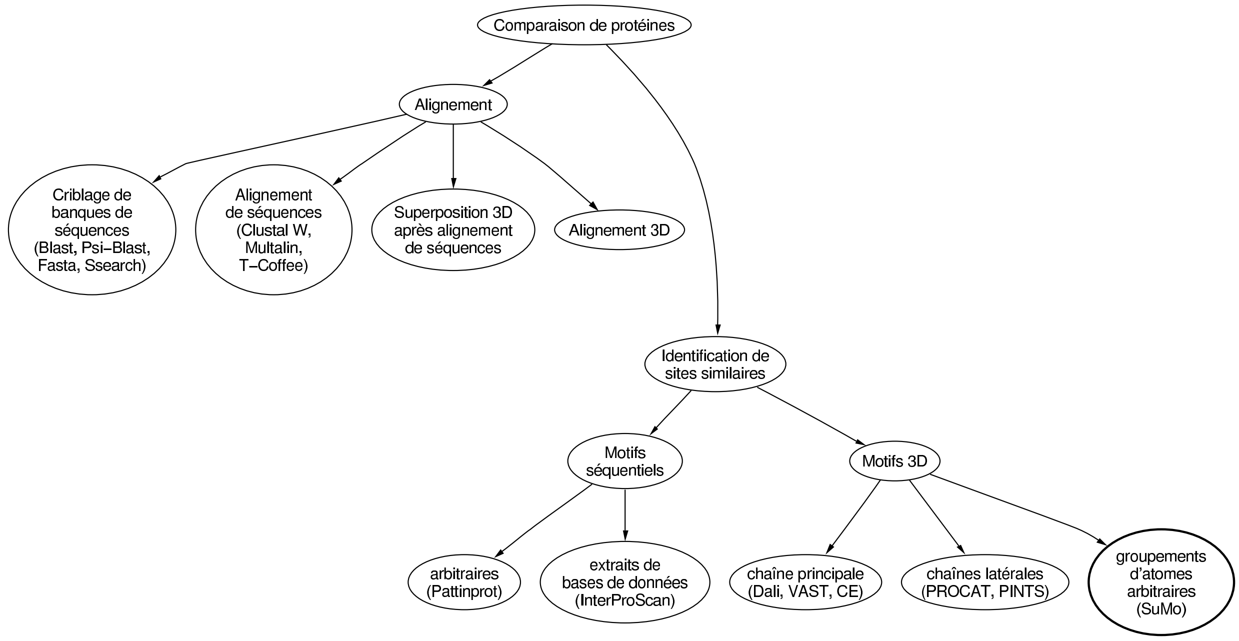


Fig. 2.2 – Stratégies bioinformatiques de recherche de sites fonctionnels dans les protéines par comparaison et quelques exemples d'outils associés

de protéines homologues. La base de données publique Prosite [4] contient plus de 1800 motifs de ce type, dont la plupart sont accompagnés d'une documentation détaillée. Une recherche dans cette base de données peut être effectuée à l'aide de l'outil Proscan.

Cependant, la représentation d'un motif par une simple expression régulière sur des caractères n'est pas forcément idéale. Des informations supplémentaires telles que la structure secondaire réelle ou prédite peut être ajoutée, ainsi que d'autres informations connues expérimentalement ou prédictibles. Une telle alternative est proposée par l'analyse des profils [19].

2.2.2.2 À partir de la structure 3D

Plusieurs outils bioinformatiques permettent d'aligner des structures de protéines ou d'identifier des domaines structuraux conservés au niveau du repliement de la chaîne principale. Nous pouvons par exemple citer Dali [20, 21] ou CE [42]. Ces approches permettent en particulier d'identifier des homologies difficiles à détecter par simple alignement de séquences. Ces approches ne permettent néanmoins pas de prendre en compte le positionnement des atomes autres que ceux de la chaîne principale. Nous nous intéresserons donc aux approches plus fines de recherche de sites fonctionnels généralement non basés sur des considérations évolutives.

Les protéines adoptant des structures relativement stables, elles peuvent sous certaines conditions être modélisées comme des objets géométriques, dont les fluctuations locales sont faibles devant les distances analysées. Sur ces objets peuvent être projetées des propriétés diverses, qui peuvent permettre de discriminer les sites sur des critères qui ne soient pas purement géométriques.

Deux classes d'approches peuvent être distinguées parmi les méthodes de recherche de sites fonctionnels par comparaison de structures 3D. Cette distinction est faite selon la modélisation primordiale de la protéine, c'est-à-dire celle qui est prise en compte en premier lieu au moment de la comparaison. Ces deux types de représentations sont les suivantes :

- modélisation de la surface des protéines,
- représentation de la protéine par des points épars associés à des propriétés indépendantes d'un point à l'autre.

Nous allons donc faire un bref tour d'horizon des fondements des différentes méthodes existantes, en faisant ressortir leurs principaux atouts et leurs limites.

Modélisation de la surface des protéines Les protéines peuvent être vues comme des solides, dont la forme de la surface va définir des cavités ou des extrusions permettant de loger des molécules ou parties de molécules. Cette modélisation correspond au modèle de la clé et de la serrure. Sur une surface représentant une protéine, il est possible de projeter différentes propriétés physico-chimiques. Une revue de ces méthodes a été réalisée en 2000 par Via *et al.* [48].

Des approches basées sur des heuristiques provenant du domaine de la vision artificielle (*machine vision*) ont été développées [18]. Ce type d'approche nécessite une représentation abstraite de la surface des protéines puis une discrétisation, c'est-à-dire sa représentation par un ensemble fini de points répartis le long de cette surface. Malheureusement, le nombre de points générés pour avoir une représentation assez fidèle de la surface théorique est beaucoup plus élevé que le nombre d'atomes présents en surface de la molécule. Ceci entraîne des calculs assez importants. Dans le but de les réduire, une sélection de points critiques peut être effectuée parmi les points représentant la surface [30]. Dans des approches de docking, cette approche a été étendue de façon à introduire un nombre limité de charnières dans les molécules [41].

D'autres heuristiques également basées sur une représentation de la surface des protéines existent [38, 39], basées sur une représentation rigide de la surface.

La limite majeure de toutes les approches utilisant une représentation de la surface des protéines est le temps de calcul. En effet, l'information initiale permettant de calculer une surface est un ensemble d'atomes, c'est-à-dire un nombre d'éléments ponctuels de position connue expérimentalement. Cependant, les heuristiques utilisées demandent une discrétisation de la surface, ce qui résulte en un éparpillement de l'information initiale.

D'autre part, le choix de la façon dont est générée une surface est libre. Elle peut être obtenue par la surface parcourue par une bille roulant virtuellement sur les atomes de la molécule. Il faut néanmoins choisir le rayon des atomes et celui de la bille de façon judicieuse. Or, plusieurs échelles peuvent être considérées, selon la diversité des propriétés fonctionnelles de la protéine, imposant plusieurs définitions différentes de la surface. De plus, certains acides aminés essentiels pour la fonction des protéines sont enfouis à l'intérieur de la protéine.

Modélisation par des éléments ponctuels Les protéines peuvent être découpées directement en un ensemble d'éléments modélisés par des objets ponctuels. Un certain nombre d'approches ont été basées sur le paradigme *un acide aminé = une sous-unité fonctionnelle*. À ce titre, nous pouvons citer [1] une approche consistant à rechercher des sous-graphes isomorphes dans des graphes où les sommets sont constitués par la position des carbones α et celle des chaînes latérales.

Par opposition, d'autres approches, celles de Wallace *et al.* [51, 50] et de Russell [40] considèrent qu'attribuer une position par chaîne latérale n'est pas suffisant. Ces approches utilisent directement la position de certains atomes. Dans les deux cas, la comparaison est basée sur des critères de distances entre certains atomes.

Chez Wallace, un atome défini spécifiquement au sein de chaque type de chaîne latérale ainsi que son environnement permettent de constituer une clé de hâchage permettant une recherche rapide de sites similaires au sein d'une base de données. Cette méthode est proposée via le serveur PROCAT [52] sur une base de données de sites enzymatiques.

L'approche de Russell est basée sur une équivalence entre certains atomes de certaines chaînes latérales. La comparaison est basée sur des distances interatomiques. Ici, nous avons la notion de groupement d'atomes que nous retrouverons dans SuMo. Cependant, il est fortement lié à la notion de chaîne latérale d'acide aminé et à la notion d'atome puisque les comparaisons s'effectuent par comparaison de distances interatomiques. Cette méthodologie est implémentée et utilisable au niveau du serveur PINTS [44] disponible depuis début 2003. Cette implémentation est associée à une fonction de score [44] dont le but est de donner une signification statistique aux résultats de comparaison. Cette statistique a pour but de donner une notion de rareté théorique à un site détecté avec un RMSD donné. Rappelons la définition du RMSD :

Définition 1 (RMSD) *Soit E et F des sous-ensembles de $\mathbf{R}^n$. La fonction* rmsd *est définie ainsi, pour tout ensemble $L \subset E \times F$:*

$$\mathrm{rmsd}(L) = \sqrt{\frac{1}{|L|} \sum_{(p,q) \in L} \|p - q\|^2}$$

où $|L|$ désigne le cardinal de l'ensemble L. Soit Φ l'ensemble des transformations rigides de $\mathbf{R}^n$ dans $\mathbf{R}^n$, c'est-à-dire les transformations qui conservent les distances et les angles orientés. La fonction RMSD *est définie ainsi :*

$$\begin{aligned}&\mathrm{RMSD}(\{(p_1, q_1), (p_2, q_2), \ldots\}) = \\ &\quad \min \{\mathrm{rmsd}(\{(p_1, \phi(q_1)), (p_2, \phi(q_2)), \ldots\}) | \phi \in \Phi \ \}\end{aligned}$$

Dans la suite, nous appellerons liste de correspondances un ensemble de paires de points équivalents. Une définition plus précise est donnée page 89.

L'idée du RMSD est de comparer des objets globalement superposables. Il n'y a pas de raison particulière de penser que le RMSD est le meilleur critère pour exprimer la similitude fonctionnelle de deux sites. En effet, de nombreux ligands des protéines sont flexibles et pourront se fixer sur des sites ayant des formes différentes mais en utilisant éventuellement le même type d'interactions.

Nous verrons que l'approche SuMo favorise l'obtention de résultats satisfaisants pour l'utilisateur d'une manière générale plutôt que le développement rigoureux d'estimateurs statistiques peu réalistes face à l'immense diversité des sites fonctionnels biologiques.

2.3 Outils de programmation

Le choix d'outils de programmation adaptés est crucial en bioinformatique, comme dans toutes les branches de l'informatique appliquée. En effet, un temps important est passé pour implémenter les algorithmes proposés et il est donc important de le réduire au maximum.

Il n'est pas question ici de faire l'apologie d'un langage de programmation donné. Il n'y a pas de preuve qu'un langage est meilleur qu'un autre pour une tâche donnée. Donnons simplement ici quelques grandes lignes qui peuvent aider à s'interroger sur le choix de tel ou tel langage :

- plaisir pris par le programmeur
- temps passé à programmer
- facilité d'écriture du code
- facilité d'extension de code déjà existant
- longueur des programmes
- qualité des compilateurs
- portabilité des compilateurs et des bibliothèques
- viabilité du langage
- disponibilité de bibliothèques utiles
- réutilisabilité du code
- facilité d'apprentissage
- qualité de la documentation disponible sur le langage
- possibilité de spécialisation de la syntaxe
- interfaçage avec d'autres langages
- degré de spécialisation du langage

Le langage de programmation qui a été choisi est *Objective Caml* [29], développé à l'INRIA-Rocquencourt.

2.3.1 Langage de programmation généraliste

Objective Caml est un langage de programmation généraliste fortement typé. Il permet d'utiliser conjointement les styles de programmation fonctionnel, impératif ou objet. La gestion de la mémoire est entièrement automatique. Le typage est entièrement statique, permettant ainsi une bonne efficacité du code compilé tout en permettant la détection des erreurs de programmation dès la compilation. La performance du code généré atteint souvent celle de code C équivalent [3] grâce au compilateur de code natif. Un compilateur de code pour machine virtuelle est également proposé. Les compilateurs et la bibliothèque standard sont disponibles sur les plate-formes les plus courantes que sont Windows, MacOS et toutes les déclinaisons de systèmes Unix.

Les utilisateurs avertis considèrent généralement que l'utilisation d'Objective Caml par rapport à C ou C++ permet un gain de productivité d'un facteur compris entre 5 et 10 pour le développement professionnel de logiciels.

2.3.2 Syntaxes spécialisées

La distribution d'Objective Caml fournit également des outils permettant de manipuler des syntaxes spécialisées.

Le couple *Ocamllex*/*Ocamlyacc* est l'équivalent de Lex/Yacc développés initialement pour le C. Ces outils permettent de facilement manipuler des langages définis par le programmeur. Ces outils ont été utilisés pour tous les langages définis pour le système SuMo.

Camlp4 [15] est un système qui permet d'étendre la syntaxe de Caml, de façon à simplifier l'écriture de code dans des situations particulières. Il permet également de définir une syntaxe à partir de zéro et propose une syntaxe révisée de Caml, et la possibilité de convertir les deux syntaxes de façon automatique. Seules des extensions syntaxiques ont été définies ou utilisées dans SuMo.

2.3.3 Stockage de données

SuMo ne nécessite pas l'utilisation d'une base de données relationnelle pour stocker les données représentant les structures 3D préparées pour la comparaison. Par contre, les données manipulées étant relativement complexes et cycliques, leur stockage a été énormément facilité par l'utilisation du module *Marshal*. Ce module est inclus dans la bibliothèque standard d'Objective Caml. Il fournit des fonctions polymorphes permettant de convertir des données de presque tous les types. Actuellement, seuls les objets ne peuvent pas être convertis dans ce format, et les fonctions ou les fermetures (*closures*) ne peuvent être relues que par le programme qui a généré les données.

Caml étant typé statiquement, il faut néanmoins que le programme qui relit les données au format Marshal connaisse leur type sans quoi cela ne sera pas détecté et entraînera généralement un plantage du programme. Ce module est une des rares façons de générer un *Segmentation fault* à partir d'un programme Caml pur. Pour cette raison, les données converties depuis le format Marshal doivent être sûres. Pour permettre à ce système d'être utilisé pour la sauvegarde de données chez des utilisateurs distants, un système de signature cryptographique a été mis en place. Ce système est décrit page 116.

2.3.4 Communication

L'interface utilisateur de SuMo repose sur un système client-serveur : les calculs essentiels sont effectués à partir d'un serveur HTTP, et le client voit s'afficher des pages web (format HTML). Le mécanisme d'exécution de programmes au niveau d'un serveur HTTP depuis un client HTTP est appelé CGI.

Si une telle interface est plus lourde à implémenter et un peu moins conviviale qu'une interface graphique conçue pour un logiciel local, elle présente néanmoins les avantages suivants :

- l'utilisateur n'a besoin d'installer aucun logiciel spécifique sur son poste de travail,
- l'utilisateur n'a pas à installer une base de données sur son poste,
- il n'y a pas besoin de distribuer le logiciel,
- le logiciel n'a pas besoin d'être extrêmement portable,
- les utilisateurs n'ont pas à se préoccuper des mises à jour et tout le monde utilise la même version.

Chapitre 3

Description du système SuMo

Le système SuMo associe une méthode et son implémentation. Les deux aspects ont été développés conjointement. Les efforts de développement se sont portés sur les aspects suivants :

- fournir un outil d'investigation des structures 3D de protéines,
- développer un outil utile pour tout biologiste s'intéressant à la fonction des macromolécules,
- réaliser l'ensemble au cours de la durée de la thèse,
- que le système soit le plus généraliste possible,
- que le logiciel soit suffisamment performant pour supporter plusieurs requêtes de calcul par jour,
- que son utilisation soit aisée,
- que son utilisation soit néanmoins souple.

Dans un premier temps va être présentée l'architecture du système développé, selon différents points de vue. La description du coeur de la méthode est ensuite détaillée, sans néanmoins détailler certains modèles, heuristiques et algorithmes complexes et indépendants du concept de macromolécule. Ces heuristiques spécialisées font l'objet d'une section à part entière. Est ensuite décrite la conception des interfaces utilisateurs, puis la répartition des tâches sur une ou plusieurs machines. Enfin, une *FAQ*[1] regroupe quelques unes des questions les plus souvent posées au sujet du système SuMo et tente d'y répondre simplement en renvoyant si nécessaire le lecteur vers les sections appropriées de ce document.

[1] Frequently Asked Questions ou Foire Aux Questions

3.1 Architecture générale

3.1.1 Les différents niveaux

SuMo est structuré en trois couches : la couche de plus haut niveau est celle de l'utilisateur normal, à travers l'interface web. La couche la plus basse est celle du langage de programmation et devrait être réservée aux développeurs. La couche intermédiaire permet au développeur d'effectuer des tests et à un utilisateur avancé d'effectuer des opérations spécifiques, non proposées par défaut au niveau de l'interface web. Les caractéristiques essentielles de ces trois niveaux sont résumées dans le tableau 3.1.

Niveau	Échelle	Type d'utilisateurs	Langage
Inférieur	Machine/Système local	Programmeur	Caml
Intermédiaire	Système de fichiers	Prog. d'interfaces	SuMo
Supérieur	Mondiale	Utilisateur	Aucun/SuMoQ

Tab. 3.1 – L'organisation de SuMo en 3 niveaux

3.1.1.1 Niveau inférieur : langage de programmation

Le niveau le plus bas d'utilisation correspond à la couche langage de programmation. L'utilisateur qui modifie SuMo à ce niveau-là est un développeur.

Le système SuMo comporte plusieurs exécutables indépendants. Il est implémenté quasi-exclusivement en Objective Caml. Les fichiers-sources pour Caml ou ses préprocesseurs (Camlp4, Ocamllex et Ocamllyacc) sont au nombre de 285, pour un total de 30000 lignes dans la version 4.4. Le programme qui implémente la méthode depuis la lecture des données structurales jusqu'à la génération de fichiers-résultats est nommé `sumo`. À lui-seul, il comporte 181 fichiers-sources correspondant à 157 modules et 20000 lignes de code. La figure 3.1 page 37 présente le graphe de dépendances entre les modules de `sumo` et permet d'avoir une idée de la complexité de l'heuristique développée.

FIG. 3.1 – Aperçu de la complexité de `sumo` : graphe de dépendances entre les 157 modules du programme `sumo` dans sa version 4.4. Les sous-modules ne sont pas pris en compte. $A \rightarrow B$ indique que le module A fait référence au module B. Ce graphe est généré grâce à Ocamldot et mis en forme par Graphviz.

3.1.1.2 Niveau intermédiaire : langage SuMo

Le niveau intermédiaire correspond à l'utilisation directe du programme `sumo`. Depuis la version 2 (septembre 2000), le programme s'utilise grâce à un langage spécifique, en raison du nombre important de tâches et d'options offertes par le programme. Ce langage a été conçu spécialement pour SuMo. Il est typé et peut s'utiliser en mode interactif, via des scripts stockés dans des fichiers indépendants, ou les deux. Par exemple, si l'on souhaite enregistrer une séquence de commandes dans un script qui travaille sur une structure donnée, on peut par exemple :

1. donner en ligne de commande le nom du fichier PDB,
2. utiliser un script générique qui lit et transforme ces données,
3. créer et utiliser un script spécifique dont les commandes sont situées dans un autre fichier.

Ceci se traduit par la commande suivante :

```
sumo -interactive \
  -preamble 'let pdb_file = PDB_file "/pdb/pdb2pel.ent";' \
  my_generic_sumo_script.sumo \
  my_specific_script.sumo
```

où le fichier `my_generic_sumo_script.sumo` peut être construit ainsi :

```
(* These commands require pdb_file to be defined! *)
let sumo_data = read pdb_file -densmax 0.5;
```

et le fichier `my_specific_script.sumo` ainsi :

```
(* See which chemical groups are used *)
print sumo_data -file "my-sumo-groups.txt";

(* Screen my database of full structures with my structure *)
let result =
 compare sumo_data DB ("/usr/local/db/sumo/automatic/full");

(* Output the result in human-readable format *)
print result -file "sumo-results.txt";
```

Les options disponibles en ligne de commande s'obtiennent en utilisant l'option `-help` (fig. 3.2 page 39) ou au niveau de la page de `man`.

L'interaction de `sumo` avec le système d'exploitation se limite au système de fichiers. Il n'y a pas de restrictions sur l'emplacement ou le nom des fichiers et des bases de données manipulés. Une description détaillée du langage SuMo et des fonctions fournies est donnée page 101.

```
[pc-bioinfo1] ~/these/rapport % sumo -help
Usage: sumo [input file] [options]

For further details on the command line options, see ‘man sumo’.
To obtain help with sumo functions, start with ‘sumo -i’ and type ‘help
();’.

Options are:
  -interactive  force interactive mode
  -i  same as -interactive
  -non-interactive  force non interactive mode (default)
  -preamble <prog>  parse these commands before the sumo input files
  -pdb-groups <file>  overrides the environment variable PDB_GROUPS
  -remove <file>  removes file at exit
  -disable-sumo-swap  disables automatic sumo swap
  -memory-compaction  not documented
  -no-memory-compaction  not documented
  -sumo-groups <file>  overrides the environment variable SUMO_GROUPS
  -verbose  displays additional information
  -release  displays the release identifier and exits
  -version  displays the version identifier and exits
  -help   Display this list of options
  --help  Display this list of options
```

FIG. 3.2 – Options en ligne de commande du programme `sumo`

3.1.1.3 Niveau supérieur : requêtes de comparaison SuMoQ

L'utilisateur standard utilise l'interface web de SuMo. Celle-ci permet un accès aux services de SuMo sans nécessiter aucune installation locale de logiciel, si ce n'est un navigateur web. Le moyen le plus simple de formuler une requête auprès du serveur web SuMo est de procéder de façon interactive, c'est-à-dire de remplir les champs demandés et de se laisser guider page par page. Ces étapes aboutissent à la génération d'un fichier-requête dans un format particulier, qui va être soumis au programme `sumo-run` chargé de lancer les comparaisons en utilisant `sumo`. Ce mécanisme, s'il est transparent pour l'utilisateur, permet néanmoins de sauvegarder la requête dans un format texte, lisible et réutilisable. Le langage de ces requêtes est appelé *SuMoQ*. Il permet de fournir à l'utilisateur des possibilités de requêtes plus riches que par le mode interactif, tout en s'affranchissant de toute référence au système d'exploitation sous-jacent : en particulier, la notion de fichier a complètement disparu au niveau de cette interface. Voici un exemple de requête permettant de cribler la base de données de sites de fixation de ligands avec la chaîne A de la structure PDB 2PEL :

```
{
  email = "martin_jambon@emailuser.net";
  title = "My query for scanning ligand binding sites";
  {
    subtitle = "Lectin 2PEL";
    scan = {
      database = "ligands";
      pdb_id = "2PEL";
      selection = "Pdb_chain \"A\"";
    } /scan;
  } /
}
```

Une description détaillée des requêtes au format SuMoQ est donné page 110.

3.1.2 SuMo vu sous différents angles

Comment est organisé SuMo ? Telle est la question à laquelle nous allons répondre ici, de façon synthétique. Néanmoins, selon la personne à laquelle cette question est posée, nous allons obtenir des réponses variées. Nous considérons ici trois classes d'individus qui sont confrontés à SuMo. Il s'agit de :

1. les utilisateurs,
2. les administrateurs système,
3. les développeurs.

Si les développeurs ont bien entendu besoin d'avoir en tête le point de vue des administrateurs et des utilisateurs, il n'en est pas de même pour ces deux autres catégories.

3.1.2.1 Le point de vue de l'utilisateur

Un utilisateur normal n'a accès à SuMo que par le serveur web. Il ne sait donc pas *a priori* comment est programmé le système, ni quelles sont les infrastructures qui se cachent derrière le serveur HTTP. L'utilisateur a accès à :

– des pages web générées dynamiquement,
– des données stockables sous forme de fichiers ou accessibles grâce à un identificateur de requête,
– des liens vers des pages web ou des logiciels d'analyse de données indépendantes de SuMo.

La figure 3.3 page 41 synthétise les différentes étapes d'une requête SuMo, ainsi que les entrées, les sorties et les interfaces avec d'autres logiciels. Il est important de noter que le système de requêtes directes SuMoQ permet à n'importe quel logiciel client HTTP de soumettre des requêtes SuMo. Les données intégrales peuvent ensuite être récupérées par le logiciel au format XML. Ainsi, SuMo peut être intégré au sein de n'importe quelle plate-forme informatique, qu'il s'agisse d'un logiciel local ou d'un méta-serveur web sans avoir à modifier SuMo et sans avoir besoin de faire tourner SuMo sur une machine locale.

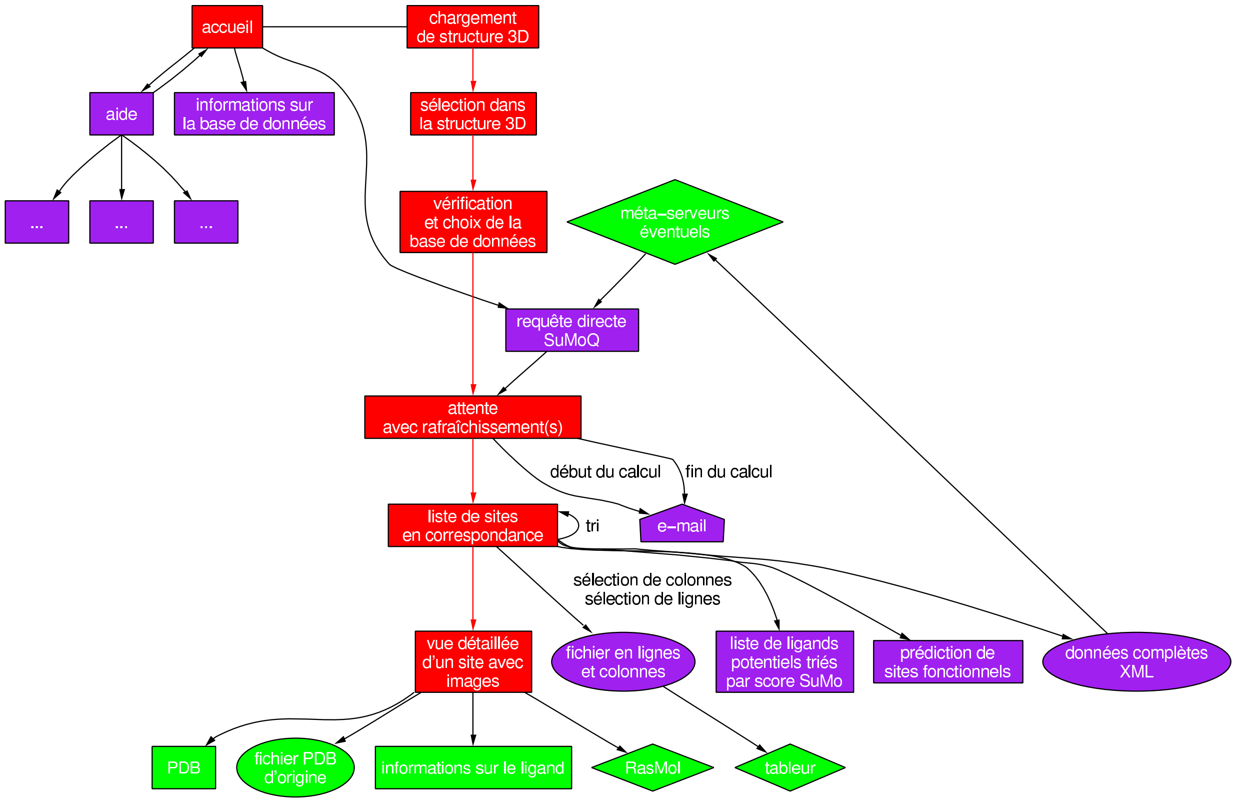


FIG. 3.3 – Organisation de SuMo vue par un utilisateur. En rouge sont représentées les étapes essentielles d'une requête SuMo, en violet les étapes facultatives, et en vert les systèmes et données ne faisant pas partie du système SuMo.

3.1.2.2 Le point de vue de l'administrateur

L'administrateur est la personne qui est chargée d'installer SuMo sur les machines locales. La figure 3.4 page 42 présente les programmes, les processus, les fichiers et les signaux mis en oeuvre lors d'une requête SuMo. La configuration présentée au niveau de cette figure utilise le système de file d'attente avec priorité développé pour SuMo et nommé `jobqueue` (voir section 3.8.1 page 123).

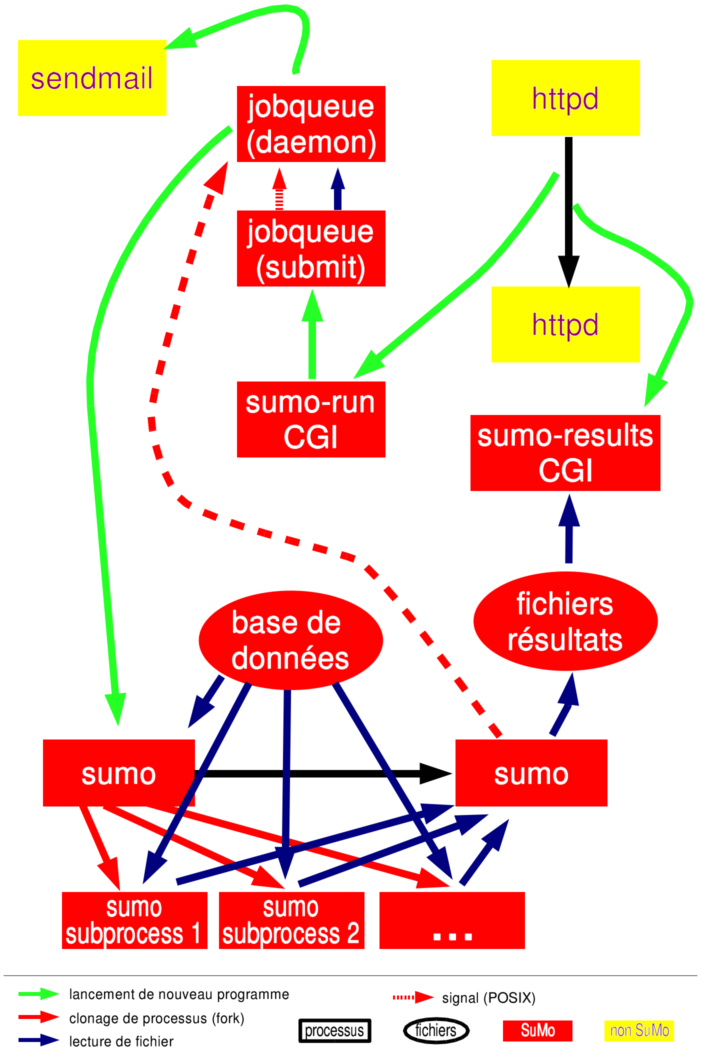


FIG. 3.4 – Interactions au niveau du système d'exploitation lors d'une requête SuMo. L'option permettant d'utiliser un cluster de machines grâce à PBS n'est pas illustrée ici. Les composants en rouge sont spécifiques au système SuMo. Les flèches en rouge continu ou pointillé nécessitent à ce jour une interaction sur la même machine.

3.1.2.3 Le point de vue du programmeur

Le programmeur, même peu familier avec le principe de fonctionnement de SuMo, doit avoir une bonne vue d'ensemble sur l'arborescence qui contient les fichiers-sources. La figure 3.5 page 44 présente l'arborescence des répertoires et de quelques fichiers essentiels. Le système d'archivage CVS est utilisé pour gérer les versions successives de chacun des fichiers. Les fichiers-sources de SuMo peuvent être assez naturellement classifiés en trois catégories :

1. les accessoires de compilation,
2. les sources des programmes utilisables en ligne de commande,
3. les sources des programmes utilisés pour l'interface web.

Les accessoires de compilation développés pour SuMo comprennent notamment les fichiers `Makefile`, le fichier `configure`, le fichier `VERSION`, le programme `substitute`[2] et l'extension syntaxique Printfer décrite section 3.7.2.5 page 121.

Les différents programmes utilisables en ligne de commande permettent d'effectuer des tâches variées. Certains sont des utilitaires lancés généralement à la main comme le programme de génération de la base de données `build-sumo-db` ou l'utilitaire `sumo-extend` d'extension automatique du fichier de configuration `pdb_groups`[3]. D'autres programmes sont utilisés par les programmes CGI mais peuvent être également utilisés directement. Il s'agit notamment de `sumo`, et des utilitaires `cns2pdb`[4], `pdb2tree`[5], `seqinfo`[6], `sumo-sort`[7] et `sumo-columns`[8].

Les programmes utilisés pour l'interface web de SuMo comportent les différents programmes CGI : `sumo-help`, `sumo-database`, `sumo-welcome`, `sumo-select`, `sumo-check`, `sumo-run`, `sumo-results` et `sumo-focus`. D'autres programmes ont été créés pour le serveur SuMo, il s'agit de `sumo-sign`[9], `jobqueue` (voir section 3.8.1 page 123) et de `sumo-clean`[10].

---

[2] remplace `@@TOTO@@` par la valeur du paramètre `TOTO`

[3] contient la correspondance entre les notations des atomes au format PDB et l'élément de la classification périodique correspondant

[4] conversion du format PDB X-Plor/CNS en un format PDB standard

[5] conversion du format PDB en un format arborescent

[6] repérage des chaînes redondantes dans les structures 3D

[7] tri de données en lignes et colonnes

[8] génération de fichiers en lignes et colonnes à partir des résultats de `sumo`

[9] signature cryptographique de données

[10] suppression des fichiers temporaires publics

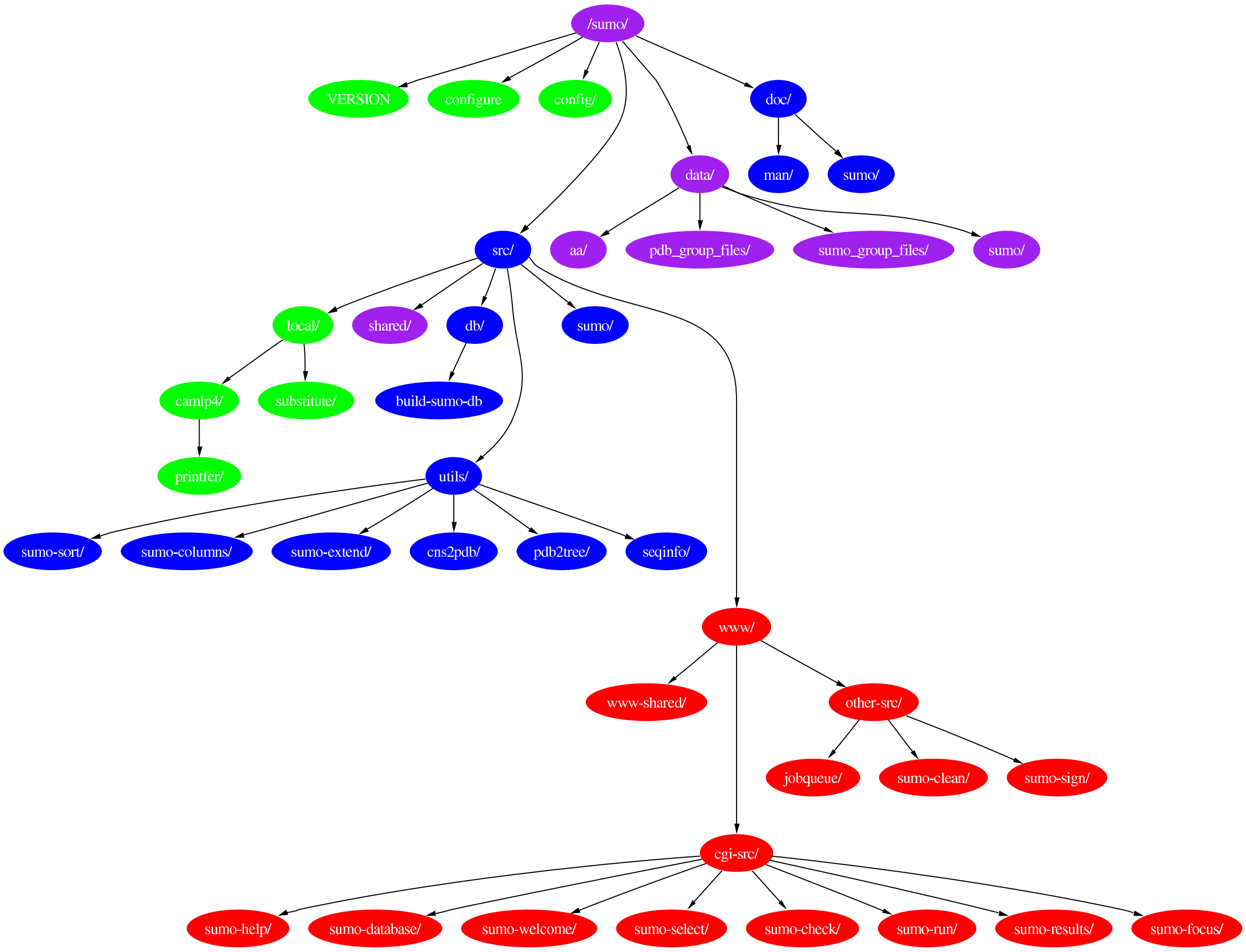


FIG. 3.5 – Arborescence simplifiée et colorée des fichiers-sources de SuMo. En vert : outils de compilation, en bleu : outils utilisables en ligne de commande, en rouge : programmes utilisés pour l'interface web, en violet : code et données partagés.

3.2 La comparaison 2 à 2 de structures 3D

Cette section décrit le coeur du système SuMo, c'est-à-dire comment à partir des données structurales telles que celles présentes dans la PDB on identifie des régions similaires entre deux structures 3D de macromolécules. Les étapes essentielles sont schématisées sur la figure 3.6.

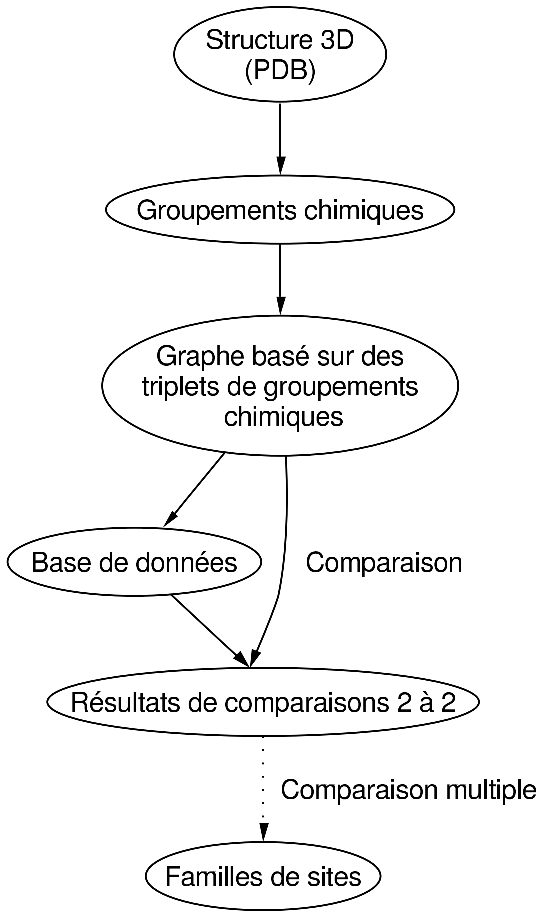


FIG. 3.6 – Étapes majeures des comparaisons par SuMo

L'étape de comparaison 2 à 2 est centrale et limitante en temps de calcul. Pour cela, un pré-traitement des données est effectué au préalable de façon à les stocker sous une forme adéquate pour une comparaison pertinente et rapide. Le coeur de la comparaison permet de générer rapidement des listes de correspondances entre les groupements chimiques des complexes moléculaires comparés.

Les comparaisons multiples et la prédiction automatisée de sites de fixation de ligands reposent sur une utilisation des résultats de comparaisons 2 à 2 entre structures ou sous-structures. Elles font l'objet de sections ultérieures.

3.2.1 Découpage en groupements chimiques

SuMo est fondé sur une description des structures tridimensionnelles de macromolécules par des groupements chimiques. La notion de *groupement chimique* dans SuMo correspond à une modélisation d'une certaine région de la structure tridimensionnelle d'un ensemble de molécules. Les limites de cette définition sont plus liées à la nature de l'heuristique de comparaison employée qu'à la conception traditionnelle que l'on peut se faire d'un groupement chimique. Ces limites seront discutées ultérieurement.

3.2.1.1 Préliminaires

Quelques préliminaires sont nécessaires avant de manipuler les coordonnées des atomes des molécules présentes dans un fichier PDB. Nous avons besoin de distinguer les différentes molécules présentes dans un complexe et de connaître leur taille, notamment afin de faire la distinction entre ce qui peut être pris en compte par SuMo et ce qui ne doit pas l'être.

L'identification des molécules Nous aurons besoin de distinguer les différentes molécules non liées covalemment présentes dans une structure donnée.

En effet, un aspect important de SuMo est la prédiction de sites de fixation de ligands par comparaison avec des sites de fixation de ligands présents dans la PDB. Pour cela, il faut avant tout être capable de distinguer les différentes molécules présentes dans une structure expérimentale. Cette information n'est pas donnée de façon fiable dans les fichiers PDB. Les fichiers PDB sont organisés en groupements chimiques tels que ceux représentant les monomères classiques des protéines et des acides nucléiques (ALA, CYS, ..., A, C, G, T, U). Ceux-ci ont leurs atomes identifiés par le mot-clé ATOM. L'autre catégorie est identifiée par le mot-clé HETATM et comprend tous les autres groupements chimiques (HOH, ATP, CA, GLC, ...). L'information de chaîne donnée dans les fichiers PDB correspond aux chaînes polypeptidiques et polynucléotidiques classiques, son rôle est néanmoins très flou en ce qui concerne les ligands. Les groupements chimiques dont les atomes sont repérés avec le mot-clé HETATM peuvent être liés covalemment à d'autres groupements ATOM ou HETATM appartenant à une protéine ou à n'importe quel autre type de molécule.

Au niveau de SuMo, une structure de données intermédiaire est générée à partir du fichier PDB. Elle permet de reformater les données de la PDB dans un format arborescent, à la syntaxe non ambiguë, et où certaines informations complémentaires ont été précalculées. Cette structure de données intermédiaire peut être exportée et importée dans la syntaxe arborescente décrite page 111. Elle peut également être exportée dans un format XML, et visualisée au niveau du serveur web SuMo. Cette structure de données élimine les distinctions entre ATOM et HETATM, et indique la molécule d'appartenance de chaque atome.

Pour déterminer la molécule à laquelle appartient chaque atome d'une structure 3D complète, il suffit de repérer quelles sont les liaisons covalentes entre les différents atomes donnés dans le fichier PDB. La limitation de ce système apparaît lorsque des portions de structure ne sont pas résolues. Ceci peut alors conduire à plusieurs molécules apparentes là où il n'y en a qu'une. Le résultat obtenu n'est néanmoins pas dénué de sens, dans la mesure ou les portions non résolues correspondent généralement à des régions hautement flexibles. Dans ce cas, considérer comme deux molécules distinctes deux éléments reliés par une boucle flexible de structure non résolue met en évidence leur indépendance et la possibilité pour ces éléments d'interagir comme deux partenaires indépendants.

L'algorithme utilisé se base sur un graphe dont les n sommets sont les atomes présents dans la structure et les arêtes représentent les liaisons covalentes, puis l'isolement des sous-graphes indépendants :

1. Pour chaque atome, recherche des voisins dans un rayon $r_{\max}$ et connexion avec les atomes dont la distance et la nature permettent de former une liaison covalente. Coût : $O(n)$
2. Marquer chaque sommet non marqué avec un nouvel identifiant et marquer récursivement avec le même identifiant chacun de ses voisins.

L'algorithme 1 page 48 précise la procédure de marquage des sommets.

Les liaisons covalentes sont actuellement attribuées sur des critères de distance. Les critères utilisés sont les suivants : si A et B sont 2 atomes, alors on considère qu'ils sont liés par une liaison covalente dès lors que leur distance est inférieure à $d_{\max}$:

$$d_{\max} = \begin{cases} 1{,}3\ \text{Å} & \text{si un des deux atomes est un hydrogène} \\ 1{,}9\ \text{Å} & \text{sinon} \end{cases}$$

Détecter les liaisons hydrogène Nous verrons que SuMo propose la définition de groupements chimiques donneurs et accepteurs de liaisons hydrogène libres. Un *donneur ou accepteur de liaison hydrogène libre* n'est pas

Algorithme 1 Marquage des sous-graphes indépendants

Require a graph $G = (V, E)$
function Mark (v, k)
  if v is marked **then** do nothing
  else
    Set-Mark (v, k)
    for all $v' \in$ Neighbors (v) **do** Mark (v', k)
$k \leftarrow 1$
Result $\leftarrow \varnothing$
for all $v \in V$ **do**
  if v is marked **then** do nothing
  else
    Mark (v, k)
    $k \leftarrow k + 1$

impliqué dans une liaison hydrogène ou ne risque pas d'être impliqué dans une liaison hydrogène suite à un petit changement de conformation locale. Un donneur de liaison hydrogène peut former zéro ou une liaison hydrogène alors qu'un accepteur peut éventuellement être impliqué dans plusieurs liaisons hydrogène.

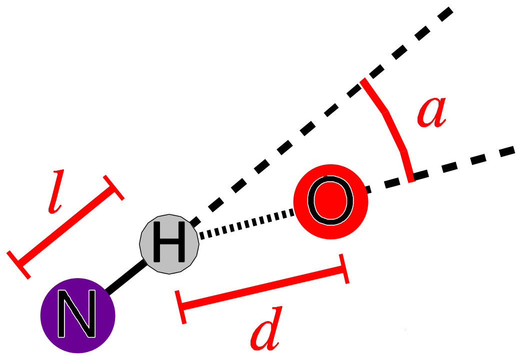


Fig. 3.7 – Modèle utilisé pour l'identification des liaisons hydrogène. Exemple d'une interaction $NH^{\delta+}$–$O^{\delta-}$.

La figure 3.7 introduit le modèle utilisé pour détecter les liaisons hydrogène, avec l'exemple d'une interaction $NH^{\delta+}$–$O^{\delta-}$. Les critères qui permettent de considérer qu'il y a une liaison hydrogène sont les suivants :

$$\begin{cases} 0 \leq a \leq \alpha \\ |d - \beta| \leq \gamma \end{cases}$$

où α, β et γ sont des paramètres. Les valeurs utilisées pour les liaisons hydrogène impliquant des atomes d'azote ou d'oxygène sont les suivantes :

$$\begin{cases} \alpha = 70° \\ \beta = 1,95\text{Å} \\ \gamma = 0,45\text{Å} \end{cases}$$

Ces paramètres englobent des positions relatives qui sont loin de la liaison hydrogène optimale en termes d'énergie. Néanmoins on suppose que des changements locaux de conformation peuvent rapidement entraîner une stabilisation de la liaison hydrogène. Rappelons que le but est de mettre en place une heuristique permettant de conserver les donneurs et accepteurs de liaisons hydrogènes qui sont prêts à interagir avec un partenaire exterieur, et d'éliminer les autres.

Le paramètre l désigne la longueur de la liaison covalente entre l'atome d'hydrogène δ^+ et l'atome lourd auquel il est lié. Cette distance est imposée : la position de l'atome d'hydrogène est calculée à partir de la direction de la liaison et de l, puisque les fichiers PDB n'indiquent généralement pas cette position. Cette valeur est fixée à 1 Å par défaut. Le calcul de la position des atomes d'hydrogène est donc effectué lorsque c'est nécessaire. Pour certains groupements tels que $-NH_3^+$, le positionnement des 3 atomes d'hydrogène est plus délicat en raison de la libre rotation du groupement chimique. Ce positionnement est effectué en considérant chacune des rotations qui permet de former une liaison hydrogène de façon optimale. La rotation conservée est une des positions qui maximise le nombre de liaisons hydrogène.

Définition d'un ligand La notion de *ligand*, telle qu'elle est entendue dans le cadre de SuMo repose sur les notions suivantes :
- un ligand est une molécule à part entière ;
- un ligand est susceptible d'établir des complexes temporaires avec des macromolécules biologiques ;
- un ligand d'une macromolécule ne doit pas être nécessaire pour obtenir un repliement stable de cette macromolécule ;
- un ligand n'est pas une macromolécule.

La notion de ■ complexe temporaire ■ entre un ligand et une macromolécule indique une interaction par liaisons non covalentes, pouvant être rompue sans déstructuration globale de la macromolécule. Un ligand tel qu'on l'entend ici n'est donc pas un élément obligatoire pour la stabilisation d'une structure plurimoléculaire.

Pour la définition automatique des ligands au sein de SuMo, les critères suivants sont utilisés :

– taille inférieure à 50 atomes non hydrogène,
– au moins un des monomères n'est pas un monomère bien connu.

Par *monomère bien connu*, nous entendons un des groupements parmi :

– un des 20 acides aminés classiques ALA, CYS, ASP, GLU, PHE, GLY, HIS, ILE, LYS, LEU, MET, ASN, PRO, GLN, ARG, SER, THR, VAL, TRP, TYR ;
– la sélénométhionine MSE ;
– l'eau HOH

Les nucléotides ne sont pas considérés comme étant des monomères bien connus. Leurs oligomères peuvent donc être considérés comme ligands dans la mesure où ils ne dépassent pas la limite de taille imposée. La liste des monomères bien connus est donnée dans le fichier de définition des groupements chimiques de SuMo (voir section A page 153).

La nomenclature des ligands est la séquence des identificateurs PDB des monomères les constituant, ordonnés par chaîne puis par numéro de groupement chimique (ex : CA, GLC-GLC, ATP, . . .). Cette nomenclature possède l'avantage d'être lisible mais elle a l'inconvénient d'être ambiguë dans certains cas. En effet, les monomères de certains ligands comme les oligosaccharides peuvent être liés de plusieurs façons différentes.

3.2.1.2 Le groupement chimique

Dans SuMo, la notion de groupement chimique est centrale. Chaque structure de macromolécule traitée par SuMo est tout d'abord convertie de façon irréversible en un ensemble de groupements chimiques. Plusieurs *types de groupements chimiques* peuvent être définis. Chacun des types de groupements chimiques permet de représenter une propriété. Seuls les groupements chimiques de même type peuvent être comparés et, le cas échéant, être considérés comme équivalents.

Le type À chaque type de groupements chimiques est associé un identificateur, par exemple `hydroxyl` ou encore `delta_minus`. Un type de groupements chimiques doit être vu comme la représentation d'une propriété chimique localisable au sein de structures 3D de macromolécules telles que les protéines. Tous les groupements chimiques d'un même type doivent être construits sur le même modèle géométrique, afin de pouvoir les comparer et éventuellement les superposer.

Le coefficient Chaque type de groupements chimiques est associé à un coefficient. Ce coefficient caractérise l'importance fonctionnelle de ce type de groupements chimiques. Ce coefficient entre en jeu dans plusieurs heuristiques au sein de SuMo. C'est un nombre positif de valeur maximale égale à 1. Toute distance considérée comme un *rayon d'influence* autour d'un groupement chimique est multipliée au préalable par ce facteur.

Les informations positionnelles Chaque groupement chimique est localisé dans l'espace. Cette localisation repose sur trois sortes de points :
- la position physique,
- la position fonctionnelle,
- des positions-cibles.

La position physique La *position physique* correspond à une position moyenne des constituants physiques — en général des atomes — responsables de l'existence du groupement chimique.

Cette position est utilisée en particulier pour sélectionner les triangles de groupements chimiques à utiliser pour représenter les structures 3D.

La position fonctionnelle La *position fonctionnelle* est la localisation de la propriété fonctionnelle du groupement chimique. Elle peut être égale ou non à la position physique.

C'est cette position qui est utilisée pour les comparaisons de structures 3D par SuMo.

Par exemple, la position fonctionnelle d'un donneur de liaison hydrogène est la position de l'accepteur de liaison hydrogène potentiel.

Les positions-cibles Les *positions-cibles* sont les positions à considérer pour l'interaction avec un ligand potentiel. Dans la mesure du possible, elles indiqueront les localisations relatives du ligand les plus probables.

Ceci permet d'améliorer la définition des sites d'interactions, en particulier des sites de fixation de ligands.

Par exemple, le groupement `aromatic` défini dans la version actuelle de SuMo a deux positions-cibles symétriques situées de part et d'autre du cycle aromatique. Pour un donneur de liaison hydrogène, on aura une position-cible égale à la position fonctionnelle.

Les données communes complémentaires Certaines données sont associées à tous les types de groupements chimiques mais sont facultatives, c'est-à-dire qu'elles sont utilisées dans les étapes de comparaison mais que leur suppresssion n'empêche pas l'algorithme de comparaison central de SuMo d'être utilisé.

L'enfouissement L'*enfouissement d'un groupement chimique* est basé sur le calcul de densité atomique au niveau de sa position fonctionnelle. Les propriétés de la fonction de densité atomique utilisée est décrite en détail section 3.6.1 page 82.

L'orientation L'orientation par rapport à la surface de la macromolécule est donnée par le vecteur $\overrightarrow{CP}$ où P est la position fonctionnelle du groupement chimique considéré et C le barycentre des atomes considérés dans le calcul de densité atomique. C est appelé *centre de masse local*.

L'environnement atomique L'environnement atomique est l'ensemble des atomes situés autour de la position fonctionnelle d'un groupement chimique, dans un rayon suffisant pour permettre la comparaison de formes (voir sections 3.2.5.1 page 67 et 3.6.2 page 83).

Les données spécifiques à un type À chaque type de groupement chimique peut être joint n'importe quel type de données complémentaires utilisées dans la comparaison 2 à 2 des paires de groupements chimiques. Une telle opération nécessite néanmoins l'ajout de code au niveau du programme `sumo`.

Afin de pouvoir définir aisément des types de groupements chimiques utiles sans avoir à reprogrammer `sumo`, SuMo fournit un langage de définition des types de groupements chimiques. Ces définitions résident dans un fichier et sont lues au démarrage de `sumo`. Ce langage fournit un ensemble de constructeurs de types prédéfinis.

A chaque type de groupement chimique est associé ce que nous appellerons un *variant géométrique* ainsi que des paramètres qui y sont associés.

Les variants géométriques Il existe actuellement 5 variants géométriques qui sont définis et implémentés dans SuMo. La figure 3.8 page 53 en donne une illustration à l'aide d'exemples de groupements chimiques. Ces variants géométriques sont numérotés de 1 à 5 :

Variant 1 C'est le variant nul : aucune information géométrique particulière n'est représentée. Toutes les directions de l'espace sont équivalentes.

Variant 2 Il consiste en un vecteur qui n'a pas de symétrique.

Variant 3 Il est constitué de 2 vecteurs opposés fonctionnellement équivalents.

Variant 4 Il est constitué de 2 vecteurs orthogonaux, dont un des deux est fonctionnellement équivalent à son opposé.

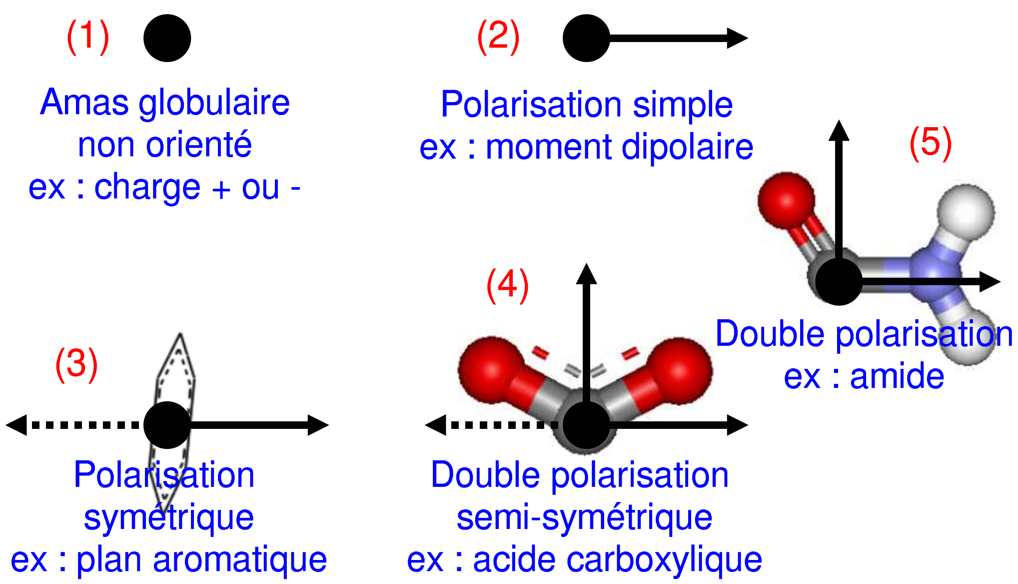


FIG. 3.8 – Exemples illustrant les 5 types de variants géométriques définis et mis en oeuvre dans SuMo.

Variant 5 Il est constitué de 2 vecteurs orthogonaux.

Les différentes fonctions prédéfinies qui permettent de définir des types de groupements chimiques sont appelés *constructeurs de types*. Chacun de ses constructeurs permet d'associer certains motifs dans la structure atomique des molécules à un ou plusieurs groupements chimiques utilisant un variant géométrique donné. Le tableau 3.2 page 54 présente une liste des différents constructeurs définis et les variants géométriques utilisés pour les groupements chimiques qu'ils génèrent.

Les caractéristiques principales des constructeurs de types de groupements chimiques sont les suivantes :

`Point` permet de générer un groupement chimique sans géométrie particulière à partir d'une position moyenne d'atomes (variant de type 1).

`Polar` permet de générer un variant de type 2 arbitraire.

`Biplan` permet de générer un variant de type 3 arbitraire.

`Plan` permet de générer un variant de type 4 arbitraire.

`Chiral` permet de générer un variant de type 5 arbitraire.

`Virtual` permet de générer un groupement de type 1 dont la position fonctionnelle et la position-cible sont différentes de la position physique. Le groupement est généré uniquement si la position est libre.

`Delta_plus` permet de générer un groupement chimique considéré comme

Constructeur de type	Variant géométrique
`Point`	1
`Polar`	2
`Biplan`	3
`Plan`	4
`Chiral`	5
`Virtual`	1
`Delta_plus`	2
`Delta_plus_plan`	2
`Delta_plus_multiple`	2
`Delta_minus`	2

Tab. 3.2 – Correspondance entre constructeurs de types de groupements chimiques et variants géométriques utilisés.

un donneur de liaison hydrogène libre, à partir de 2 positions.

`Delta_plus_plan` permet de générer un groupement chimique considéré comme un donneur de liaison hydrogène libre, à partir de 3 positions et d'un angle (ex : amide $-CONH_2$).

`Delta_plus_multiple` permet de générer plusieurs groupements chimiques considérés comme des donneurs de liaisons hydrogène libres, à partir de 2 positions, d'un angle et du nombre d'atomes d'hydrogène (ex : $-NH_3^+$).

`Delta_minus` permet de générer un groupement chimique considéré comme un accepteur de liaison hydrogène libre, à partir de 2 positions et du nombre minimal de liaisons hydrogène acceptées pour être considéré comme non-libre.

Les paramètres Chaque constructeur de types de groupements chimiques attend des paramètres, dont certains possèdent des valeurs par défaut. Ces paramètres comprennent en particulier :
- la définition des positions-cibles,
- la translation de la position physique vers la position fonctionnelle,
- les déviations d'angles maximales autorisées lors des étapes de comparaison.

Annotation des groupements chimiques Des informations complémentaires peuvent être associés aux différents groupements chimiques, sans qu'elles n'entrent en jeu dans les heuristiques de comparaison.

Groupement PDB de rattachement Le *groupement PDB de rattachement* d'un groupement chimique SuMo peut être défini lorsque cela a un sens. Les informations le concernant sont la chaîne (ex : A), le nom du groupement PDB (ex : ALA) et son numéro. Dans la version 4.4 de SuMo, le mode de définition des groupements chimiques SuMo conduit automatiquement à la prise en compte de ces informations.

Par mesure de commodité, ces informations peuvent être utilisées comme critère de sélection des groupements chimiques (section 3.7.1.3 page 106). Néanmoins, elles ne sont pas considérées comme fiables pour identifier automatiquement les différents composants structuraux d'un complexe macromoléculaire.

Molécule de rattachement Les différentes molécules identifiées au cours du prétraitement des données structurales (section 3.2.1.1 page 46) sont identifiées par un numéro. Chaque groupement chimique est associé à une *molécule de rattachement* lorsque cela a un sens.

De même que précédemment, cette information est utilisable pour sélectionner certains groupements chimiques. Ce moyen de sélection nécessite néanmoins de connaître le numéro de la molécule que l'on souhaite sélectionner. Actuellement, cette fonctionnalité est notamment utilisée pour sélectionner automatiquement les sites d'interaction avec des ligands.

Étiquettes (labels) Un étiquetage optionnel des groupements chimiques permet de différencier chaque groupement chimique instancié, lorsque toutes les autres annotations sont identiques. Par exemple, nous pouvons définir jusqu'à trois accepteurs de liaisons hydrogène libres pour un aspartate donné : un au niveau de la chaîne principale et deux au niveau de la fonction carboxylate. Ces groupements chimiques sont tous du type `delta_minus`, appartiennent au même polypeptide et au même acide aminé. Nous attribuerons donc les *étiquettes* complémentaires `backbone`, `e1` et `e2` à ces groupements chimiques afin de pouvoir les identifier sans ambiguïté.

L'étiquette peut être utilisée comme un moyen de sélection au même titre que les informations évoquées dans les paragraphes précédents. Actuellement, un étiquetage automatique est proposé au niveau du langage de définition des groupements chimiques.

Marquage de régions arbitraires Des ensembles arbitraires de groupements chimiques d'une structure 3D donnée peuvent être annotés. Ces régions annotées sont ensuite utilisées lors de l'affichage des résultats. Une annotation concernant les sites de fixation de ligands est réalisée automatiquement lors de la construction de la base de données de structures complètes. Elle peut également être réalisée par l'utilisateur lors de la soumission d'une requête par le système SuMoQ.

Ainsi, chaque groupement chimique est associé à la liste de régions marquées auxquelles il appartient. Ces régions sont annotées par un commentaire structuré de façon arbitraire, et certaines caractéristiques concernant leur structure et leur taille y sont associées. Pour le mode de définition des régions marquées, se reporter à la section sur SuMoQ, page 110.

Syntaxe du langage de définition des types de groupements chimiques Le langage qui a été mis en place est assez complexe et nous ne donnons ici que les grandes lignes de sa syntaxe. Le listing complet du fichier de définition des groupements chimiques utilisé dans la version 4.4 de SuMo est joint en annexe page 153.

Le principe de la définition de types de groupements chimiques repose sur la notion de motif à différents niveaux. `<<"ALA" "GLY">>` signifie (selon la nomenclature PDB) ■ le groupement chimique PDB courant est une alanine ou une glycine ■. `<<"ALA">>[-1]` signifie ■ le groupement chimique PDB précédent dans la séquence est une alanine ■. De façon similaire, `<<"N" "C" "CA">>` signifie ■ le groupement considéré doit posséder un des atomes N, C ou CA ■. Un motif `aa.atom` permet de définir une position correspondant au premier atome qui valide le motif. Une séquence `aa1.atom1 aa2.atom2` permet de définir la moyenne des positions des atomes qui valident le motif. Pour y voir un peu plus clair, voici un fichier de définition des types de groupements chimiques valide syntaxiquement et commenté :

```
(* This is a comment *)

(* These are the types of chemical groups that we want to use: *)
Export (my_first_chemical_group
        delta_minus);    (* my_favorite_chemical_group
                            is defined but not used *)

Not_special ("HOH");   (* indicates that water is not an
                          interesting monomer for a ligand *)

aa = <<"ALA" "GLY">>;   (* definition of a set of PDB groups named aa *)
arg = <<"ARG">>;

my_first_chemical_group = (* this is the identifier of the new type of
                             chemical groups *)

    Delta_plus [backbone]    (* [backbone] is an optional label *)
          (aa.<"N">,      (* matches atoms "N" from group "ALA" or "GLY" *)
```

```
          aa[-1].<"C"> aa.<"CA">,  (* aa[-1] means any aa in the same chain
                                      with index = current_index - 1 *)
          aa.<"N">,
          target = 0.0,            (* optional field *)
          functional_shift = 2.8,  (* optional field *)
          angle = 140.)            (* optional field *)

  | Delta_plus         (* other pattern *)
            (arg.<"NE">,
             arg.<"CD"> arg.<"CZ">, (* the point is the average of
                                       the 2 atoms matched with
                                       arg.<"CD"> and arg.<"CZ"> *)
             arg.<"NE">,
             target = 0.0,
             functional_shift = 2.8,
             angle = 140.)
;

delta_minus {0.6} =  (* Weight is 0.6, not 1 *)
  Delta_minus [backbone] (aa.<"O">,
                          aa.<"C">,
                          aa.<"O">,
                          1,
                          target = 1.)
;

my_favorite_chemical_group =
  Point                          (* Point is the simplest constructor *)
    (<<"PHE">>.<"CA"> <<"PHE">>[1].<"CA">);


Hbond (my_first_chemical_group, delta_minus); (* we define a hydrogen bond,
                                                 using the default
                                                 parameters *)
```

Groupements chimiques fantômes Les *groupements chimiques fantômes* sont des groupements chimiques générés systématiquement par SuMo mais qui ne sont pas pris en compte pour former la structure de données utilisée pour la comparaison. Ils sont utilisés pour représenter les atomes des ligands afin d'être utilisés dans les processus de sélection de groupements chimiques (section 3.7.1.3 page 106).

Tout atome qui n'est pas reconnu comme étant un atome d'hydrogène et qui appartient à un groupement PDB qui n'a pas été déclaré comme bien connu va constituer un groupement chimique fantôme. Un groupement PDB bien connu est un groupement qui n'a pas été sélectionné à l'aide de la directive `Not_special`. Sa position est la position de l'atome considéré. Son type est généré automatiquement à partir du nom du groupement PDB comme dans les exemples suivants :

$$\texttt{GLC} \rightarrow \texttt{pdb\_glc}$$
$$\texttt{A} \rightarrow \texttt{pdb\_a}$$

3.2.2 Association en triplets

L'heuristique de comparaison conçue pour SuMo repose sur la construction de triplets de groupements chimiques, auxquelles sont associées différentes informations. Les triplets générés constituent les sommets d'un graphe qui constitue la structure de données utilisée pour la comparaison de structures 3D.

Dans cette section sont décrites le choix des triplets, les données qu'ils contiennent et la construction du graphe de triplets.

3.2.2.1 Choix des triplets

L'utilisation de triplets de groupements chimiques permet de représenter des micro-sites de taille fixe, aisément comparables. Les triplets sont choisis de façon à correspondre au mieux à des unités structurales qui sont susceptibles d'avoir une certaine pertinence fonctionnelle. Leur taille, c'est-à-dire la longueur des arêtes des triangles formés ne devra pas être trop grande.

Étant donné n groupements chimiques représentant une structure 3D de molécule, le nombre de triplets ordonnés distincts que l'on peut générer est C_n^3, c'est-à-dire $O(n^3)$. Pour des raisons de temps de calcul et de mémoire, il n'est pas envisageable de générer tous les triplets possibles dans une macromolécule moyenne, où le nombre de groupements chimiques définis dépasse fréquemment 1000.

Ainsi, pour les deux raisons citées précédemment, le nombre de triplets de groupements chimiques formés sera largement inférieur au nombre de triplets possibles. Les critères de sélection sont établis manuellement, en essayant de satisfaire au mieux les contraintes suivantes :

- représentation du plus grand nombre possible de sites fonctionnels,
- représentation du plus petit nombre possible de sites non associés à une fonction biologique particulière,

– comparaison suffisamment rapide pour faire du criblage de bases de données,
– stockage et accès aux données possibles dans des limites raisonnables.

Pour cela, il faut éviter de générer un trop grand nombre de triplets tout en essayant de conserver ceux qui ont le plus de chances d'être associés à un phénomène biochimique intéressant.

Nous appellerons *triangles physiques* les triangles constitués par les positions physiques des groupements chimiques des triplets, et *triangles fonctionnels* les triangles formés par les positions fonctionnelles des groupements chimiques. Lorsque le type de triangle n'est pas précisé, il s'agit des triangles fonctionnels.

Les critères de sélection et leurs paramètres sont donnés ici pour la version 4.4 mais évoluent fréquemment d'une version de SuMo à l'autre.

Longueur des arêtes Plus le groupement chimique est de grande taille, plus il va avoir une influence sur une grande région. Ainsi, les critères de sélection sur la longueur des arêtes sont établis en fonction des coefficients des groupements chimiques. Pour 2 groupements chimiques de positions physiques p_1 et p_2 et de coefficients k_1 et k_2, tout triplet contenant ces groupements chimiques devra vérifier la contrainte :

$$\frac{k_1 + k_2}{2} \cdot l_{\min} \leq \|p_1 - p_2\| \leq \frac{k_1 + k_2}{2} \cdot l_{\max}$$

où $l_{\max}$ vaut 8,5 Å et $l_{\min}$ vaut $0{,}3.l_{\max}$.

Somme de la longueur des arêtes Afin d'éviter une surabondance de triplets de grande taille dont les sommets sont déjà impliqués dans suffisamment de petits triplets, une borne maximale pour la somme des arêtes des triangles est imposée.

Afin d'éviter de former des triangles formés par des groupements chimiques reliés rigidement entre eux, une taille minimale est également imposée.

Ainsi, tout triplet de positions physiques p_1, p_2 et p_3 doit vérifier la contrainte :

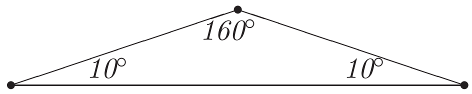
$$s_{\min} \leq \|p_1 - p_2\| + \|p_2 - p_3\| + \|p_3 - p_1\| \leq s_{\max}$$

où $s_{\min}$ vaut 6 Å et $s_{\max}$ vaut 20 Å.

Angles La définition d'un plan à partir de 3 positions a d'autant moins d'intérêt que ces points sont près d'être alignés. C'est ici la position fonctionnelle des groupements chimiques qui est considérée[11] puisque c'est elle qui est prise en compte lors de la comparaison des triplets.

Pour éviter ce cas de figure, les angles des triangles ne doivent être ni trop grands ni trop petits. Chaque angle doit être compris entre 10° et 160°.

En effet, nous considérons le triangle suivant comme triangle limite :



3.2.2.2 Propriétés des triplets

Un triplet de groupement chimiques est l'entité minimale qui entre en jeu dans la comparaison de structures 3D de macromolécules par SuMo. Cet objet possède, outre les propriétés déjà associées à chaque groupement chimique, des propriétés d'ordre supérieur.

Repère du triplet À chaque triplet T est associé un triangle formé par les positions fonctionnelles des groupements chimiques. Nous associons à ce triangle un repère, que l'on appellera *repère du triplet*. Le repère du triplet T noté R_T est défini à partir des positions fonctionnelles ordonnées p_1, p_2 et p_3 de la façon décrite au paragraphe suivant. L'intérêt de ce repère est que pour superposer 2 triplets T_1 et T_2, nous considérons qu'il suffit d'exprimer au préalable l'environnement de T_1 dans le repère R_{T_1} et l'environnement de T_2 dans le repère R_{T_2}.

Soit O la position moyenne des points p_1, p_2 et p_3. O constitue le centre du repère du triplet T. Notons v_1, v_2 et v_3 les vecteurs $\overrightarrow{Op_1}$, $\overrightarrow{Op_2}$ et $\overrightarrow{Op_3}$. Soit u le vecteur unitaire caractérisant le plan défini par p_1, p_2 et p_3 tel que le produit mixte de (v_1, v_2, u) soit positif. Nous appellerons *vecteurs standard* du triplet T les vecteurs liés et unitaires $\tilde{v}_1$, $\tilde{v}_2$ et $\tilde{v}_3$ tels que la somme S suivante soit minimale :

$$S = \widehat{v_1, \tilde{v}_1} + \widehat{v_2, \tilde{v}_2} + \widehat{v_3, \tilde{v}_3}$$

où $\widehat{a, b}$ désigne l'angle non orienté formé par les deux vecteurs a et b, donné par $\cos^{-1} \frac{a.b}{\|a\|.\|b\|}$. Les vecteurs standard $\tilde{v}_1$, $\tilde{v}_2$ et $\tilde{v}_3$ s'obtiennent par rotation des vecteurs v_1, v_2 et v_3 autour de l'axe défini par u respectivement par les

[11]SuMo version 4.5

angles α_1, α_2 et α_3 suivants :

$$\alpha_1 = \frac{1}{2}\left(\widehat{v_1, v_2} - \widehat{v_3, v_1}\right)$$
$$\alpha_2 = \frac{1}{2}\left(\widehat{v_2, v_3} - \widehat{v_1, v_2}\right)$$
$$\alpha_3 = \frac{1}{2}\left(\widehat{v_3, v_1} - \widehat{v_2, v_3}\right)$$

Le repère R_T du triplet T est le repère orthonormé défini par $(O, \tilde{v}_1, u \wedge \tilde{v}_1, u)$ où $\wedge$ désigne le produit vectoriel.

Propriétés héritées des groupements chimiques Nous pouvons transposer un certain nombre de propriétés des groupements chimiques au niveau des triplets.

Types de triplets Le *type d'un triplet* de groupements chimiques est le triplet ordonné des types des groupements chimiques. Par exemple, un triplet constitué de groupements `delta_plus`, `aromatic` et `guanidinium` aura pour type (`aromatic`, `delta_plus`, `guanidinium`).

A chaque type de triplet correspond un nombre de permutations égal à 1, 2 ou 6 selon que tous les groupements chimiques sont de types différents, ou que deux sont identiques, ou que les trois types sont identiques.

L'orientation des groupements chimiques Les variants géométriques des groupements chimiques sont repris en les plaçant dans le repère du triplet.

L'environnement atomique L'environnement atomique est également repris : les atomes considérés autour de chaque groupement chimique sont fusionnés en un seul ensemble puis leurs coordonnées sont exprimées dans le repère du triplet.

> *Étant donné que la position des atomes environnants est actuellement utilisée uniquement pour définir la forme de l'environnement local, et que l'heuristique utilisée pour comparer les formes n'est pas sensible à la redondance des points, il n'y a pas strictement besoin de supprimer les positions des atomes redondantes.*

Le volume de l'environnement atomique est pré-calculé, puisqu'il est utilisé dans l'heuristique de comparaison de formes.

Longueur des arêtes Les distances entre les sommets du triangle fonctionnel du triplet sont enregistrées pour être utilisées lors de la comparaison.

Orientation par rapport à la molécule À chaque groupement chimique est associé un point appelé centre de masse local (section 3.2.1.2 page 52). Pour un triplet T, le centre de masse local est défini comme la moyenne C des centres de masse locaux des groupements chimiques de positions P_1, P_2 et P_3. Le produit mixte de $(\overrightarrow{CP_1}, \overrightarrow{CP_2}, \overrightarrow{CP_3})$ est enregistré au niveau du triplet et permet dans une certaine mesure de caractériser l'orientation du plan du triangle par rapport à la macromolécule.

3.2.3 Le graphe de triplets

La représentation des macromolécules utilisée pour leur comparaison consiste en un graphe dont les sommets sont des triplets de groupements chimiques et dont les arêtes relient les triplets vérifiant certains critères de proximité.

3.2.3.1 Sommets

Chaque sommet du graphe constitué d'un triplet de groupements chimiques. Selon le nombre de permutations associé au type du triplet, 1, 2 ou 6 sous-variantes du triplet seront générées.

3.2.3.2 Arêtes

Les arêtes connectent les triplets dont les triangles fonctionnels associés sont *quasi-adjacents*, c'est-à-dire que pour 2 triplets de positions (p_1, p_2, p_3) et (q_1, q_2, q_3) il doit exister exactement 2 paires d'indices (i_p, i_q) et (j_p, j_q) telles que :

$$\begin{cases} i_p \neq j_p \\ i_q \neq j_q \\ \left\|p_{i_p} - q_{i_q}\right\| \leq d_{\max} \\ \left\|p_{j_p} - q_{j_q}\right\| \leq d_{\max} \end{cases}$$

où $d_{\max}$ est une distance constante et généralement petite devant la longueur des arêtes des triangles. Les triangles (p_1, p_2, p_3) et (q_1, q_2, q_3) alors sont dits quasi-adjacents au niveau de (p_{i_p}, q_{i_q}) et (p_{j_p}, q_{j_q}). Cette propriété est illustrée figure 3.9 page 63. Notons que deux triangles adjacents seront nécessairement quasi-adjacents.

Dans SuMo version 4.4, la valeur de $d_{\max}$ est de 1 Å.

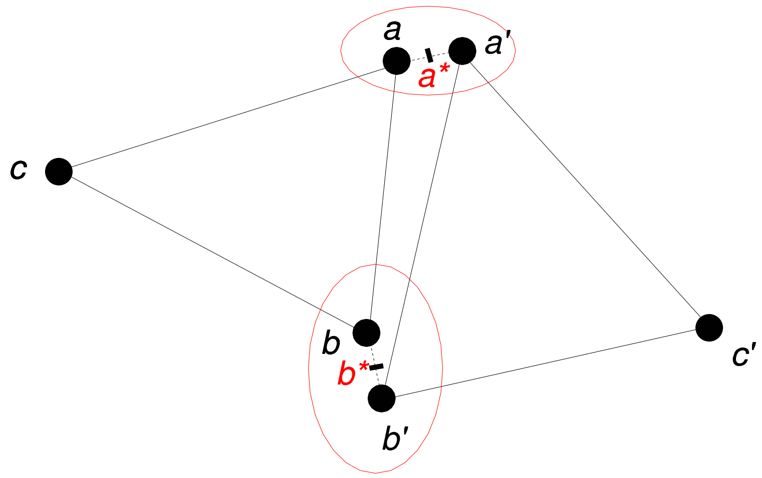


FIG. 3.9 – Exemples de triangles (a, b, c) et (a', b', c') quasi-adjacents au niveau de (a, a') et (b, b').

Au niveau d'une arête est stocké l'angle formé entre les 2 triangles quasi-adjacents. Cet angle peut *a priori* être compris entre $-\pi$ et π modulo 2π. Il est défini à l'aide des paires de points responsables de la quasi-adjacence des triangles. Soit 2 triangles (a, b, c) et (a', b', c') quasi-adjacents au niveau des paires de points (a, a') et (b, b'). Les positions moyennes $a^\star$ et $b^\star$ sont définies ainsi :

$$\begin{aligned} a^\star &= \frac{1}{2}(a + a') \\ b^\star &= \frac{1}{2}(b + b') \end{aligned}$$

Désignons par w le vecteur $\overrightarrow{a^\star b^\star}$. Celui-ci définit un plan orienté $\mathcal{P}$. Soit v_1 et v_2 les vecteurs définissant les plans orientés donnés respectivement par (a, b, c) et (a', b', c'). Appelons $\tilde{v_1}$ et $\tilde{v_2}$ leurs projetés selon w. L'angle θ entre les 2 triangles quasi-adjacents (a, b, c) et (a', b', d) est alors défini comme l'angle formé par les vecteurs $\tilde{v_1}$ et $\tilde{v_2}$ dans le plan orienté $\mathcal{P}$. Si l'ordre de a et b et de a' et b' est conservé comme dans (b, c, a) ou (c', a', b'), l'angle entre les triangles quasi-adjacents est le même. Par contre, s'il y a inversion soit dans l'un, soit dans l'autre des triplets de points comme dans (b, a, c), l'angle sera $\theta + \pi$.

> *La version 4.4 de SuMo n'impose pas de contraintes sur la valeur de cet angle pour qu'une arête du graphe soit créée contrairement à certaines versions antérieures.*

3.2.4 Stockage des données

Le temps nécessaire pour générer les graphes de triplets de groupements chimiques à partir d'une structure 3D de protéine est généralement de quelques secondes. Si l'on compte 5 secondes par structure et que l'on souhaite comparer un site 3D avec chacune des 20000 protéines de la PDB, le criblage nécessiterait un minimum de 24 heures. En pratique, ce type de criblage prend moins d'une heure grâce au précalcul des graphes de triplets.

3.2.4.1 Format

Les graphes de triplets — accompagnés de quelques informations telles qu'une description sommaire tirées du fichier PDB d'origine ou des annotations de sites — sont stockés dans des fichiers sous une forme facilement lisible par le programme `sumo`.

La conversion des données en une chaîne de caractères stockée dans un fichier et la conversion inverse sont assurées par les fonctions du module Marshal d'Objective Caml (section 2.3.3 page 33).

3.2.4.2 Compression

En raison du volume important des données stockées, il peut être intéressant de les compresser à partir du moment où le temps nécessaire à la décompression reste modéré devant le temps mis pour lire les données depuis le fichier et la comparaison. La compression permet en outre de gagner du temps lorsque les données doivent transiter à travers un réseau, via un système de fichiers distants tel que NFS.

L'utilisation directe d'un utilitaire de compression de fichiers tel que `gzip` ne permet pas d'obtenir un taux de compression satisfaisant des fichiers au format Marshal. En effet, une grande partie des données est constituée de nombres flottants double précision, sur 64 bits correspondant au standard IEEE 754. Sur ces 64 bits, 51 sont utilisés pour la mantisse, ce qui permet d'obtenir une précision relative de 2^{-52} c'est-à-dire environ 2.10^{-16}. Or les coordonnées données dans les fichiers PDB sont exprimées avec au plus 7 chiffres décimaux, soit une précision relative de 5.10^{-8} dans le meilleur des cas. Cette précision est égale à $2^{-24,3}$. Puisque 24 bits permettent d'atteindre une précision de 2^{-25}, les nombres flottants utilisés dans le graphe de triplets peuvent être arrondis de façon à n'utiliser que les 24 premiers bits. Ainsi, les 27 autres bits sont positionnés à 0. Ceci conduit à l'obtention de suites d'au moins 27 zéros consécutifs au niveau de la représentation des nombres flottants utilisée par Marshal. Ces données sont alors aisément compressées par `gzip`, avec un gain de place d'environ 45% par rapport aux données compressées sans arrondi préalable.

> *Ce système de compression est sensible aux grandes translations de coordonnées. Il n'est utilisable tel quel uniquement pour un maximum de 7 chiffres significatifs. C'est le cas dans tous les fichiers PDB en raison de la syntaxe imposée, mais cela peut devenir faux avec d'autres formats de fichiers. Par exemple 1000003,27 serait arrondi à 1000003.*

Par exemple, la taille occupée par un tableau de 10000 nombres flottants choisis aléatoirement entre 0 et 10 est indiquée dans le tableau suivant :

Taille	Sans compression		Compression par `gzip`	
	Sans arrondi	Arrondi	Sans arrondi	Arrondi
Octets	80025	80025	76270	42134
Proportion	100%	100%	95%	53%

3.2.5 Coeur de la comparaison

La comparaison des structures 3D de macromolécules consiste à identifier des régions semblables à partir des graphes de triplets de groupements chimiques.

3.2.5.1 Triplets similaires

La première étape de la comparaison des graphes de triplets est la recherche des paires de triplets similaires. Plusieurs critères sont utilisés successivement pour tester si une paire est acceptable ou non. Chaque critère non satisfait est éliminatoire. Afin d'optimiser les calculs, les critères sont testés dans l'ordre des paragraphes suivants.

Remarque préliminaire Si plusieurs permutations doivent être prises en compte parce que les triplets comparés possèdent des groupements de types identiques, alors chaque permutation du premier triplet doit être considérée face à une des permutations du second triplet. Par exemple, si l'on a à comparer les triplets

$$\left(x_1^{(\texttt{acyl})}, x_2^{(\texttt{acyl})}, x_3^{(\texttt{hydroxyl})}\right) \text{ et } \left(y_1^{(\texttt{acyl})}, y_2^{(\texttt{acyl})}, y_3^{(\texttt{hydroxyl})}\right)$$

alors il faudra tester également la possibilité de correspondance entre l'autre permutation du premier triplet

$$\left(x_2^{(\texttt{acyl})}, x_1^{(\texttt{acyl})}, x_3^{(\texttt{hydroxyl})}\right) \text{ et } \left(y_1^{(\texttt{acyl})}, y_2^{(\texttt{acyl})}, y_3^{(\texttt{hydroxyl})}\right)$$

Néanmoins, les étapes ne nécessitant pas d'identification des sommets de triangles équivalents n'ont pas à considérer les multiples permutations possibles.

Triplets de même type Seuls des triplets de même type peuvent être considérés comme similaires. Rappelons que les types des triplets sont ordonnés selon le type des groupements chimiques.

Considérons qu'il y a environ 1000 types de triplets possibles, et qu'ils sont aussi fréquents les uns que les autres dans chaque structure de macromolécule. Le nombre de comparaisons de triplets à effectuer pour comparer deux structures de taille n_1 et n_2 naïvement est $n_1.n_2$. Un gain de temps peut être réalisé si de part et d'autre les triplets sont classés par type puis que seuls les triplets de même type sont comparés. Le nombre de comparaisons alors effectuées est alors de l'ordre de $1000 \cdot \frac{n_1}{1000} \cdot \frac{n_1}{1000}$ lorsque n_1 et n_2 sont grands, soit $\frac{n_1.n_2}{1000}$ au lieu de $n_1.n_2$.

Longueur des arêtes Pour que deux triplets soient considérés comme similaires, les arêtes des triangles doivent être de longueur voisine au sens où la déformation relative de chaque arête est inférieure à un seuil. Pour toute paire d'arêtes à comparer de longueurs d_1 et d_2, la *déformation relative* est définie par la fonction δ suivante :

$$\delta(d_1, d_2) = \frac{|d_1 - d_2|}{\frac{1}{2}(d_1 + d_2)}$$

> *Cette fonction δ est la même que celle utilisée dans la définition générale de la déformation relative (définition 3.12 page 92), en prenant implicitement la moyenne des 2 distances comparées comme distance de référence.*

La déformation relative maximale tolérée est de 0,25 dans la version 4.4 de SuMo.

Disposition du plan Le produit mixte précalculé qui permet de donner une notion de la disposition du plan du triangle par rapport à l'ensemble de la structure est comparé entre les 2 triplets. La différence maximale acceptée dans SuMo 4.4 est de 50 Å^3.

Enfouissement L'enfouissement évalué par la densité atomique est comparé. La différence maximale d'enfouissement tolérée dans la version 4.4 est de 0,08 g.mol^{-1}.Å^{-3}.

Orientation des groupements chimiques L'orientation des groupements chimiques est donnée par leurs variants géométriques. Nous avons vus que les points et les vecteurs constituant les variants géométriques avaient été préalablement exprimés par rapport au repère du triplet. Ainsi, la superposition des triangles est déjà effectuée.

Parmi les variants géométriques proposés par SuMo actuellement, seuls des vecteurs unitaires définissent l'orientation des groupements chimiques. Ces vecteurs, exprimés dans le repère du triplet auxquels ils appartiennent sont prêts à être comparés. L'angle entre les vecteurs doit être inférieur à un certain seuil. Chaque angle-seuil est donné au niveau de la définition du type de groupement chimique. La définition complète des types de groupements chimiques est donnée en annexe page 153.

Comparaison de forme locale Cette opération consiste à comparer la forme de l'environnement à partir des positions des différents atomes environnants dont les coordonnées sont exprimées dans le repère du triplet auxquels ils appartiennent. L'heuristique de comparaison de forme qui a été mise au point est décrite en détail section 3.6.2 page 83. C'est l'opération la plus coûteuse mise en jeu pour dans la comparaison des triplets, elle intervient donc seulement si toutes les autres conditions sont vérifiées.

De la même façon que pour les autres étapes basées sur l'identification des sommets des triangles, toutes les permutations de l'un ou de l'autre des triplets doivent être considérées.

Le score minimal pour considérer que les deux environnements ont une forme similaire est de 0,65 sur une échelle qui va de 0 à 1.

3.2.5.2 Connexion des paires

L'étape suivante de la comparaison est la formation de ce que nous appellerons graphe de comparaison. Il s'agit de constituer un graphe dont les sommets sont les paires de triplets similaires identifiés lors de l'étape précédente. Deux conditions sont nécessaires et suffisantes pour connecter deux sommets du graphe de comparaison : le double voisinage et la conservation de l'angle entre les triplets.

Le double voisinage Les paires de triplets similaires (x_1, y_1) et (x_2, y_2) peuvent être connectées par une arête si x_1 et x_2 sont connectés par une arête dans le graphe de triplets auxquels ils appartiennent, et qu'il en soit de même pour leurs équivalents y_1 et y_2.

> *x_1 et x_2 ne sont pas forcément distincts. Il en va de même naturellement pour y_1 et y_2.*

L'angle entre les triplets Soit deux paires de triplets similaires (x_1, y_1) et (x_2, y_2) vérifiant la condition précédente du double voisinage. Pour être connectées dans le graphe de comparaison, l'arête x_1x_2 du premier graphe de triplets et l'arête y_1y_2 de l'autre graphe doivent porter un angle similaire. Notons ces angles θ_1 et θ_2. Rappelons qu'ils sont compris entre $-\pi$ et π. La différence $|\theta_1 - \theta_2|$ entre ces deux angles modulo 2π doit être inférieure à un seuil λ donné.

Dans la version 4.4 de SuMo, le seuil λ est fixé à 40°.

3.2.5.3 Isolement des sous-graphes indépendants

L'étape suivante de la comparaison consiste à extraire les sous-graphes indépendants du graphe de comparaison.

Obtention des sous-graphes indépendants Les sous-graphes indépendants du graphe de comparaison sont extraits en utilisant l'algorithme 1 page 48.

Nature du résultat obtenu Le résultat obtenu est donc un ensemble de sous-graphes indépendants. Chaque sous-graphe correspond à une liste de correspondances entre triplets de groupements chimiques. Les paires de triplets sont converties en paires de groupements chimiques : c'est à partir de ce point que la notion de triplet disparaît. Au final, les sous-graphes sont convertis en une liste de correspondances non-redondante entre groupements chimiques. Néanmoins, chaque groupement chimique peut éventuellement être impliqué dans plusieurs paires de la même liste de correspondances comme dans :

$$\{(g_1, g'_1), (g_2, g'_2), (g_3, g'_1), (g_4, g'_4)\}$$

3.2.5.4 Filtrage des résultats

Une étape de filtrage des résultats a lieu sur les listes de correspondance retournées par l'heuristique décrite précédemment. Cette étape permet satisfaire certaines conditions, qui ne seraient pas exactement satisfaites au niveau de la recherche de sous-structures similaires. Il peut par exemple s'agir de grandeurs calculées sur la globalité d'une liste de correspondances donnée et non-extensives, telles que le RMSD ou le volume tel que défini à la section 3.6.2.3 page 87.

Contrairement aux versions de SuMo 1 et 2, les versions 3 et 4 ne modifient pas les listes de correspondances de groupements chimiques une fois celles-ci obtenues à partir des listes de correspondances de triplets. Ainsi, certaines listes de correspondances seront conservées, et d'autres non, mais aucun sous-ensemble ne sera généré. En effet, SuMo considère comme naturelle l'heuristique de comparaison utilisée, basée un découpage en triplets de groupements chimiques et une tolérance restreinte mais locale au niveau de l'angle formé entre les triplets quasi-adjacents. Chaque triplet modélise une zone potentielle d'ancrage, sans degré de liberté, d'un fragment rigide de ligand flexible.

La notion de flexibilité fonctionnelle Nous appellerons *flexibilité fonctionnelle* la diversité des sites de fixation d'un même ligand qui utilisent les même types d'interactions. Les sites eux-mêmes ne sont pas forcément flexibles au point de fixer le ligand dans la même conformation.

Ainsi, la flexibilité fonctionnelle de certains sites de fixation de ligands implique la non-superposabilité de ces sites au niveau des points responsables de l'interaction avec le ligand. Pour cela, depuis la version 4 de SuMo, le RMSD n'est plus utilisé comme critère de qualité pour estimer la ressemblance de sites 3D.

La déformation au lieu de la superposition La notion de déformation vient remplacer celle de superposition. Le but de cette notion est de pénaliser les déformations locales tout en admettant que de petites déformations locales puissent contribuer à la non-superposabilité de sites conservant pourtant les mêmes propriétés fonctionnelles.

Une définition élaborée de la déformation et de son calcul est donnée section 3.6.3 page 89. Elle permet de prendre en compte non-seulement la déformation des distances entre objets élémentaires, mais également leurs changements d'orientation en tenant compte de leurs symétries éventuelles.

L'estimation globale de la déformation des listes de correspondances entre groupements chimiques permet de reprendre à plus grande échelle des critères pris en compte lors de la comparaison des triplets de groupements chimiques. Ces critères sont la déformation relative de la longueur des arêtes des triangles et la similarité de l'orientation des variants géométriques après superposition.

Le seuil acceptable après le calcul global de la déformation doit être plus strict que celui toléré au niveau des paires de triplets équivalents. Ceci permet de contraindre la déformation moyenne tout en autorisant quelques déformations localement plus importantes.

La déformation maximale tolérée est de 0,15 dans la version 4.4 de SuMo.

3.3 Comparaison multiple

Un système de comparaison multiple de structures ou de sous-structures 3D a été développé et implémenté dans SuMo. Néanmoins, il n'est pas lié spécifiquement à SuMo et pourrait être utilisé en aval de n'importe quel outil d'identification de listes de correspondances entre objets localisés dans l'espace de la même façon que les groupements chimiques.

3.3.1 Principe général

Le système de comparaison multiple qui a été mis au point n'est pas un système d'alignement structural. Il doit être vu comme un système de recherche de familles de sites partagées entre différentes structures 3D.

La première étape de la comparaison de n structures 3D consiste à obtenir des listes de correspondances entre groupements chimiques entre toutes ces protéines, en effectuant les $n.(n-1)/2$ comparaisons 2 à 2.

Ensuite, dans chaque structure, les sites suffisamment chevauchants à l'intérieur de chaque structure sont regroupés.

L'étape finale de la comparaison consiste à mettre en place les familles de sites correspondants entre les différentes structures.

3.3.2 Les étapes de la comparaison multiple

La figure 3.10 page 71 illustre les différentes étapes de la comparaison multiple, en représentant les différents types d'objets manipulés :

- structure 3D
- groupements chimiques
- correspondances entre groupements chimiques
- sites
- correspondance entre sites
- sites caractéristiques
- correspondance entre sites caractéristiques
- familles de sites caractéristiques

3.3.2.1 Obtention des correspondances entre groupements chimiques

La comparaison de chaque paire de structures 3D aboutit à des listes de correspondances. Chacune de ces listes correspond à une couleur donnée sur la figure 3.10. Chaque trait de la figure 3.10B indique une paire de groupements chimiques en correspondance.

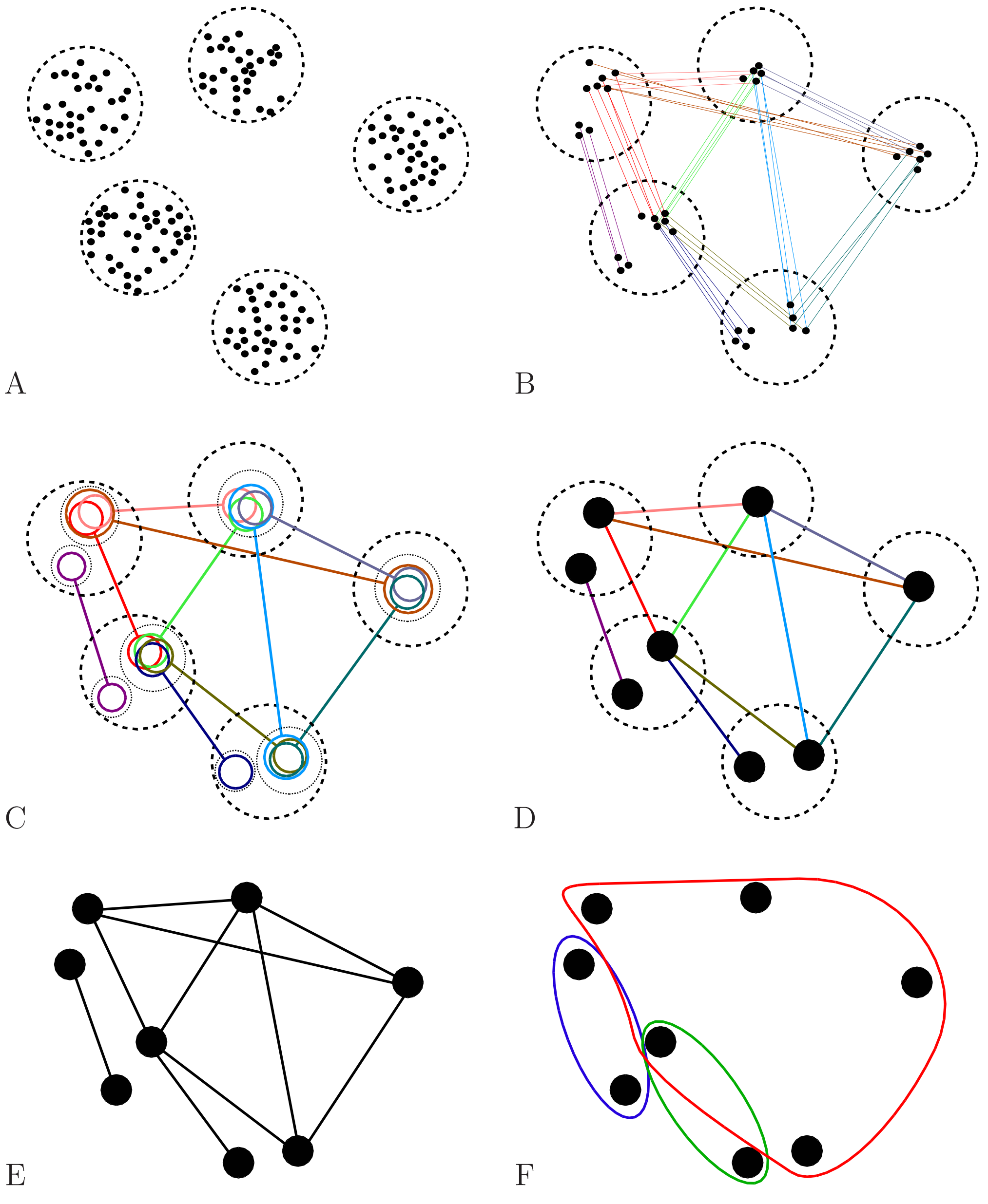


FIG. 3.10 – Étapes de la comparaison multiple de structures de macromolécules. **A** : 5 structures 3D représentées par des groupements chimiques. **B** : recherche des correspondances entre toutes les structures 3D 2 à 2. **C** : notion de site et de sites correspondants. **D** : regroupement des sites suffisamment chevauchants en sites caractéristiques. **E** : graphe de sites caractéristiques. **F** : 3 familles de sites caractéristiques.

3.3.2.2 Obtention de sites

Toute liste de correspondances L est associée à une paire de sites (A, B) définis par :

$$\begin{cases} A &= \{a|(a,b) \in L\} \\ B &= \{b|(a,b) \in L\} \end{cases}$$

Nous dirons alors qu'il existe une correspondance entre les sites A et B.

3.3.2.3 Regroupement des sites suffisamment chevauchants

Pour chaque structure 3D ont été identifiés un certain nombre de sites par comparaison avec les autres structures. Pour une structure donnée, les sites identifiés vont être regroupés lorsqu'ils sont considérés comme suffisamment chevauchants. Un site peut être pris en compte dans plusieurs regroupements. Un regroupement de sites est appelé *site caractéristique*.

Chevauchement entre 2 sites Nous décidons qu'il y a chevauchement entre deux sites A et B à partir d'un score de chevauchement ϕ :

$$\phi(A, B) = |A \cap B| - k. \min(|A|, |B|)$$

où k est une constante appelée *facteur de couverture* et dont la valeur est comprise entre 0 et 1. A et B sont considérés comme chevauchants lorsque $\phi(A, B) \geq 0$.

Le facteur de couverture indique le chevauchement minimal entre les deux ensembles exprimé par rapport au plus petit de ces ensembles. La valeur du facteur de couverture k est fixée à $\frac{2}{3} + \epsilon$ dans la version actuelle de SuMo, ce qui indique que strictement plus de deux tiers des groupements chimiques du plus petit des deux sites doivent également appartenir à l'autre site.

Chevauchement entre n sites Pour que n sites soient considérés comme chevauchants, il est nécessaire et suffisant que tous les sites soient chevauchants 2 à 2.

La recherche de tous les ensembles de sites chevauchants d'une structure 3D donnée revient à rechercher toutes les cliques du graphe dont :

- les sommets représentent les sites,
- les arêtes représentent les chevauchements entre les sites pris 2 à 2.

Seules les cliques maximales sont considérées pour former ce que nous appellerons des sites caractéristiques. Il s'agit de paquets de sites de taille maximale. Une définition de ce problème NP-complet classique est donnée page 97.

Sites caractéristiques Un site caractéristique vient d'être défini comme un regroupement de n sites. Chaque groupement chimique i appartenant à un site caractéristique est associé au nombre de sites m_i auxquels il appartient et qui ont été associés pour générer le site caractéristique. Le rapport m_i/n pourra être interprété comme une indication si le groupement chimique i est plutôt obligatoire pour une éventuelle fonction biologique s'il est proche de 1, et comme facultatif s'il est faible.

3.3.2.4 Correspondance entre sites caractéristiques

La correspondance entre sites caractéristiques issus de structures 3D différents est basée sur la correspondance entre les sites qui les composent, comme illustré sur les sous-figures 3.10C et 3.10D page 71. Il existe une correspondance entre deux sites caractéristiques $\mathcal{A}$ et $\mathcal{B}$ s'il existe au moins une correspondance entre un site appartenant à $\mathcal{A}$ et un site appartenant à $\mathcal{B}$.

3.3.2.5 Graphe de sites caractéristiques

Un graphe est construit, dans lequel les sommets représentent les sites caractéristiques et les arêtes les correspondances entre ces sites caractéristiques. Ce type de graphe est illustré par la sous-figure 3.10E.

3.3.2.6 Familles de sites caractéristiques

Des familles de sites caractéristiques sont extraites à partir du graphe précédent. Ces familles ont la propriété de pouvoir être chevauchantes. La recherche de familles est basée sur une recherche de sous-graphes presque complets nommés f-cliques et décrits en détail section 3.6.4 page 97. La fonction f utilisée dans SuMo version 4.4 est la suivante :

$$f : n \to \left\lfloor 0{,}24 \cdot \left(n - \frac{1}{2}\right)\right\rfloor$$

Le tableau suivant présente les premières valeurs prises par la fonction f :

n	$f(n)$
$1, 2, 3, 4$	0
$5, 6, 7, 8$	1
$9, 10, 11, 12$	2

Cette approche permet de générer des groupes de sites caractéristiques présentant des similitudes détectées par SuMo presque partout. Le résultat est illustré par la sous-figure 3.10F.

3.4 Bases de données

Au début de l'année 2003, la PDB comporte environ 20000 entrées, dont la plupart sont des structures comportant des protéines. La taille de cette base de données, telle que stockée au format officiel PDB et non compressée occupe un espace d'environ 11 gigagoctets (Go). Bien que les capacités de stockage proposées par les disques durs actuels permettent de stocker plusieurs fois des volumes de données équivalents sans investissement important, d'autres problèmes se posent.

Lors d'un criblage de base de données distante par un processus local, le transfert de l'intégralité de la base de données aura lieu au criblage. Deux types d'efforts peuvent être réalisés pour pallier à ce problème :

1. acquérir du matériel qui permette de stocker la base de données sur chaque machine cliente ou encore d'avoir un réseau à très haut débit entre le serveur de fichiers et le client
2. limiter la taille de la base de données en effectuant des compressions d'information

Une difficulté importante concerne la mise en place d'une base de données à partir de la PDB en vue d'un criblage par SuMo. Il s'agit d'éliminer les redondances à la fois structurales et fonctionnelles qui sont évidentes pour l'utilisateur, et conserver toutes celles qui ne le sont pas. En effet, il faut éviter dans la mesure du possible de diluer les résultats de criblage avec de nombreux résultats déjà connus et évidents, afin de faire ressortir ce qui ne l'est pas. Pour cela, quelques heuristiques de sélection ont été mises en place pour construire les bases de données SuMo.

Deux bases de données sont générées et utilisées par SuMo. La première est la base de données de structures-cibles potentielles. La seconde est la base de données de sites fonctionnels avérés, concernant actuellement les sites de fixation de ligands.

3.4.1 Structures-cibles potentielles

La base de données de structures-cibles potentielles n'est autre qu'une représentation intégrale de la PDB au format SuMo, après avoir effectué l'élimination de certaines redondances.

3.4.1.1 Élimination de redondances

Nous supposons que le nombre parfois important de structures disponibles pour des protéines de séquences et repliements très similaires voire identiques ne constitue pas une redondance, bien au contraire.

Ainsi, chaque structure de la PDB fait l'objet d'une entrée dans la base de données de structures-cibles potentielles. Néanmoins, la structure de certaines protéines est donnée sous forme de multimères. C'est à ce niveau-là que certaines informations vont être considérées comme redondantes et éliminées. Quelques précautions sont à prendre, elles font l'objet des paragraphes qui suivent.

Conserver l'environnement Par exemple si l'on a identifié qu'une structure de protéine était un homodimère et que l'on décide d'ignorer une des deux chaînes dans la représentation, il faut que les propriétés de l'environnement associées aux groupements chimiques et aux triplets soient les mêmes que si l'on considérait le dimère complet. Les paramètres concernés sont notamment :

- la densité atomique locale,
- l'orientation par rapport à la macromolécule,
- la position des atomes environnants,
- le caractère libre ou occupé des liaisons hydrogène.

Ces paramètres sont préservés par l'utilisation de l'option `-select` lors de la génération du graphe de triplets à partir des données structurales initiales. Par opposition, l'option `-restrict` permet de faire comme si la protéine était monomérique mais dans la même conformation. L'option `-restrict` ne sera donc pas utilisée ici et d'une manière générale doit être utilisée avec précaution. Ces options concernent la fonction `read` du langage SuMo, présenté section 3.7.1 page 101.

Conserver les zones à cheval Des sites fonctionnels peuvent être le résultat de la multimérisation de protéines et justement être situés à cheval sur plusieurs monomères. La solution qui a été adoptée consiste à sélectionner le monomère souhaité ainsi qu'une zone suffisamment large tout autour du monomère sélectionné. Cette zone correspond à un rayon de 6 Å dans la version 4.4 de SuMo. Ce rayon est utilisé pour la sélection à l'aide du mot-clé `around` dont la signification précise est donnée au niveau de la description du langage de sélection, section 3.7.1.3 page 106. Par exemple, pour ne sélectionner que le monomère A d'un homodimère AB, la sélection effectuée est `Pdb_chain "A" or 6.0 around (Pdb_chain "A")`.

3.4.1.2 Identification des chaînes redondantes

Actuellement, le seul critère d'élimination des chaînes redondantes est basée sur la séquence des polymères telle que donnée dans le fichier PDB. Un des problèmes que l'on rencontre est que certains monomères — en général des acides aminés — peuvent être non résolus et causer une interruption dans la séquence apparente du polymère. Pour éviter ce problème, une stricte identité entre les séquences extraites du fichier PDB n'est pas nécessaire pour considérer que deux chaînes sont équivalentes. Un autre problème des fichiers PDB est qu'une même notation de chaîne peut concerner plusieurs molécules réellement distinctes.

Pour considérer que deux chaînes sont équivalentes, leurs séquences sont déterminées d'après la numérotation de leurs monomères dans le fichier PDB. Les monomères qui ne font pas partie des 20 acides aminés classiques sont ignorés. Les séquences d'une longueur inférieure à 10 sont ignorées. Les 2 séquences s_1 et s_2 comparées doivent avoir au moins 90% d'identité au sens où au moins 90% des sous-séquences de longueur 4 de s_1 doivent exister dans s_2 et vice-versa.

Un programme indépendant a été réalisé pour connaître les différentes chaînes non négligeables d'un fichier PDB, et les équivalences entre ces chaînes selon les critères précédemment cités. Il s'agit de l'utilitaire `seqinfo`. Il est utilisé directement pour générer les chaînes à sélectionner pour limiter les redondances dans la base de données.

3.4.1.3 Taille finale de la base de données

La taille de la base de données des structures-cibles est environ 2 fois la taille de la PDB. Rappelons que cette base de données est compressée avec un taux d'environ 50% alors que la PDB ne l'est pas.

3.4.2 Sites de fixation de ligands

Une base de données de sites de fixation de ligands est générée et utilisée par SuMo.

3.4.2.1 Sélection

La sélection des sites de fixation de ligands est effectuée par identification des groupements chimiques dont au moins une des positions-cibles est située à une distance de moins de 4 Å d'un des atomes non-hydrogène du ligand. Ceci est implémenté au niveau de la construction `around` du langage de sélection des groupements chimiques décrit section 3.7.1.3 page 106. La définition des ligands est donnée section 3.2.1.1 page 49.

3.4.2.2 Élimination des redondances

L'élimination des redondances est basé sur le même principe que pour la base de données de structures-cibles. Par exemple, dans le cas d'un homodimère de protéines AB, la sélection du site de fixation de la molécule numérotée 3 se fera grâce au script de sélection suivant :

```
((4.0 around Molecule_index 3) and (not Molecule_index 3))
and
(Pdb_chain "A" or 6.0 around (Pdb_chain "A"))
```

3.4.2.3 Élimination des sites trop petits

Les sites trop petits pour être détectés comme significatifs dans le meilleur cas de figure avec SuMo ne sont pas enregistrés dans la base de données.

3.4.2.4 Enregistrement des types de triplets

La liste ordonnée des types de triplets constituant chaque site de fixation de ligand est également enregistrée. Elle est enregistrée dans un fichier indépendant.

Il s'agit d'une optimisation pour permettre la comparaison plus rapide face à d'autres sites de petite taille. En effet, si deux sites comparés n'ont pas un seul triplet du même type, alors il n'y a pas besoin de charger les données complètes pour effectuer la comparaison. Cette optimisation est utilisée pour la comparaison systématique de tous les sites entre entre eux, présentée section 3.5.4 page 80.

3.4.2.5 Taille de la base de données

La base de données de sites de fixation de ligands comporte environ 11000 sites, pour un ensemble de 1800 ligands, à partir de la PDB qui compte environ 20000 entrées à ce jour, et en utilisant la version 4.4 de SuMo.

Cette base de données occupe environ 400 mégaoctets (Mo) d'espace disque.

3.5 Annotation prédictive

Un système de prédiction automatisée de sites de fixation de ligands a été développé. Il est basé sur un post-traitement des résultats de criblage de la base de données de sites de fixation de ligands. Bien que les résultats bruts de criblage constituent déjà une prédiction, le système présenté ici permet d'en effectuer automatiquement une synthèse, sous forme d'annotation de sites fonctionnels potentiels présents sur la structure étudiée.

Ce système est basé sur deux notions essentielles, celle de famille de ligands et celle de spécificité de détection.

3.5.1 Familles de ligands

Soit B l'ensemble des ligands connus, en l'occurence ceux trouvés dans la PDB. Une *famille de ligands* est une bipartition (I, E) de B, où I définit les ligands internes à la famille et E les ligands externes à la famille.

> *Lors d'une mise à jour de la PDB s'accompagnant de la définition de nouveaux ligands, il convient donc de mettre à jour la définition de toutes les familles définies.*

Soit S la fonction qui à un ensemble de ligands associe l'ensemble de sites de fixation de ce ligand dans la base de données considérée. Une *famille de sites de fixation de ligands* est définie par un couple $(S(I), S(E))$ à partir d'une famille de ligands (I, E).

Les familles de ligands sont générées de façon arbitraire par l'administrateur local de SuMo et sont stockées manuellement au niveau du système de fichiers de la base de données. Par défaut, chaque identificateur de ligand l est utilisé pour générer une famille singleton $(\{l\}, B - \{l\})$.

3.5.2 Spécificité apparente

Soit une fonction f de recherche de correspondances entre un ensemble de groupements chimiques quelconques et un ensemble S de n sites de fixation de ligands :

$$f(X, S) = \{C_1, C_2, \ldots, C_i, \ldots, C_n\}$$

où X est l'ensemble de groupements chimiques que l'on utilise pour cribler la base de données de sites S et C_i désigne le résultat de la comparaison de X avec le site numéro i de S. C_i est un ensemble de listes de correspondances :

$$C_i = \{L_1, L_2, \ldots, L_{k_i}\}$$

Chaque résultat de comparaison C_i est alors converti en un seul site rassemblant tous les groupements chimiques apparaissant dans les listes de correspondances L_1 à L_{k_i}. Le sous-ensemble de X obtenu est un site noté M_i.

Nous pouvons maintenant définir la fonction m_S qui pour un groupement chimique g donné appartenant à X renvoie le nombre de fois où ce groupement chimique est présent dans les sites M_1 à M_n :

$$m_S(g) = \sum_{i=1}^{n} \begin{cases} 1 & \text{si } g \in M_i \\ 0 & \text{sinon} \end{cases} \tag{3.1}$$

La *sélectivité* du groupement chimique g pour l'ensemble de sites S est définie par la fonction ϕ_S suivante :

$$\begin{aligned} \phi_S(g) &= \frac{m_S(g)}{|S|} \\ &= \frac{m_S(g)}{n} \end{aligned} \tag{3.2}$$

La *spécificité apparente* d'un groupement chimique g appartenant à un ensemble X quelconque pour une famille de sites donnée par le couple $(S(I), S(E))$ est définie par la fonction ψ suivante :

$$\psi(g, I, E) = \frac{\phi_{S(I)}(g)}{\phi_{S(E)}(g)}$$

En pratique, la fonction f est l'heuristique de comparaison de SuMo. Il est important de noter que la spécificité au niveau d'un groupement chimique g dépend de X, dès le moment où X représente une sous-structure 3D et non une structure complète. En effet, si X est une sélection de groupements chimiques dont certains appartiennent à un site de fixation d'un ligand donné mais que ce site de fixation n'est pas totalement inclus dans X, alors il sera difficile de prédire une spécificité pertinente des groupements chimiques pour le ligand considéré.

Pour que la spécificité apparente puisse être interprétée avec fiabilité, il convient de l'associer au nombre de sites et de groupements chimiques considérés dans les calculs. Anisi nous considérons que $m_{S(I)}(g)$ doit être au moins égal à 10 pour que la spécificité apparente soit considérée comme utilisable pour le groupement chimique g et la famille (I, E) considérée.

3.5.3 Application : prédiction et annotation

Le criblage de la base de données complète de sites de fixation de ligands avec une structure d'intérêt X peut fournir une liste de résultats difficile à analyser en raison de sa longueur. Pour cela, pour chaque groupement chimique de X, la spécificité apparente pour chacune des familles de ligands est calculée dès lors qu'elle est considérée comme fiable au sens défini précédemment. Ceci garantit que SuMo détecte une similitude avec un site interne à la famille dans au moins 10 cas.

Une spécificité apparente proche de 1 indique que SuMo est très mauvais pour différencier les sites internes à une famille des sites externes à cette famille.

Une spécificité plus élevée, par exemple égale à 10, indique que le groupement chimique considéré est 10 fois plus souvent détecté comme appartenant à un site de fixation de ligand interne à la famille qu'un site de fixation externe à la famille.

La construction judicieuse de familles de ligands en fonction de la qualité de l'heuristique de comparaison et les similitudes de structure chimique de ces ligands peuvent permettre d'obtenir des spécificités plus élevées qu'en considérant tous les ligands comme différents et concurrents.

3.5.4 Application : auto-validation

Pour évaluer dans quelle mesure SuMo prédit correctement les sites fonctionnels, il est possible de le tester sur des sites fonctionnels connus.

Ainsi, chaque site de fixation de ligand de la base de données SuMo peut être utilisé pour être prédit par criblage de la base de données de sites. Pour chaque famille à laquelle il appartient, la spécificité apparente de chacun de ses groupements chimiques va être calculée, lorsque c'est possible.

3.5.4.1 Spécificité à l'échelle du site fonctionnel

Nous allons maintenant définir la spécificité apparente d'un site fonctionnel X. Ceci permet d'avoir une définition de la spécificité apparente non pas au niveau du groupement chimique mais au niveau du site fonctionnel que l'on considère. Nous définissons la fonction Φ_S suivante, par analogie avec la fonction sélectivité ϕ définie par l'expression 3.2 page 79 :

$$\Phi_S(X) = \frac{\sum_{g \in X} m_S(g)}{\sum_{g \in X} |S|}$$

où S désigne l'ensemble de sites criblés et m_S la fonction définie par l'expression 3.1 page 79. La spécificité apparente du site X pour la famille de ligands (I, E) est donnée la fonction Ψ suivante :

$$\Psi(X, I, E) = \frac{\Phi_I(X)}{\Phi_E(X)}$$

3.5.4.2 Calcul

Soit une base de données de sites de fixation de ligands qui contient n sites. Pour calculer $\Psi(X, I, E)$ pour tous les sites X de la base de données et pour toutes les familles (I, E) quel que soit leur nombre, il faut effectuer $n.(n-1)/2$ comparaisons de sites 2 à 2. Si l'on ne peut stocker les résultats de chaque comparaison pendant toute la durée du calcul, ce qui est le cas en pratique, il faut alors effectuer $n.(n-1)$ comparaisons, chaque comparaison étant effectuée deux fois.

Afin d'effectuer les comparaisons plus rapidement, une optimisation a été apportée : le chargement complet des données décrivant le deuxième site à comparer n'est effectué que si les 2 sites ont au moins un triplet de groupements chimiques de même type. Ceci nécessite le chargement d'un fichier beaucoup plus petit et plus simple. En pratique, ce système permet de ne pas effectuer environ trois comparaisons sur quatre, ce qui correspond à peu près à un gain équivalent en temps de calcul, soit un facteur 4.

Pour la version 4.4 de SuMo et les 11000 sites considérés en mars 2003, le temps nécessaire pour effectuer ce calcul sur une machine munie de deux processeurs de type Intel Pentium cadencés à 800MHz est de 10 jours. Si la taille des sites, liée au nombre de triplets les constituant, est multipliée d'un facteur k assez proche de 1, le temps de calcul augmente environ d'un facteur k^2.

3.6 Détail d'heuristiques

Lors de la mise en place d'une heuristique de bonne qualité, il est essentiel de ne pas séparer dans le temps et dans l'espace les différentes phases de conception que sont :

1. la formulation du problème biologique,
2. la modélisation du problème,
3. l'écriture d'un algorithme.

Les critères de qualité pour chacune de ces étapes sont (1) la pertinence de l'analyse ou de la prédiction apportée, (2) la qualité de l'approximation induite par la modélisation et (3) la durée du calcul permettant de traiter une instance du problème. Chacun de ces critères est essentiel : une solution doit apporter des informations utiles au biologiste et cette solution doit pouvoir être obtenue après une durée raisonnable. La façon dont est modélisé le problème va déterminer la réutilisabilité des approches adoptées. Travailler sur un système très simplifié va dans certains cas permettre des calculs très rapides et en utilisant des programmes préexistants mais sans pour autant répondre convenablement au problème biologique. Travailler sur un modèle complexe va rendre très difficile une réutilisation ultérieure et demander beaucoup d'efforts concernant la mise au point d'algorithmes.

Une très forte communication entre les personnes chargées des différentes étapes est donc nécessaire. En pratique, il est beaucoup plus efficace qu'une seule et même personne soit chargée des trois étapes plutôt que de faire travailler trois spécialistes — typiquement un biologiste, un mathématicien et un informaticien — sur chacune des étapes de conception.

Dans cette section sont présentées différentes heuristiques mises au point initialement dans le cadre de la comparaison de structures de macromolécules biologiques. Néanmoins, les problèmes posés et les solutions apportées sont indépendants de la notion de molécule. Les problèmes ayant trait à la disposition d'atomes en 3 dimensions sont la plupart du temps généralisables à des points dans un espace euclidien de dimension quelconque.

3.6.1 Fonction de densité atomique

Une *densité* correspond à une quantité de matière dans un volume donné. En milieu continu, une notion locale de la densité peut être définie comme la limite du rapport de la quantité de matière par le volume autour du point considéré lorsque ce volume tend vers 0, c'est-à-dire $\frac{dm}{dx.dy.dz}$. Néanmoins, en milieu discret tel qu'une macromolécule modélisée par des atomes ponctuels, cette définition n'est pas adéquate. L'idée qui est retenue dans ce cas est un intermédiaire entre une estimation globale de la densité et une estimation ponctuelle. Le but est d'effectuer une interpolation et un lissage de façon à obtenir une cartographie continue à partir d'une cartographie discrète, c'est-à-dire d'un ensemble de masses ponctuelles.

Une *densité locale* consiste à effectuer, pour chaque point P_i considéré une pondération de l'espace telle que le poids décroisse avec l'éloignement. Ainsi la densité locale est le rapport de la masse pondérée par le volume pondéré. Soit ϕ une fonction de pondération centrée :

$$\phi : \mathbf{R}^3 \to \mathbf{R}^+$$

Pour un point P_i de coordonnées (x_i, y_i, z_i), la fonction de pondération s'exprime sous la forme suivante :

$$w_i : (x, y, z) \mapsto \phi(x - x_i, y - y_i, z - z_i)$$

Il est important que ϕ soit sphérique, c'est-à-dire :

$$x^2 + y^2 + z^2 = x'^2 + y'^2 + z'^2 \Rightarrow \phi(x, y, z) = \phi(x', y', z')$$

De plus, une définition raisonnable de ϕ nécessite qu'elle soit décroissante avec l'éloignement de l'origine, c'est-à-dire :

$$x^2 + y^2 + z^2 < x'^2 + y'^2 + z'^2 \Rightarrow \phi(x, y, z) \geq \phi(x', y', z')$$

Si $\mathbf{P}$ est un ensemble discret de points de l'espace associés à une masse donnée par la fonction m, la densité associée à un point quelconque P_i se formule de la façon suivante :

$$\text{densité}(P_i) = \frac{\frac{\sum_{P_j \in \mathbf{P}} w_i(P_j).m(P_j)}{\sum_{P_j \in \mathbf{P}} w_i(P_j)}}{\int\int\int_{\mathbf{R}^3} w_i(x, y, z).dx.dy.dz} \tag{3.3}$$

Afin que les unités de masse et de distance utilisées gardent un sens physique, et qu'en milieu homogène la densité ne dépende pas de la fonction ϕ choisie, il est important que ϕ soit normalisée, c'est-à-dire :

$$\int\int\int_{\mathbf{R}^3} \phi(x, y, z).dx.dy.dz = 1$$

Afin de garantir la continuité de la densité, il est également nécessaire et suffisant que ϕ soit continue.

Seuls les critères précédemment énoncés ont été retenus pour la définition de ϕ. Le choix des autres critères a été guidé par la facilité de calcul. Ainsi, la définition utilisée utilise la fonction ϕ suivante :

$$\phi(x, y, z) = \begin{cases} 4\left(1 - \frac{r}{r_{\max}}\right) & \text{si } r \leq r_{\max} \\ 0 & \text{sinon} \end{cases} \tag{3.4}$$

où r désigne la norme de (x, y, z), c'est-à-dire $\sqrt{x^2 + y^2 + z^2}$.

L'avantage principal de cette fonction est que les points situés dans un rayon supérieur à $r_{\max}$ peuvent être ignorés dans les calculs. Ainsi, en particulier dans le cas des macromolécules, le coût du calcul de densité en un point ne dépend pas du nombre d'atomes du système. D'autres types de fonctions ayant cette même propriété auraient néanmoins pu être choisies. La figure 3.6.1 page 84 représente la densité calculée pour différentes valeurs de $r_{\max}$ dans un système composé de 6 points formant un hexagone régulier.

3.6.2 Comparaison de forme locale

3.6.2.1 Formulation du problème

Étant donné deux objets superposés, peut-on dire que leur forme est semblable ? Ce problème est rencontré lorsque l'on a identifié des paires de

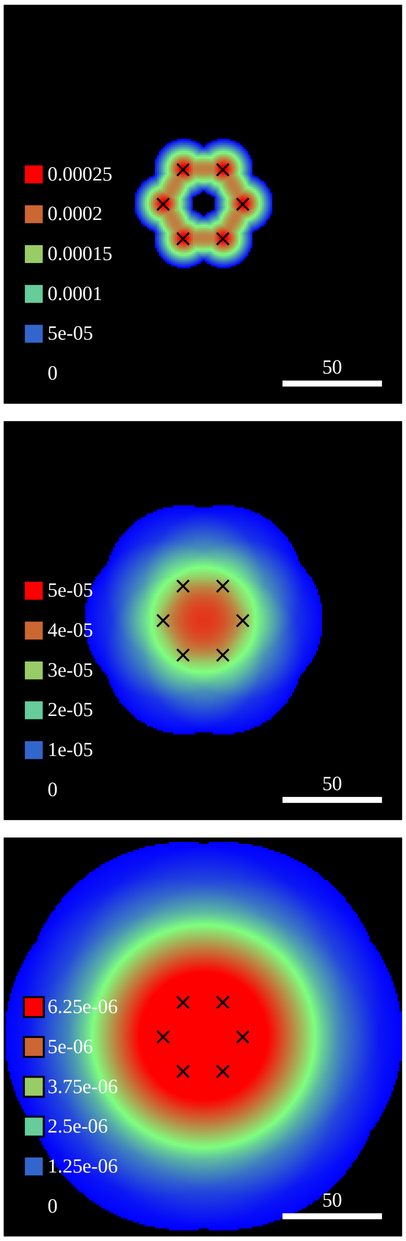


FIG. 3.11 – Densité d'un nuage de points calculée selon la fonction 3.3 page 83 utilisant la fonction de pondération 3.4 page 83 avec différentes valeurs de r_{max}, respectivement 15, 40 et 80. Les points sont matérialisés par les croix et appartiennent tous au plan de la projection. Les zones noires indiquent une densité nulle et sont en continuité avec les zones bleues.

groupements chimiques équivalents puis superposé les 2 molécules selon ces paires. Si l'on souhaite comparer l'encombrement de l'environnement autour des groupements chimiques qui nous intéressent, il faut mettre en place une heuristique de comparaison de forme qui bien entendu ne soit restreinte qu'à la région qui nous intéresse. Les données qui permettent de comparer les formes des molécules sont les positions des atomes, éventuellement complétées par une information sur leur rayon. Une molécule représentée par des atomes sphériques peut être vue comme l'union d'un ensemble fini de boules. La différence de forme de 2 sous-molécules M_1 et M_2 superposées peut être estimée par le volume de la non-intersection de M_1 et M_2, c'est-à-dire l'ensemble $(M_1 \cup M_2) - (M_1 \cap M_2)$ noté $\mathrm{excl}(M_1, M_2)$. Le calcul des volumes des ensembles M_1, M_2 et $\mathrm{excl}(M_1, M_2)$ n'est pas trivial d'une part, et d'autre part oblige à sélectionner les atomes que l'on considère comme appartenant dans l'environnement des points qui nous intéressent. Ceci aurait pour effet de rendre discontinue la fonction de comparaison de formes. Le problème est illustré en 2 dimensions par la figure 3.6.2.1 page 86.

Les critères pour une bonne heuristique de comparaison de formes locales sont les suivants :
- continuité de la fonction par rapport à la position des points du système ;
- temps de calcul en $O(n)$ où n est la taille de l'environnement considéré pour la comparaison de forme.

Nous appellerons un point tout objet dont un des attributs est un élément d'un ensemble euclidien. Par exemple 2 points ayant la même position mais ayant des identificateurs différents seront considérés comme distincts. Dans la suite nous appellerons volume toute fonction vérifiant pour tous ensembles de points A et B :

1. $\mathrm{volume}(A) \geq 0$
2. $\mathrm{volume}(A \cup B) \leq \mathrm{volume}(A) + \mathrm{volume}(B)$
3. Pour toute transformation f conservant toutes les distances entre les éléments de A on a :
   $\mathrm{volume}(A) = \mathrm{volume}(f(A))$
4. Pour tous points (p, q) on a :
   $\lim_{\|p-q\| \to 0} \mathrm{volume}(A \cup \{p\}) = \mathrm{volume}(A \cup \{q\})$
5. Pour tous points (p, q) on a :
   $\lim_{\|p-q\| \to +\infty} \mathrm{volume}(\{p, q\}) = \mathrm{volume}(\{p\}) + \mathrm{volume}(\{q\})$

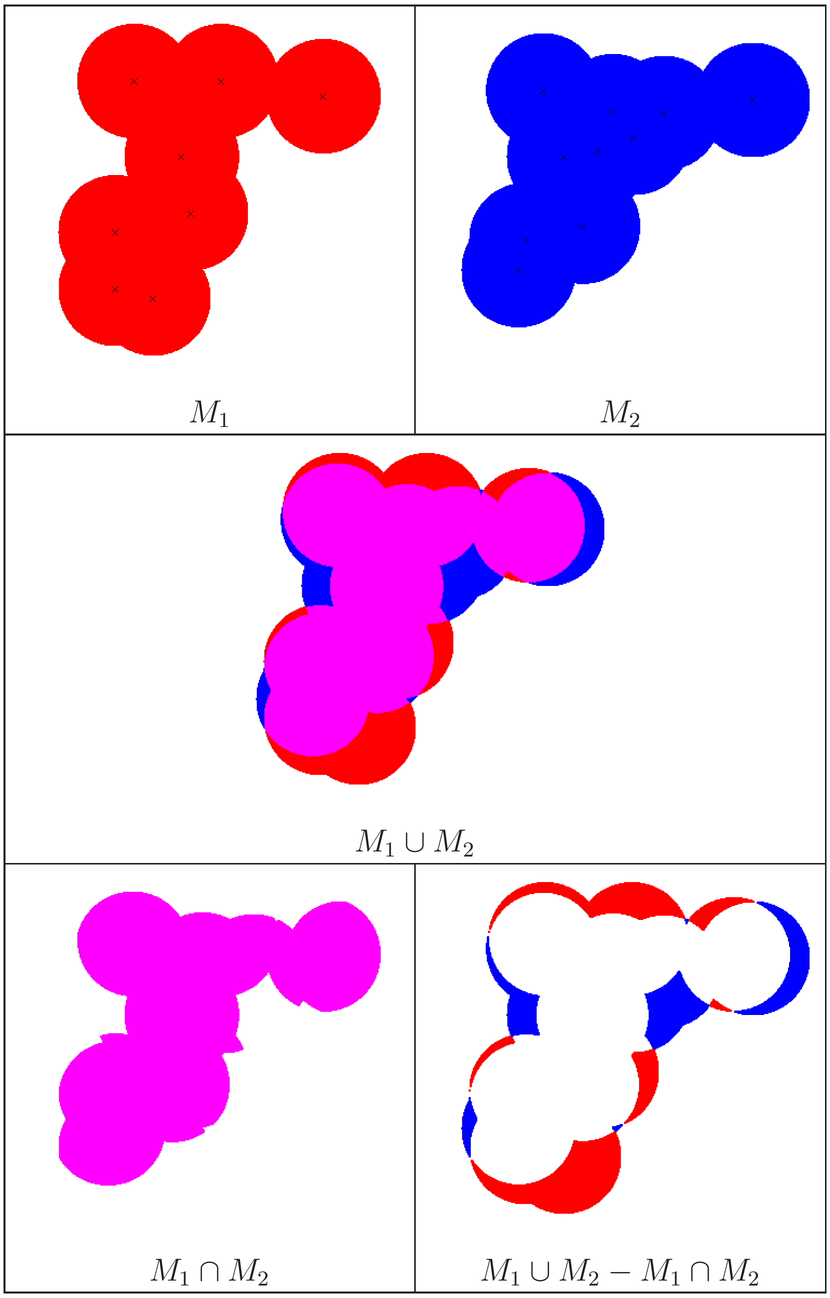


FIG. 3.12 – Union et intersection d'ensembles de sphères M_1 et M_2.

3.6.2.2 La fonction de score générale

Soit M_1 et M_2 deux ensembles de points que l'on a superposés et pour lesquels on aimerait savoir si leur forme est semblable. Notons $\text{comparison}_{\text{shape}}$ la fonction de comparaison de forme de 2 ensembles de points superposés. Il a été choisi de la définir ainsi :

$$\text{comparison}_{\text{shape}}(M_1, M_2) = \frac{\text{volume}(M_1) + \text{volume}(M_2) - \text{volume}(M_1 \cup M_2)}{\text{volume}(M_1 \cup M_2)} \tag{3.5}$$

La définition 3.5 est inspirée de l'expression du rapport de l'intersection par l'union de solides notés ici E_1 et E_2 :

$$\frac{\text{intersection}}{\text{union}} = \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|} = \frac{|E_1| + |E_2| - |E_1 \cup E_2|}{|E_1 \cup E_2|} \tag{3.6}$$

En effet, le volume de l'intersection de solides permet de quantifier la zone de recouvrement et celui-ci est maximal lorsque les 2 solides sont parfaitement superposés. Néanmoins, lorsque l'on travaille sur des ensembles discrets de points servant à modéliser des objets plus abstraits que des solides, l'intersection de ces points n'a pas le sens souhaité. Néanmoins, le volume de l'union de ces points possède bien les propriétés attendues, à savoir une union minimale lorsque les ensembles sont parfaitement superposés et une union maximale lorsque les distances entre les 2 ensembles tendent vers l'infini. Ainsi, lorsque M_1 et M_2 sont parfaitements superposés, le score vaut la valeur maximale de 1.

3.6.2.3 La fonction volume

A chaque point p_i considéré est associé une position et un poids noté w_i. La fonction volume choisie est la suivante :

$$\text{volume}(\{p_1, \ldots, p_n\}) = \sum_{i=1}^{n} m_i \tag{3.7}$$

où les m_i vérifient le système suivant :

$$\forall i \in [\![1, n]\!] \quad m_i = w_i \cdot \frac{m_i}{\sum_{j=1}^{n} f(d_{i,j}).m_j} \tag{3.8}$$

où $d_{i,j}$ est la distance entre les points p_i et p_j et f une fonction continue, décroissante, telle que $f(0) = 1$ et $\lim_{d \to +\infty} f(d) = 0$. Nous l'appellerons *fonction d'influence* puisqu'elle indique l'influence exercée par un point sur un autre, cette influence décroissant avec l'éloignement. La fonction d'influence qui a été choisie est la suivante :

$$f(d) = 2^{-\left(\frac{d}{\delta}\right)^s} \tag{3.9}$$

Cette fonction met en jeu 2 paramètres, δ et s. Pour la comparaison de forme de sites de macromolécules, les valeurs choisies sont les suivantes :
- $\delta = 2$Å
- $s = 2$

Le système 3.8 page 87 peut-être réécrit de la façon suivante :

$$\forall i \in [\![1, n]\!] \quad w_i = \sum_{j=1}^{n} f(d_{i,j}).m_j \tag{3.10}$$

Ainsi, le problème présenté sous cette forme consiste à résoudre un système linéaire de n équations à n inconnues notées m_j. Nous verrons cependant dans le paragraphe suivant que ce n'est pas cette approche qui a été retenue pour calculer les volumes.

3.6.2.4 Calcul du volume

Nous venons de voir que pour calculer le volume associé à n points pondérés tel que défini précédemment, il suffit de résoudre un système linéaire de n équations à n inconnues. Néanmoins, le coût de la résolution exacte d'un tel système s'exprime en $O(n^{2+c})$ où c est une constante strictement positive inférieure à un, variant selon les algorithmes connus et utilisés. La forme initiale du problème (équation 3.8 page 87) a été en réalité conçue de manière à permettre un calcul approché, par itérations successives.

La base de l'heuristique est la suivante : le point p_i subit l'influence de tous les autres points du système, et ce d'autant plus qu'ils sont proches de p_i. Cette influence se caractérise par une diminution de la masse m_i. Le poids w_i correspond à la *masse propre* du point p_i, c'est-à-dire la masse que ce point a lorsqu'il est isolé. Si 2 points p_i et p_j de poids respectifs égaux à 1 sont confondus et éloignés de tous les autres points du système, cela revient au même que s'il n'y avait qu'un seul de ces points à cette position ; chaque point aura donc une masse de 0,5. Ainsi, en initialisant chacune des masses m_i à leur valeur maximale w_i, on va pouvoir faire converger les valeurs des m_i vers les solutions du système en appliquant itérativement la procédure suivante, dérivée directement de l'expression 3.8 page 87 :

$$m_i^{(t+1)} \leftarrow w_i \cdot \frac{m_i^{(t)}}{\sum_{j=1}^{n} f(d_{i,j}).m_j^{(t)}} \tag{3.11}$$

En utilisant la fonction d'influence 3.9 page 88, de nombreux coefficients $f(d_{i,j})$ sont très faibles devant 1, qui est la valeur de chaque coefficient de la diagonale, c'est-à-dire $f(i,i)$. Pour le calcul, nous pouvons choisir de négliger les points situés à une distance de plus de 3δ, car dans ce cas leur influence réciproque est de :

$$\begin{aligned} f(3\delta) &= f(6\mathring{A}) \\ &\simeq 2.10^{-3} \end{aligned}$$

Ainsi, pour le calcul itératif de chaque w_i ne sont considérés que les points voisins de moins de 6 Å du point p_i.

Lorsque la densité des points est limitée, comme c'est le cas lorsqu'ils représentent des atomes ou des groupements d'atomes, le nombre de points voisins considérés lors du calcul d'un w_i est borné par une constante. La formule 3.11 page 88 telle qu'utilisée dans les calculs approchés n'est donc pas de taille $O(n)$ mais de taille bornée $O(1)$.

Le nombre de cycles à effectuer dépend de la précision souhaitée. Cette précision doit être en accord avec l'approximation effectuée lorsque l'on néglige les coefficients de petite taille. De plus il serait déraisonnable d'espérer une précision meilleure que les incertitudes expérimentales sur la position des atomes. En pratique, les 3 premiers chiffres significatifs obtenus sont toujours corrects lorsque l'on applique 10 cycles. Le calcul approché est donc tout-à-fait convenable étant donné la précision recherchée. Le temps mis pour calculer le volume associé à n points s'exprime en $O(n)$.

3.6.3 Estimation de déformation

3.6.3.1 Nature des problèmes

Considérons deux ensembles d'objets, $A = \{a_1, a_2, \ldots\}$ et $B = \{b_1, b_2, \ldots\}$. Ces objets sont identifiés au moins par leur position dans l'espace. Nous avons identifié une liste L de n paires d'objets que l'on considère comme équivalents. C'est une relation binaire entre A et B que nous appellerons *liste de correspondances*. L est de la forme :

$$L = \{(a_{i_1}, b_{j_1}), (a_{i_2}, b_{j_2}), \ldots, (a_{i_n}, b_{j_n})\}$$

où chaque élément a_{i_k} ou b_{j_l} peut être éventuellement présent dans plusieurs couples de la liste L.

Si chaque paire (a, b) de L représente une équivalence pour réaliser une propriété locale, alors un même élément a pourra être impliqué dans plusieurs propriétés locales différentes.

Problème 1 *Quelle est la qualité de la correspondance donnée par L ?*

Le problème 1 n'a pas de sens si la notion de qualité de la correspondance n'est pas définie au préalable. Par exemple, un critère de qualité utilisé fréquemment dans le domaine de la biologie structurale est le RMSD (voir définition 1 page 31). Dans le cas de l'identification de sites fonctionnels 3D dans les macromolécules, et en particulier pour les sites de fixation de ligands flexibles, nous avons considéré dans la section 3.2.5.4 page 69 qu'il semblait préférable d'adopter un critère qui tienne compte essentiellement des déformations locales et non des déformations globales comme c'est le cas avec le RMSD.

3.6.3.2 Décider de ce qui est local

La solution adoptée est donc basée sur une déformation de l'environnement local de chaque paire (a_{i_k}, b_{j_l}). La définition de ce qui est local est lié à une notion de distance critique, en-dessous de laquelle, les structures sont considérées comme rigides et au-dessus de laquelle elles sont considérées comme virtuellement flexibles. Par virtuellement flexible, nous entendons flexibilité fonctionnelle, telle que dans la définition 3.2.5.4 page 69.

Dans le cas des structures 3D de molécules, la distance critique sera située entre la taille des groupements chimiques rigides les plus gros et la distances fluctuantes les plus petites.

Nous considérons comme essentiels les doubles cycles aromatiques tels que ceux du tryptophane ou des bases puriques. D'autre part, la flexibilité des ligands courants commence à l'échelle des assemblages moléculaires A–B–C–D, où l'on observe une rotation autour de l'axe B–C, et donc une variation dans la distance entre les atomes A et D. Dans le cas d'une structure carbonée $C_{(A)}$–CH_2–CH_2–$C_{(D)}$, la distance entre A et D La distance fluctue couramment entre 3 Å et 3,9 Å. La taille d'un double cycle aromatique est de l'ordre de 5-6 Å. Il est donc raisonnable de choisir pour la distance critique un compromis situé dans la fourchette 3-6 Å.

3.6.3.3 Solution adoptée

Lorsque l'on souhaite définir une notion quantitative de la déformation locale en dimension 3, nous avons la possibilité de la baser sur la déformation de tétraèdres de petite taille, en considérant la variation des angles solides et des distances définissant ces tétraèdres. Ceci permet de différencier les stéréoisomères ou isomères optiques, c'est-à-dire les structures symétriques dans un miroir, contrairement à l'étude de la déformation de triangles ou de segments.

C'est pourtant la déformation de segments qui a été choisie comme base d'estimation de la déformation locale car :

– en pratique, dans SuMo, les isomères optiques sont éliminés avant l'estimation de déformation,
– l'implémentation est beaucoup plus aisée et les calculs plus légers.

Dans SuMo, c'est la prise en compte de l'angle orienté entre les triangles adjacents qui permet d'éviter les isomères optiques. Pour plus de détails, voir la section 3.2.5.2 page 68.

Cas des objets uniquement ponctuels Ici, nous définissons la déformation à partir d'objets uniquement ponctuels. Cette définition sera étendue dans les sections suivantes pour traiter des solides possédant éventuellement des symétries, comme c'est le cas des groupements chimiques tels que modélisés dans SuMo (section 3.2.1 page 46).

Le principe de l'estimation de déformation est le suivant : pour chaque paire d'objets équivalents (a_i, b_j) sont comptabilisées les variations de distances entre cette paire de points et toutes les autres paires. La variation de distances est appelée déviation, voici sa définition :

Définition 2 (Déviation entre 2 paires de points) *Soit deux paires de points $P = (a_i, b_j)$ et $Q = (a_k, b_l)$ appartenant à une liste de correspondances L. La déviation entre P et Q est la différence entre les distances euclidiennes a_ia_k et b_jb_k :*

$$\begin{aligned} \text{déviation}(P, Q) &= \Big| \, \|a_i - a_k\| - \|b_j - b_k\| \, \Big| \\ &= |a_ia_k - b_jb_k| \end{aligned}$$

Nous définissons également la distance entre 2 paires de points :

Définition 3 (Distance entre 2 paires de points) *Soit deux paires de points $P = (a_i, b_j)$ et $Q = (a_k, b_l)$ appartenant à une liste de correspondances L. La distance entre P et Q est définie comme la moyenne arithmétique des distances euclidiennes a_ia_k et b_jb_k :*

$$\begin{aligned} \text{distance}(P, Q) &= \frac{1}{2} \left(\|a_i - a_k\| + \|b_j - b_k\| \right) \\ &= \frac{1}{2} \left(a_ia_k + b_jb_k \right) \end{aligned}$$

L'estimation de déformation d'une liste de correspondances L fait intervenir deux facteurs essentiels :

1. Le *coefficient d'importance* est un coefficient pondérateur. Il donne l'importance de la variation de distance mesurée.
2. La *distance de référence* permet d'exprimer la déformation de façon relative et dépendante de la distance considérée.

Ces 2 facteurs sont décrits de façon détaillée par la suite. Ils ne dépendent que de la *distance entre les 2 paires de points*.

Coefficient d'importance Nous considérons que la déformation d'une distance a un impact fonctionnel d'autant plus important que la distance concernée est faible. Au-delà d'une certaine distance $d_{\max}$, l'importance de la conservation des distances est considérée comme nulle. La fonction suivante est utilisée :

$$\text{importance}(P, Q) = w_0 - \frac{w_0}{d_{\max}} \cdot \text{distance}(P, Q)$$

où w_0 est une constante. Dans l'implémentation, $d_{\max}$ joue le rôle de *cut-off*. Nous pouvons par exemple choisir une valeur de 10 Å.

Déformation relative Nous considérons que la déformation doit être exprimée relativement par rapport à une distance caractéristique de référence, appelée également distance de référence. A ce niveau, la notion de distance critique d_c telle qu'évoquée section 3.6.3.2 page 90 est importante. En effet, pour les faibles distances, la distance de référence sera constante alors que pour les grandes distances, la distance de référence sera la distance elle-même. La fonction suivante a été choisie :

$$d_{\text{ref}}(P, Q) = \begin{cases} \text{distance}(P, Q) & \text{si distance}(P, Q) > d_c \\ d_c & \text{sinon} \end{cases}$$

La déformation de la distance $d(P, Q)$ est définie comme le rapport δ entre la déviation mesurée et la distance de référence :

$$\delta(P, Q) = \frac{\text{déviation}(P, Q)}{d_{\text{ref}}(\text{distance}(P, Q))} \tag{3.12}$$

δ peut être considérée comme une déviation relative à une distance caractéristique de référence.

L'expression en deux phases de la fonction d_{ref} permet de pénaliser de façon équivalente toutes les variations des petites distances, et de considérer des déformations réellement relatives pour les variations des grandes distances. Un exemple raisonnable de valeur pour d_c est de 3 Å.

Formulation générale La déformation d'une liste de correspondances $L = \{(a_{i_1}, b_{j_1}), (a_{i_2}, b_{j_2}), \ldots, (a_{i_n}, b_{j_n})\}$ est définie par la fonction ci-dessous :

$$\text{déformation}(L) = \frac{\sum_{P \in L} \sum_{Q \in L-\{P\}} \text{importance}(P,Q).\delta(P,Q)}{\sum_{P \in L} \sum_{Q \in L-\{P\}} \text{importance}(P,Q)}$$

Il s'agit donc d'une moyenne des déviations rapportées aux distances caractéristiques de référence.

Extension aux objets non ponctuels sans symétrie

Intérêt de l'extension La déformation telle que définie précédemment permet de donner une idée de la distance entre 2 ensembles de points extraits d'une liste de correspondances. Ici, nous allons étendre notre définition de la déformation afin qu'elle puisse prendre en compte non seulement les translations mais également les rotations des objets élémentaires lorsque ceux-ci ne sont pas uniquement définis par des points mais comportent également une information d'orientation.

Par exemple, si un géologue souhaite quantifier le changement de position des pierres d'un éboulis entre deux dates données, il peut être judicieux pour lui de modéliser les pierres non pas seulement par des points mais aussi prendre en compte l'orientation de ces pierres. La prise en compte du changement d'orientation va le renseigner sur le mode de déplacement de ces pierres, plutôt par glissement ou plutôt par roulement.

Définition des objets manipulés Dans un premier temps, nous considérons des objets sans symétrie interne, modélisés par ce que nous appelons des pseudo-solides finis, tels que définis ci-dessous :

Définition 4 (Pseudo-Solide) *Nous appelons pseudo-solide la modélisation d'un objet par un couple (p, S) où p est un point représentant sa position et S est un ensemble de points quelconques. p est appelé position du pseudo-solide et les points de S sont appelés sommets du pseudo-solide.*

Définition 5 (Pseudo-Solide fini) *Nous appelons pseudo-solide fini tout pseudo-solide (p, S) ayant un nombre de sommets $|S|$ fini.*

Redéfinition de la déformation Soit L une liste de correspondances entre des pseudo-solides finis. 2 pseudo-solides en correspondance ont nécessairement le même nombre de sommets. Le calcul des facteurs d'importance et de la distance caractéristique de référence utilisent la distance entre les positions, comme dans la définition simple. Par contre, au lieu de considérer les déviations des distances entre toutes les positions des pseudo-solides, nous considérons les déviations des distances entre chaque position de pseudo-solide et chaque sommet appartenant à un autre pseudo-solide.

Adaptation aux objets symétriques

Problématique Lorsque les objets manipulés sont modélisés avec une symétrie, c'est-à-dire que certaines orientations de ces objets sont équivalentes d'un point de vue fonctionnel, il est important de les prendre en compte.

Par exemple, une fonction acide carboxylique d'un aspartate possède 2 atomes d'oxygène *a priori* équivalents, notés `OG1` et `OG2`. Nous pouvons modéliser cet objet par la position moyenne des oxygènes, et 2 sommets additionnels positionnés au niveau des oxygènes. Il faut donc trouver une solution qui considère que les atomes `OG1` et `OG2` sont fonctionnellement interchangeables.

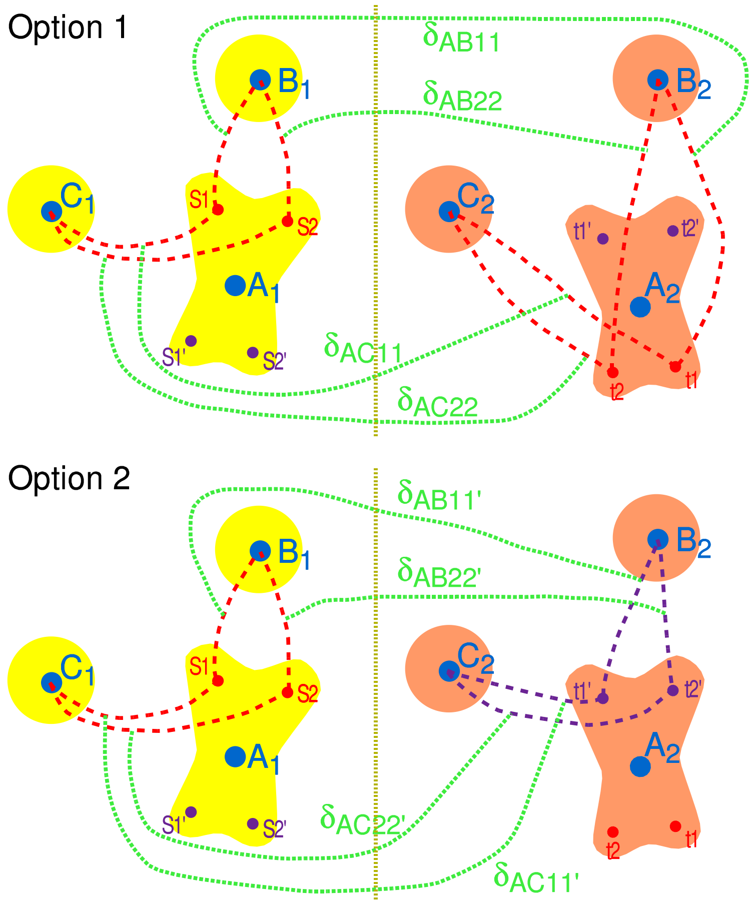
Pseudo-solides symétriques

Définition 6 (Pseudo-solide symétrique) *Un pseudo-solide symétrique est un couple* $(p, \{S_1, \ldots, S_k\})$ *où* p *est un point appelé position, et* S_1 *à* S_k *sont des ensembles distincts de points, de même cardinalité, et appelés variants symétriques. Les éléments des variants symétriques sont appelés sommets.*

Les différents variants symétriques d'un pseudo-solide permettent de modéliser les transformations de l'objet qui sont fonctionnellement invariantes. Il n'est pas nécessaire que ces transformations soient isométriques.

Extension de la définition de déformation Soit L une liste de correspondances entre pseudo-solides symétriques. Dans chaque paire de pseudo-solides correspondants, le nombre de variants symétriques peut être différent, mais ces variants doivent comporter le même nombre de sommets.

La figure 3.13 page 95 illustre la définition finale de la déformation que nous adoptons. Dans cet exemple, nous avons une correspondance entre 3 objets, où un de ces objets (position (A_1, A_2)) est modélisé par deux variants symétriques composés de 2 points et les 2 autres objets sont modélisés par un seul point sans symétrique qui correspond à la position-même de l'objet.



$$L = \{ \quad \begin{array}{l} \left((A_1, \{\{s_1, s_2\}, \{s_1', s_2'\}\}), (A_2, \{\{t_1, t_2\}, \{t_1', t_2'\}\}) \right), \\ \left((B_1, \{\{B_1\}\}), (B_2, \{\{B_2\}\}) \right), \\ \left((C_1, \{\{C_1\}\}), (C_2, \{\{C_2\}\}) \right) \quad \} \end{array}$$

$$\delta_{AB11} = \frac{|s_1B_1 - t_1B_2|}{d_{\text{ref}}\left(\frac{1}{2}(A_1B_1 + A_2B_2)\right)}$$

$$w_A.\delta_A = \min \left\{ \begin{array}{ll} w_{AB}.\delta_{AB11} + w_{AB}.\delta_{AB22} + w_{AC}.\delta_{AC11} + w_{AC}.\delta_{AC22} & \text{Option 1} \\ w_{AB}.\delta_{AB11'} + w_{AB}.\delta_{AB22'} + w_{AC}.\delta_{AC11'} + w_{AC}.\delta_{AC22'} & \text{Option 2} \\ w_{AB}.\delta_{AB1'1} + w_{AB}.\delta_{AB2'2} + w_{AC}.\delta_{AC1'1} + w_{AC}.\delta_{AC2'2} & \text{Option 3} \\ w_{AB}.\delta_{AB1'1'} + w_{AB}.\delta_{AB2'2'} + w_{AC}.\delta_{AC1'1'} + w_{AC}.\delta_{AC2'2'} & \text{Option 4} \end{array} \right.$$

$$\text{déformation}(L) = \frac{w_A.\delta_A + w_B.\delta_B + w_C.\delta_C}{w_A + w_B + w_C}$$

FIG. 3.13 – Estimation de la déformation à partir d'une liste de correspondances L entre pseudo-solides finis symétriques. 2 options sur les 4 possibles sont présentées. L'option choisie pour la déformation locale δ_A est celle qui minimise la déformation. Ici, l'option 2 sera probablement préférée à l'option 1.

La définition proposée est basée sur le choix de la meilleure paire de variants symétriques entre 2 objets se correspondant. Nous sommes amenés à définir la *déformation locale*.

Définition 7 (Déformation locale) *Soit une paire d'objets* $P_i = (X, Y)$ *en correspondance modélisés par des pseudo-solides finis symétriques, pris dans une liste de correspondances* $L = \{P_1, \ldots, P_i, \ldots P_n\}$. *Nous appellerons* A_1 *et* A_2 *leurs positions, et dénoterons par les lettres indexées* S *et* T *leurs variants symétriques :*

$$\begin{aligned} X &= (A_1, \{S_1, S_2, \ldots, S_{k_1}\}) \\ Y &= (A_2, \{T_1, T_2, \ldots, T_{k_2}\}) \end{aligned}$$

Notons r *le nombre de points des variants symétriques de* X *et de* Y, *c'est-à-dire* $|S_1|$. *La déformation locale en* (X, Y) *est la plus faible somme des déviations entre une paire de variants symétriques* (S_l, T_m) *et les positions de tous les autres objets du système :*

$$\begin{aligned} &\text{déformation locale}(L, (X, Y)) \cdot \text{poids local}(L, (X, Y)) = \\ &\min_{(1 \le l \le k_1, 1 \le m \le k_2)} \left(\textstyle\sum_{1 \le a \le r} \sum_{((p,V),(q,W)) \in L - \{P_i\}} w(\alpha).\delta(\beta) \right) \end{aligned}$$

où w *est la fonction importance qui donne le coefficient d'importance tel que défini page 92 ;* α *représente le couple* $((A_1, A_2), (p, q))$ *;* β *représente le couple* $((S_l[a], T_m[a]), (p, q))$. *Le poids local est la somme des coefficients d'importance utilisés dans la somme, c'est-à-dire :*

$$\begin{aligned} \text{poids local}(L, (X, Y)) &= \sum_{1 \le a \le r} \sum_{((p,V),(q,W)) \in L - \{P_i\}} w((A_1, A_2), (p, q)) \\ &= r. \left(\sum_{((p,V),(q,W)) \in L - \{P_i\}} w((A_1, A_2), (p, q)) \right) \end{aligned}$$

La déformation globale est simplement la moyenne pondérée des déformations locales :

Définition 8 *(Déformation) Pour une liste de correspondances* L, *la déformation est définie de la façon suivante :*

$$\text{déformation}(L) = \frac{\sum_{P \in L} \text{poids local}(L, P) \cdot \text{déformation locale}(L, P)}{\sum_{P \in L} \text{poids local}(L, P)}$$

Variante La moyenne arithmétique de termes fractionnaires, telles que les déformations relatives élémentaires données par la fonction δ, ne permet pas d'obtenir une déformation globale qui s'exprime sous la forme d'une somme de déviations divisée par une somme de distances de références. Dans ce cas on peut préferer conserver séparément les termes en dénominateur et en numérateur, effectuer les moyennes arithmétiques sur ces ensembles, et seulement à la fin diviser le dénominateur moyen par le numérateur moyen. Ainsi, nous allons remplacer la moyenne arithmétique des termes d_i/n_i de coefficients w_i par :

$$\frac{\sum_i w_i d_i}{\sum_i w_i n_i} \qquad \text{(moyenne adoptée)}$$

au lieu de :

$$\frac{\sum_i w_i \frac{d_i}{n_i}}{\sum_i w_i} \qquad \text{(moyenne arithmétique)}$$

Ainsi, la moyenne elle-même s'exprime sous forme fractionnaire et la moyenne de moyennes revient à directement faire la moyenne des éléments d'origine.

C'est cette forme qui est utilisée actuellement pour faire la moyenne des déformations locales, en conservant les déformations élémentaires δ sous leur forme fractionnaire.

3.6.4 Cliques incomplètes

3.6.4.1 Définitions

Définition 9 *Soit un graphe $G = (V, E)$. Soit V' un sous-ensemble des sommets V du graphe G. V' est une clique de G si et seulement si chaque paire de sommets prise dans V' forme une arête, c'est-à-dire un élément de E.*

Définition 10 *Soit un graphe $G = (V, E)$ et f une fonction croissante de* $\mathbf{N}$ *dans* $\mathbf{N}$. *Soit V' un sous-ensemble de sommets de V. Si chaque élément de V' est au moins connecté à $|V'| - 1 - f(|V'|)$ autres sommets de V', alors nous dirons que V' est une f-clique du graphe G.*

Définition 11 *Nous appellerons f-clique stable toute f-clique dont les éléments peuvent être ordonnés sous la forme $(v_1, v_2, \ldots, v_n)$ de sorte que les sous-ensembles suivants soient tous des f-cliques :*

$$\begin{array}{l} \{v_1\} \\ \{v_1, v_2\} \\ \{v_1, v_2, v_3\} \\ \vdots \\ \{v_1, v_2, v_3, \ldots, v_n\} \end{array}$$

Remarquons qu'une clique correspond au cas particulier de f-clique (stable) lorsque f est la fonction constante nulle.

Définition 12 *Une f-clique stable maximale est une f-clique stable d'un graphe G telle qu'elle ne soit incluse dans aucune autre f-clique stable de G.*

Définition 13 *Une f-clique stable maximum est une f-clique stable de cardinalité maximale pour un graphe donné.*

Les deux définitions précédentes sont classiques pour les cliques et sont ici une généralisation aux f-cliques stables pour des valeurs de k quelconques. La figure 3.14 page 99 présente différents exemples de graphes à 5 sommets. Pour chacun de ces graphes sont indiqués les f-cliques maximales, les f-cliques stables maximales et les cliques maximales où f est la fonction suivante :

$$f : n \rightarrow \left\lfloor \frac{n-1}{2} \right\rfloor$$

Nous traiterons ici du problème de l'identification de toutes les cliques maximales d'un graphe donné, généralisé aux f-cliques stables. Le problème de la clique maximum est connu pour être NP-complet sur des graphes quelconques. Donner une liste exhaustive des cliques maximales est donc au moins aussi coûteux. De plus, ici nous nous intéressons au cas plus général des f-cliques. Aucune solution en temps polynomial n'est donc connue pour ce problème : les algorithmes développés seront donc à utiliser pour des graphes ayant une structure particulière ou des graphes de très petite taille.

3.6.4.2 Algorithmes

L'algorithme mis au point pour extraire toutes les f-cliques stables maximales d'un graphe fonctionne par ■ extension de graine ■. Les propriétés suivantes sont nécessaires pour que ce type d'algorithme soit valide. Elles découlent directement de la définition des f-cliques stables.

Propriété 1 *Soit un graphe G. Toutes les f-cliques stables de taille $n+1$ peuvent être obtenues à partir de l'ensemble des f-cliques stables de taille n, pour $n \geq 1$.*

Cette propriété vient du fait que toute f-clique stable A de taille $n+1$ peut s'écrire sous la forme $B \cup \{v\}$ où B est une f-clique stable de taille n.

Propriété 2 *Toutes les f-cliques stables de taille n ($n \geq 1$) peuvent être obtenues à partir de l'ensemble des f-cliques stables de taille 1, c'est-à-dire chacun des sommets.*

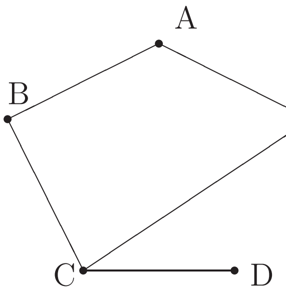
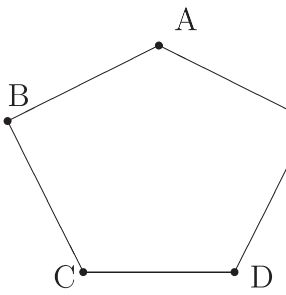
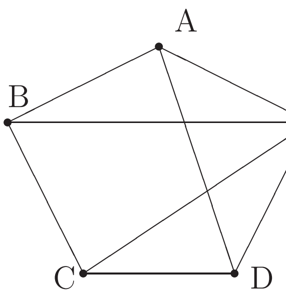
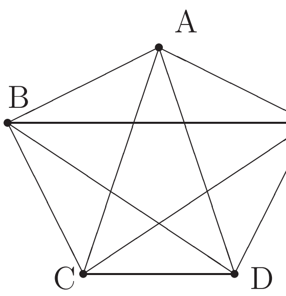
	f-cliques maximales	f-cliques stables maximales	cliques maximales
	ABCE, CD	ABC, ABE, ACE, BCD, BCE, CDE	AB, AE, BC, CD, CE
	ABCDE	ABC, BCD, CDE, DEA, EAB	AB, AE, BC, CD, DE
	ABCDE	ABCDE	ABE, ADE, BCE, CDE
	ABCDE	ABCDE	ABCDE

FIG. 3.14 – f-cliques maximales, f-cliques stables maximales et cliques maximales dans différents exemples de graphes. $f(n) = \left\lfloor \frac{n-1}{2} \right\rfloor$; autrement dit, chaque sommet doit être connecté à au moins 50 % des autres sommets de la f-clique considérée.

Il est donc envisageable de construire un algorithme qui va construire toutes les f-cliques stables de taille n à partir de l'ensemble des f-cliques stables de taille $n-1$. Chaque f-clique stable d'un graphe $G = (V, E)$ est contruite en $O(|V|)$ à partir d'une f-clique de taille inférieure. Si m est le nombre total de f-cliques stables dans le graphe G, le temps de calcul des f-cliques stables maximales par cet algorithme s'exprime en $O(m.|V|)$.

L'algorithme 2 précise les différentes étapes de l'extraction des f-cliques stables maximales. Les types utilisés pour manipuler les différents types d'ensembles rencontrés ne sont pas précisés.

Algorithme 2 Extraction des f-cliques maximales

```
Require : a graph G = (V, E)
V = {v1, v2, ..., vn}
Cliques ← {{v1}, {v2}, ..., {vn}}
Result ← ∅
while Cliques ≠ ∅ do
  Extensions ← ∅
  for all Clique ∈ Cliques
    for all Vertex ∈ Clique
      for all Neighbor ∈ Neighbors(Vertex)
        if Neighbor ∉ Clique then
          if (Clique, Neighbor) ∉ Extensions then
            Extensions ← {(Clique, Neighbor)} ∪ Extensions
  New-cliques ← ∅
  for all (Clique, Vertex) ∈ Extensions
    if Clique ∪ {Vertex} is still a f-clique then
      New-cliques ← {Clique ∪ {Vertex}} ∪ New-cliques
      remove Clique from Cliques
  Result ← Cliques ∪ Result
  Cliques ← New-cliques
```

On pourra par exemple représenter un ensemble de f-cliques stables par une table de hâchage dont les données sont une f-clique stables et dont les clés sont calculées à partir de la succession ordonnée des numéros des sommets qui la composent. Par exemple, l'ensemble des sommets numérotés 4, 23, et 6 donnera lieu à un tableau {4 ;6 ;23} à partir duquel la clé de hâchage sera calculée. La bibliothèque standard du langage Objective Caml fournit un module de manipulation de tables de hâchage [54]. Ce module, nommé Hashtbl, permet d'utiliser une fonction de hâchage (calcul de la clé) à partir de n'importe quel type de données et le redimensionnement automatique des tables de hâchage en fonction du remplissage. Pour des informations détaillées sur les tables de hâchage, le lecteur pourra consulter l'ouvrage [46].

3.7 Interfaces utilisateur

Les 3 niveaux de SuMo ont été présentés section 3.1.1 page 36. Le niveau du programmeur n'est pas un niveau d'utilisation. Le niveau intermédiaire par contre, même s'il n'est pas destiné à l'utilisateur moyen, est essentiel pour servir de triple interface entre le programme `sumo`, le système d'exploitation et les utilisateurs, humains ou non. Le niveau supérieur est celui de tous les utilisateurs et propose une indépendance totale vis-à-vis du système d'exploitation sous-jacent. Il est constitué par une interface client-serveur basée sur l'exécution de programmes (CGI) par un serveur HTTP et un affichage des données utilisant essentiellement le langage HTML.

3.7.1 Scripts SuMo natifs

Le langage SuMo permet de donner des instructions au programme `sumo`. Son implémentation permet son utilisation en mode interactif ainsi qu'en fournissants des fichiers de scripts. Dans tous les cas de figure, il est interprété à la volée, et les erreurs sont détectées au fur et à mesure de l'exécution des commandes. Cette restriction importante n'est pas gênante pour une utilisation occasionnelle afin d'effectuer des tests. Elle n'est pas non plus gênante si les scripts sont générés automatiquement par un autre logiciel, comme c'est le cas des programmes CGI de l'interface web de SuMo.

Dans un premier temps est présenté le langage SuMo à proprement parler. Les fonctions disponibles font l'objet d'une section à part entière.

3.7.1.1 Le langage SuMo

Programme Un programme est un mélange de déclarations et d'expressions, séparées par des points-virgules (`;`) si nécessaire.

Expressions Une expression peut être une simple valeur. Dans ce cas, cette valeur est renvoyée. S'il s'agit d'une séquences de valeurs, la première de celles-ci est supposée être une fonction et est appliquée en prenant comme arguments les éléments du reste de la séquence.

Déclarations Après son évaluation, une expression `expr` peut soit être ignorée :

```
expr;
```

soit être associée à un nom :

```
let ident = expr;
```

où `ident` est une séquence de caractères commençant par une minuscule ('a' à 'z') et éventuellement suivie de lettres ou de chiffres ('a' à 'z', '0' à '9' et '_').

ex :

```
sumo> 0;
- : Int
sumo> let x = 0;
x : Int
```

Fonctions Seules les fonctions prédéfinies peuvent être utilisées, mais sont considérées comme n'importe quel objet du langage. Dans l'exemple suivant, nous vérifions d'abord le type de la valeur associée à l'identificateur `print`, puis nous l'associons à un nouvel identificateur `output` et nous le testons.

```
sumo> print;
- : Function
sumo> let output = print;
output : Function
sumo> output "hello";
hello
- : Unit
```

Types prédéfinis Tous les types sont prédéfinis. Il s'agit de `Unit`, `Bool`, `Int`, `Float`, `String`, `Function`, `Message`, `PDB_file`, `DB`, `DB_entry`, `Molecular_graph`, `Comparison_result`, `Multi_comparison_result`, `'a Option`, `List`, `'a Disk_entry`. `'a` (alpha) peut être n'importe quel type, comme en Caml.

Constructeurs

Constructeurs de types Les constructeurs de types sont utilisés pour créer de nouveaux objets d'un type donné. Tous les constructeurs de types commencent par une lettre majuscule prise dans l'intervalle 'A'-'Z'.

Les constructeurs de types sont :

```
PDB_file string
DB string
Message string
DB_entry (string_db, string_id)
DB_entry (db, string_id)
```

où `string`, `string_db`, `string_id` sont des expressions de type `String` et `db` est une expression de type `DB`.

Options Les options sont des constructeurs de types polymorphes arbitraires qui attendent un argument. Elles commencent par le caractère moins ('-') et sont suivies de lettres minuscules.

ex :

```
sumo> print "Hello World!" -file "hello.txt";
- : Unit
```

produit le même effet que :

```
sumo> let fileoption = -file "hello.txt";
fileoption : String Option
sumo> print "Hello World!" fileoption;
- : Unit
```

Listes Les listes sont des ensembles hétérogènes d'éléments. `[]` est la liste vide, `["foo"; 12; 2.3]` est une liste qui contient 3 éléments de type `String`, `Int` et `Float`.

Commentaires Les commentaires commencent par `(*` et se terminent par un `*)` non chevauchant. Les commentaires imbriqués sont acceptés.

Mots-clés et caractères spéciaux Les mots-clés suivants sont réservés par le langage et ne peuvent donc pas être utilisés comme identificateurs :

```
= " ; , ( ) [ ] let true false (* *)
```

Les caractères suivants peuvent être utilisés pour séparer les mots-clés ou pour indenter les scripts : ' ' '\t' '\n' (ASCII décimal : 32, 9, 10).

3.7.1.2 Les primitives du langage SuMo

Nous décrirons ici succintement le rôle des différentes fonctions disponibles dans la version 4.4 de SuMo, sans préciser la nature de leurs arguments ni de leurs différentes options. Une description de ces options est donnée au niveau de l'aide interactive du logiciel. Cette aide est accessible par la fonction `help`. Voici l'affichage du début d'une session `sumo` présentant l'effet de la commande `help ()` :

```
[pc-bioinfo1] ~/devel/sumo/src/sumo % ./sumo -i
        SuMo version 4.4-Boom

Reading PDB definitions from '/home/martin/.sumo/pdb_groups'... done
Reading SuMo definitions from '/home/martin/.sumo/sumo_groups'... done
sumo> help ();
HELP TOPICS: automatic_read compare exit help list_idents multi
             multiselect output_prediction output_std print print_custom
             print_idents read remove screen site_stat_analysis
             site_statistics store update_db_summaries
AVAILABLE FUNCTIONS: automatic_read compare exit help list_idents
                     multi multiselect output_prediction output_std print
                     print_idents read remove site_stat_analysis
                     site_statistics store update_db_summaries
Try 'help "help";' for details on this function.
- : Unit
sumo>
```

La commande `help topic` permet d'accéder à l'aide associée à la rubrique `topic` si elle existe. Le tableau 3.3 page 105 présente le rôle des différentes fonctions disponibles.

Tab. 3.3: Les fonctions du langage SuMo

Identificateur	**Description**
`automatic_read`	Extraction et enregistrement des sites de fixation de ligands
`compare`	Comparaison d'ensembles de structures 3D prétraitées
`exit`	Terminaison du programme avec un code d'erreur éventuel
`help`	Aide interactive
`list_idents`	Liste de tous les identificateurs définis
`multi`	Comparaison multiple à partir des résultats de comparaison 2 à 2
`multiselect`	Élimination des familles de sites caractéristiques considérées comme très proches d'une famille ayant un score supérieur
`output_prediction`	Calcul et exportation de la prédiction de sites fonctionnels pour d'autres applications
`output_std`	Exportation des résultats de comparaison pour d'autres applications
`print`	Affichage de données de types quelconques
`print_idents`	Affichage de tous les identificateurs ainsi que leurs valeurs associées
`read`	Prétraitement de fichier contenant les données structurales en données utilisables pour les comparaisons
`remove`	Suppression d'un fichier
`site_stat_analysis`	Régénération de certaines données statistiques enregistrées au niveau de la base de données

Identificateur	Description
`site_statistics`	Comparaison de tous les sites de fixation de ligands de la base de données standard entre eux
`store`	Enregistrement de données structurales pré-traitées
`update_db_summaries`	Mise à jour d'informations globales sur la base de données

3.7.1.3 Système de sélection 3D

Le système de sélection des groupements chimiques au sein d'une structure consiste en un langage spécifique. Un programme — généralement court — écrit dans ce langage permet de générer un prédicat permettant d'accepter ou de refuser un groupement chimique au sein d'une structure 3D donnée.

Ce langage est celui qui est utilisé pour l'option `-select` de la command `read` de SuMo, et un sous-ensemble de ce langage est utilisé pour l'option `-restrict` de la même commande.

Le langage est basé sur la combinaison de prédicats élémentaires en utilisant les opérateurs booléens classiques `and`, `or` et `not` ou le prédicat `around` qui prend en argument un prédicat et des paramètres numériques.

Les prédicats élémentaires permettent de sélectionner le groupement chimique considéré par différentes informations qu'il porte ou que porte son environnement.

Les constructeurs de prédicats élémentaires Les prédicats basés sur les informations trouvées dans les fichiers PDB nécessitent bien entendu que ces données proviennent initialement d'un fichier au format PDB. Ces données sont considérées comme optionnelles. Actuellement le format PDB est le seul format public utilisé par SuMo, mais ceci pourra changer dans les versions ultérieures. Par contre, les autres prédicats sont basés sur des informations nécessairement connues par SuMo.

Le type de groupement chimique Un prédicat basé sur le type de groupement chimique peut être formulé à l'aide du constructeur `Group`.

Ex. :
```
Group "aromatic"
Group "pdb_atp"
```

L'étiquette du groupement chimique Le constructeur permettant de sélectionner un groupement chimique avec une étiquette donnée est nommé `Label`.

Ex. : `Label "backbone"`

Le numéro de la molécule Le numéro de la molécule assigné par SuMo lors du prétraitement du fichier de structure peut être utilisé. C'est ce constructeur qui est utilisé pour désigner les ligands par exemple lors de la construction de la base de données de sites de fixation de ligands. Le constructeur est `Molecule_index`.

Ex. : `Molecule_index 3`

Le nom PDB du monomère Le nom du groupement ou monomère utilisé dans le fichier PDB peut être utilisé comme critère de sélection. Il s'agit de celui auquel est rattaché le groupement chimique SuMo, s'il est défini. Le constructeur est `Pdb_group`.

Ex. : `Pdb_group "CYS"`

Le numéro PDB du monomère La sélection du groupement chimique peut également s'effectuer sur le numéro du groupement PDB associé au groupement chimique SuMo considéré, à l'aide du constructeur `Pdb_index`.

Ex. : `Pdb_index 57`

Cette sélection peut également s'effectuer sur un intervalle d'indices en utilisant le même constructeur et le mot-clé `to`.

Ex. : `Pdb_index 50 to 60`

Le nom PDB de la chaîne Le nom PDB de la chaîne à laquelle le groupement chimique est éventuellement rattaché peut être sélectionné grâce au constructeur `Pdb_chain`. Notons qu'en pratique le nom d'une chaîne PDB est constituée de zéro ou d'un seul caractère, pas forcément alphanumérique.

Ex. :
```
Pdb_chain "A"
Pdb_chain ""
```

L'opérateur `around` L'opérateur `around` a été conçu pour sélectionner des groupements chimiques dont au moins une des positions-cibles est située à une certaine distance de la position fonctionnelle du groupement chimique considéré, lorsque celui-ci vérifie un prédicat donné. Cet opérateur prend donc en argument un intervalle de distances et un prédicat.

Ex. :
```
4.5 around Molecule_index 5
[1.,3.] around Group "imidazole"
4 around Group "pdb_atp"
```

L'intervalle de distance peut être désigné comme dans les exemples précédents soit par une simple distance d qui sera interprétée comme l'intervalle $[0, d]$ ou directement par un intervalle de distances de la forme $[d_1, d_2]$.

Les opérateurs booléens classiques Les opérateurs booléens permettent de combiner des prédicats. Ce sont les opérateurs binaires `and` et `or`, et l'opérateur unaire `not`.

Priorités La priorité des opérateurs est donnée ici :

$$\texttt{not} > \texttt{and} > \texttt{or} > \texttt{around}$$

Les parenthèses sont prioritaires devant tous les autres opérateurs.

Ex. : `4 around Group "pdb_atp" and not "pdb_atp"`
ne sélectionne rien du fait de la priorité de `and` sur `around`.
`(4 around Group "pdb_atp") and not "pdb_atp"`
permet de sélectionner les groupements chimiques en interaction avec l'ATP, représenté par les groupements fantômes de type `"pdb_atp"`.

Exemples complexes Si l'on souhaite considérer les régions en interaction avec au moins deux atomes de calcium situés l'un de l'autre à une distance comprise entre 3 Å et 7 Å, on peut générer ce prédicat par le programme suivant :

```
(4 around (Group "pdb_ca" and [3.0,7.0] around Group "pdb_ca"))
```

De façon semblable, si l'on souhaite sélectionner les sites de fixation de molécule d'ATP qui ne sont pas en interaction avec ion Mg^{2+}, un script de ce type conviendra :

```
(4 around (Group "pdb_atp" and not 4 around Group "pdb_mg"))
```

3.7.2 Interface CGI/HTML

L'interface conçue pour une utilisation normale de SuMo est celle basée sur le serveur web. Les requêtes peuvent être soumises par les utilisateurs de façon interactive ou directement en envoyant une requête au format SuMoQ.

3.7.2.1 Requêtes interactives

Une *requête interactive* consiste en plusieurs étapes successives, depuis le chargement des données structurales jusqu'à l'analyse détaillée des résultats. Voici ces étapes :

1. Chargement des données structurales en indiquant un identificateur PDB ou en chargeant son propre fichier PDB
2. Propositions de sélection dans la structure au niveau des différentes chaînes ou des sites de fixation de ligands
3. Affichage des groupements chimiques SuMo générés et indication de ceux qui seront effectivement pris en compte dans la représentation utilisée pour les comparaisons. Choix de la base de données à cribler ou d'un sous-ensemble de base de données. Lancement des comparaisons.
4. Attente pendant les comparaisons avec affichage de la progression des calculs
5. Affichage des résultats globaux sous différentes formes possibles. Enregistrement ou exportation des résultats.
6. Affichage de détails sur chaque site mis en évidence avec visualisation

Le lecteur est invité à essayer le système par lui-même, à partir de l'adresse actuelle du serveur SuMo :

```
http://sumo-pbil.ibcp.fr
```

Toute requête de comparaison utilisant le serveur web SuMo passe par une étape où elle est formulée dans le langage SuMoQ. Cette conversion est faite de façon transparente pour l'utilisateur du mode interactif. Cependant, il a la possibilité de sauvegarder sa requête au format SuMoQ. Les intérêts de sauvegarder la requête plutôt que de simplement conserver les résultats sont essentiellement les suivants :

– Ce fichier fait office de ■ matériel et méthodes ■, c'est-à-dire qu'il permet de décrire exactement la démarche effectuée au niveau de SuMo, et de reproduire les mêmes résultats aux mises à jour près de la base de données et pour une version donnée du logiciel.
– La faible taille du fichier contenant la requête permet de le stocker plus facilement que les résultats parfois très volumineux de SuMo.
– Contrairement aux résultats de comparaisons, une requête reste valable d'une version à l'autre de SuMo. Sa conservation permet une mise à jour simple et sans erreur des résultats lors d'un changement de version de SuMo.
– Le fichier généré automatiquement peut servir de modèle pour générer des requêtes plus complexes impossibles à formuler en mode interactif.

La section suivante décrit le langage de requêtes mis au point.

La limitation de ce système est néanmoins que toutes les structures utilisées doivent être disponibles et identifiables au niveau du serveur web. Un code temporaire est néanmoins associé à chaque structure fournie par l'utilisateur. Ce code est réutilisable dans d'autres requêtes tant que les données sont conservées au niveau du serveur.

3.7.2.2 Requêtes SuMoQ

Introduction Le langage de requêtes SuMoQ permet de fournir directement des requêtes de comparaisons éventuellement complexes au serveur web, en une seule étape. Voici un exemple de requête simple :

```
{
  email = "martin_jambon@emailuser.net";
  title = "My query for scanning ligand binding sites";
  {
    subtitle = "Lectin 2PEL";
    scan = {
      database = "ligands";
      pdb_id = "2PEL";
      selection = "Pdb_chain \"A\"";
    } /scan;
  } /
}
```

Cette requête permet d'effectuer une comparaison de la chaîne A de la structure PDB d'identificateur standard 2PEL avec chacun des sites de fixation de ligands de la base de données. Un e-mail sera envoyé à l'adresse indiquée lors du lancement des comparaisons et lorsque les comparaisons seront terminées en indiquant l'URL permettant d'accéder aux résultats. Les titres et sous-titres sont optionnels. Si précisé, le sous-titre est utilisé dans le champ `Subject` des e-mails.

Syntaxe La syntaxe utilisée a été mise au point pour SuMo, initialement pour représenter de façon arborescente les données structurales tirées de la PDB. Cette syntaxe permet de la même façon que XML de représenter toutes sortes de données différentes, du moment qu'elles s'expriment sous une forme hiérarchique. La syntaxe mise au point permet de s'affranchir de certaines lourdeurs présentes au niveau de XML tout en fournissant une syntaxe encore plus simple. Les différents aspects de cette syntaxe sont les suivants :

– Chaque noeud comporte un certain nombre de champs. Ces champs sont anonymes ou étiquetés et associés à une donnée.
– Un noeud ne peut pas avoir plusieurs champs portant la même étiquette mais peut comporter un nombre indéfini de champs anonymes.
– Une donnée est constituée par une valeur atomique ou par un noeud.
– Les données atomiques sont typées. 5 types différents existent.
– Les marques de fin de champ sont facultatives mais doivent être correctes si elles sont utilisées.
– Des identificateurs peuvent être utilisés pour éviter la répétition de certaines données.

Cette syntaxe est décrite ci-dessous. Les symboles terminaux sont en minuscules, en gras et en couleur. Les symboles non-terminaux sont en majuscules. Les expressions régulières correspondant aux caractères (CHAR) et aux chaînes de caractères (STRING) ne sont pas détaillées. Les commentaires sont ouverts par (* et fermés par *) et peuvent être imbriqués. Les espaces, tabulations, retours à la ligne et retours chariot sont ignorés s'ils ne font pas à l'intérieur d'une constante CHAR ou STRING.

```
TREE     ::= INT
           | FLOAT
           | BOOL
           | CHAR
           | STRING
           | { ((LABEL =) ? TREE (/LABEL) ? ;)* }
           | IDENT
           | define IDENT = TREE
INT      ::= -?[0-9]+
FLOAT    ::= -?[0-9]*.[0-9]*
BOOL     ::= true
           | false
STRING   ::= "..."
CHAR     ::= '...'
IDENT    ::= [a-z][a-z,A-Z,_,0-9]*
LABEL    ::= IDENT
```

L'intérêt essentiel de cette syntaxe par rapport à XML est la combinaison de la facilité d'édition de tels fichiers et de la lisibilité, grâce à l'utilisation de marques de fin de champ uniquement lorsque l'on juge que c'est nécessaire. Les champs anonymes permettent de définir des ensembles d'objets sans avoir à les associer à une paire de balises du genre <item> et </item>. Pour exprimer un ensemble de nombres entiers, il suffira d'écrire par exemple {1 ; 83 ; 2 ; 122} au lieu de quelque chose comme <list> <n>1</n> <n>83</n> <n>2</n> <n>122</n> </list> en XML.

Tout langage basé sur cette syntaxe est libre d'interpréter les champs de la façon dont il le souhaite.

Sémantique de SuMoQ Dans SuMoQ, certains champs sont obligatoires, d'autres sont facultatifs. Certains champs comme email sont obligatoires dans un certain contexte et facultatifs dans un autre contexte. L'ordre des champs étiquetés ou anonymes n'a pas d'importance.

Toute requête SuMoQ est constituée en réalité d'un nombre quelconque de requêtes de comparaison. Grâce à une seule requête SuMoQ on peut soumettre plusieurs requêtes de comparaisons qui seront traitées par le serveur dans un ordre indéterminé.

La spécification complète du langage SuMoQ est donnée par la figure 3.15 page 114. Le champ email doit être spécifié si la requête concerne au moins un criblage de base de données complète. D'autre part, si le serveur considère que la requête demandée est trop coûteuse, il est possible qu'il la refuse. Par exemple la taille de la structure sous-structure utilisée pour cribler la base de données de structures-cibles est limité à un certain nombre de triplets. Ce paramètre peut être fixé librement par l'administrateur système au moment de l'installation de SuMo.

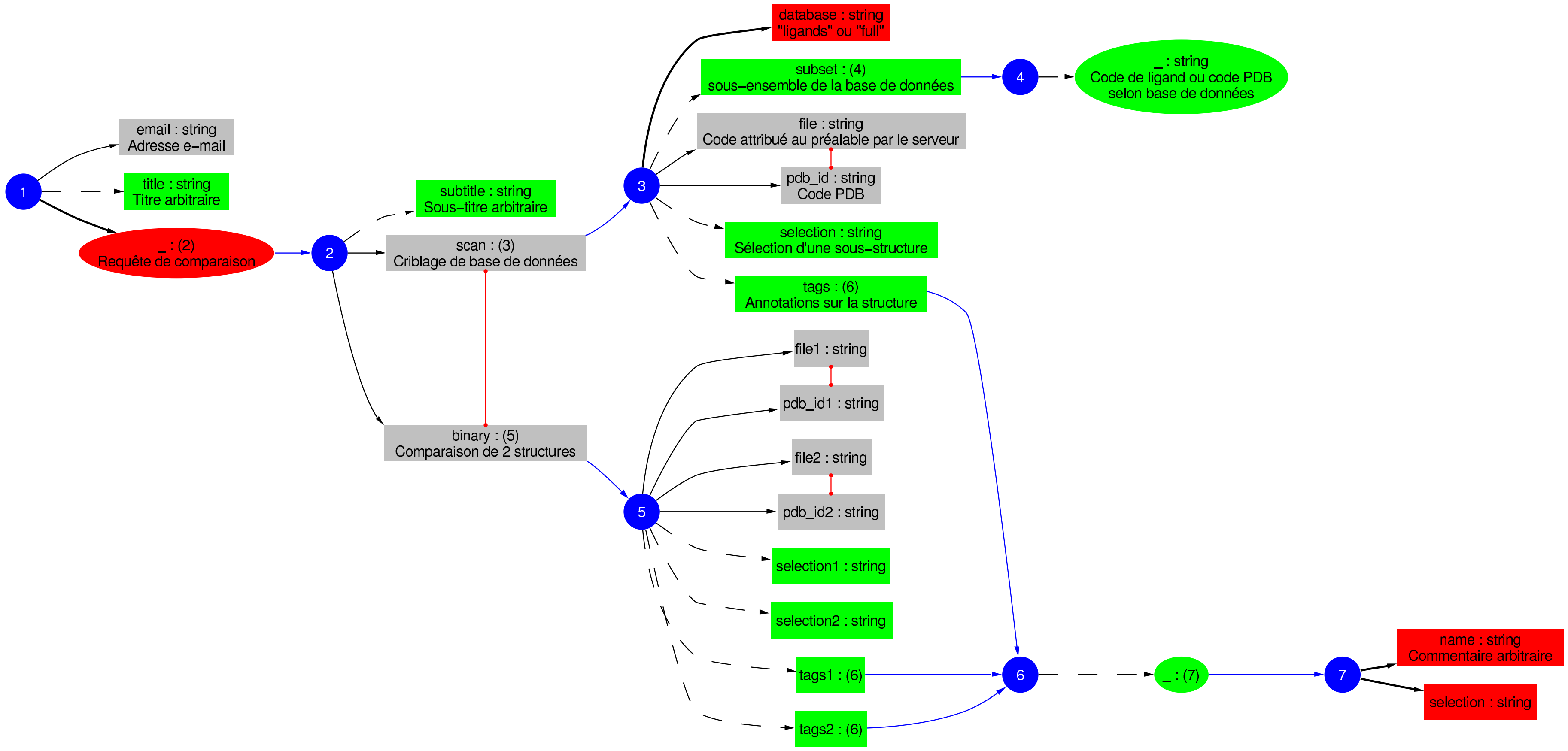


FIG. 3.15 – Spécification de la structure des requêtes au format SuMoQ. Les ronds numérotés indiquent les différents types de noeuds. Le type (1) est celui du noeud-racine. `field : type` indique un champ étiqueté par `field` et de type `type`. _ désigne un champ anonyme. Les champs anonymes sont représentés par des ellipses. Ils peuvent être présents en nombre indéfini sous un même noeud. Les champs toujours obligatoires sont en rouge, les champs toujours facultatifs sont en vert, et les champs parfois obligatoires sont en gris. Les connecteurs rouges non orientés indiquent qu'un des deux champs doit être spécifié.

Les champs `selection`, `selection1` et `selection2` doivent être associés à une chaîne de caractères correspondant à un prédicat de sélection de groupements chimiques tel que décrit section 3.7.1.3 page 106.

Les champs `tags`, `tags1` et `tags2` permettent d'annoter des régions arbitraires au moyen d'un commentaire librement choisi et d'un prédicat de sélection.

La comparaison de 2 structures complètes est possible en utilisant le champ `binary`. Ce type de comparaison n'est pas proposé en mode interactif en raison de la difficulté d'interprétation de résultats de telles comparaisons.

3.7.2.3 Présentation des résultats

HTML Les résultats d'une requête de comparaison sont affichés par défaut sous la forme d'une page HTML. Chaque paire de sites similaires que SuMo a détectée constitue un ligne d'un tableau rassemblant quelques informations essentielles, dont le titre et la description courte extraite du fichier PDB d'origine. Le nombre de lignes du tableau pouvant être assez élevé, seules 50 lignes sont affichées à la fois et des liens permettent de naviguer de page en page ou d'afficher la totalité du tableau. Un tri selon l'une ou l'autre des colonnes peut être effectué.

Au niveau de chaque paire de sites similaires, un lien permet d'accéder à plus de détails sur cette correspondance. La liste de correspondances entre groupements chimiques est présentée. Des images sont générées automatiquement de façon à mettre en évidence les acides aminés portant les groupements chimiques des deux sites. Les deux sites sont présentés dans une orientation permettant de les visualiser au mieux. Cette orientation consiste à faire pointer vers l'oeil de l'observateur la moyenne des vecteurs $\overrightarrow{CP}$ tels que définis section 3.2.1.2 page 52, au niveau du premier site. Le deuxième site est orienté de façon à minimiser le RMSD entre les positions fonctionnelles des groupements chimiques correspondants. Les images sont obtenues après modification des coordonnées grâce aux logiciels MolScript [25] et Raster3D [2, 34, 33]. Des scripts permettant de visualiser les structures 3D et les sites détectés directement avec RasMol sont également proposés. Des liens divers vers des sites web externes sont également proposés.

Deux autres types d'affichage des résultats globaux sont proposés. Il s'agit de la liste des ligands associés au score du meilleur site de fixation détecté pour chaque ligand. La liste est triée par ordre décroissant de scores. L'autre affichage est la prédiction des sites de fixation de ligands par annotation des groupements chimiques de la structure requête et utilisation de la fonction spécificité définie section 3.5 page 77.

Sauvegarde des résultats Outre la requête au format SuMoQ, les résultats de la ou des requêtes de comparaison peuvent être sauvegardées, c'est-à-dire enregistrées dans un format qui puisse être relu par le serveur SuMo. Cette sauvegarde n'est pas forcément nécessaire puisque les données sont conservées quelques jours au niveau du serveur SuMo. Il suffit de mémoriser le code associé à la requête pour pouvoir accéder à toutes les données ultérieurement.

Format Le format des données est celui généré par les fonctions du module Marshal d'Objective Caml. Pour relire ces données, il faut connaître leur type et en être sûr.

Signature cryptographique Afin d'éviter les plantages dûs au chargement de données dans un format incorrect, que ce soit accidentel ou volontaire, il est nécessaire de s'assurer que ces données ont été générées par le serveur web SuMo.

Pour cela, une signature cryptographique est ajoutée en tête du fichier. Le protocole de génération de la signature qui a été mis en place est le suivant :

1. Calcul d'un résumé cryptographique du fichier, c'est-à-dire une chaîne de caractères courte très difficile à deviner si les données ne sont pas connues de façon exacte. Ceci constitue une signature non cryptée des données.
2. Cryptage de la signature.

Le module Digest d'Objective Caml est utilisé pour générer la signature non cryptée. Le cryptage de la signature utilise la bibliothèque Cryptgps. La clé permettant de décrypter la signature est uniquement connue par le serveur web SuMo et change à chaque fois que le format des résultats change. Le fait de crypter une signature plutôt que le fichier complet permet justement de rendre les résultats lisibles par SuMo ou tout programme qui peut relire ce format, simplement en ignorant la signature placée en début de fichier.

Restriction aux meilleurs résultats Les paramètres de SuMo ont été mis en place de façon à ce que tous les résultats retournés soient susceptibles d'intéresser les utilisateurs dans un grand nombre de cas. Néanmoins, vu le volume important des résultats et la volonté de certains utilisateurs de ne conserver que les résultats avec de très bon scores, il est possible de préciser un seuil au-dessous duquel on considère que les résultats ne sont pas intéressants. Cette commodité ne permet pas de régénérer une annotation prédictive de sites de fixation de ligands en utilisant le nouveau seuil.

Exportation L'exportation de données, contrairement à la sauvegarde, ne permet pas de les relire grâce au logiciel qui les a générées. Les résultats de SuMo peuvent être exportés dans différents formats, répondant à la demande des utilisateurs.

La version 4.5 de SuMo propose l'exportation dans les formats suivants :
- Format XML. Permet d'enregistrer les mêmes données que celles sauvegardées mais dans un format facilement portable.
- Format tableur. Organisation des données en lignes et colonnes.
- Format texte libre. Lisible mais non conçu pour être traité par un logiciel.

Format XML Le format XML utilisé est celui généré directement à partir du type Caml des résultats de comparaison. Ceci est réalisé grâce à la bibliothèque IoXML [16] qui fournit une extension syntaxique de Caml en utilisant Camlp4.

Format tableur Un fichier organisé en lignes et colonnes peut être généré à partir des résultats de comparaison. Ce mécanisme permet de traiter les données chiffrées des résultats de comparaison de façon plus complexe qu'avec les fonctions de tri fournies au niveau de la page web.

Le format peut être modifié par l'utilisateur de différentes façons :
- Choix du séparateur de colonnes.
- Choix des champs à exporter et de leur ordre.
- Affichage optionnel de la spécification des colonnes.
- Sélection de lignes selon les annotations structurales au niveau des sites détectés.

La chaîne de caractères servant de séparateur de colonnes par défaut est un point-virgule. Pour changer le séparateur, il suffit de remplir la case appropriée au niveau du formulaire HTML.

Le nombre de champs pris en compte est assez important. Ceci peut résulter en des fichiers très volumineux qui contiennent beaucoup de données qui n'intéressent pas forcément l'utilisateur. Pour cela, l'utilisateur peut fournir son propre format. Le format utilisé consiste en une chaîne de caractères quelconque dans laquelle, au niveau de chaque paire de sites considérée, les champs de la forme {*id*} sont remplacés par la valeur associée à *id* ou bien un point d'interrogation si *id* n'est pas un nom de champ valide. Voici un exemple de format :

```
{molecule_identifier2} & {molecule_name2} & {sumo_score} \\
```

Ce format permet de générer un tableau à trois colonnes valide pour LaTeX, et dont les lignes sont formées comme dans cet exemple :

```
1BOU & ATP & 2.69744 \\
```

Le nom des différents champs disponibles est obtenu lorsque le format par défaut est choisi. En effet la première ligne du fichier est constituée par la chaîne de caractère indiquant le format. Une option permet de ne pas inclure le format dans le fichier généré.

Les annotations d'ensembles de groupements chimiques ont éventuellement pu être réalisées. Rappelons qu'une annotation est un couple (E, S) où E est un ensemble de groupements chimiques et S un commentaire. Certaines annotations sont réalisés automatiquement lors de la construction de la base de données ou par l'utilisateur lors de la formulation des requêtes de comparaison au format SuMoQ. Chaque groupement chimique peut être associé à un nombre indéfini d'annotations. L'intersection entre un ensemble de groupements chimiques annoté donné et un site donné est appelée *intersection de zone annotée*. L'intersection de zone annotée de taille maximale pour un site donné est appelée *annotation maximale*. La taille maximale est déterminée en réalité de 3 façons différentes :

- volume des groupements chimiques SuMo,
- somme des coefficients des groupements chimiques,
- nombre de groupements chimiques.

Certains paramètres de l'annotation maximale de chaque site considéré font l'objet de champs spécifiques. Ces champs contiennent le commentaire associé à l'annotation et des informations de taille. Pour permettre à l'utilisateur de n'extraire que les sites concernés par une annotation spécifique, un mécanisme de sélection a été mis en place. Les champs optionnels ■ Tag filter 1 ■ ou ■ Tag filter 2 ■ permettent de sélectionner les annotations qui intéressent l'utilisateur par la formulation d'une expression régulière. Ainsi, les annotations ne validant pas l'expression régulière donnée sont ignorées pour la recherche de l'annotation maximale.

Les filtres d'annotation sont des expressions régulières où `*` désigne une chaîne de caractères quelconque, `?` désigne un caractère quelconque et `|` permet de séparer différents motifs au choix.

Ex. : `sacch*binding|gluc*binding`

Format texte libre Ce format n'a pour but que de regrouper l'ensemble des résultats de comparaison en un seul fichier lisible sans formattage préalable. Les groupements chimiques équivalents sont indiqués au niveau de chaque paire de sites.

Ces fichiers sont générés directement par la fonction `print` de `sumo`.

3.7.2.4 Système d'aide en ligne

Un système d'aide en ligne a été mis en place et est régulièrement développé. Il a pour but d'apporter des réponses aux interrogations les plus fréquentes des utilisateurs. Le sommaire de l'aide est accessible depuis n'importe quelle page web de SuMo grâce au lien HELP, reproduit figure 3.16.

FIG. 3.16 – Sommaire de l'aide en ligne de SuMo.

Certains termes employés dans les différentes pages web de SuMo ne sont pas expliqués *in situ* pour des raisons de place et de lisibilité, mais un astérisque permet d'accéder directement à la rubrique d'aide correspondante. Toutes les rubriques d'aide de la version 4.5 sont reproduites en annexe page 163.

3.7.2.5 Développement du système

Nous verrons dans cette section quelques points importants mis en jeu dans le développement de l'interface web de SuMo.

Langages et bibliothèques externes utilisés Comme le reste du système SuMo, tous les programmes développés pour l'interface web sont écrits en Objective Caml. Le tableau 3.4 présente les différents logiciels externes et les bibliothèques utilisés pour la compilation ou l'exécution du système. Les utilitaires classiques tels que GNU `make` ou `bash` ne sont pas mentionnés.

Tab. 3.4: Bibliothèques et logiciels externes utilisés pour le développement du serveur web SuMo.

Nom	Nature	Fonction dans SuMo
Objective Caml	Compilateur	Compilation de tous les programmes.
Camlp4	Préprocesseur	Utilisation des extensions syntaxiques Printfer et IoXML.
OcamlMakefile [35]	Makefile	Makefile générique pour la compilation de projets complexes utilisant OCaml.
IoXML	Bibliothèque	Extension syntaxique de Caml pour générer du code XML à partir de n'importe quel type de données.
Cryptgps	Bibliothèque	Bibliothèque de cryptographie utilisée pour générer les signatures cryptées des données sauvegardées par les utilisateurs.
Ocamlnet [45]	Bibliothèque	Récupération des paramètres CGI.

TAB. 3.4: (suite)

Nom	**Nature**	**Fonction dans SuMo**
MolScript	Logiciel	1re étape de génération dynamique d'images des sites similaires détectés par SuMo.
Raster3D	Logiciel	2e étape de génération dynamique d'images des sites similaires détectés par SuMo.

Extension syntaxique Printfer

Motivations Dans le langage Caml comme en C et dans de nombreux autres langages de programmation, une fonction nommée `printf` permet d'effectuer l'affichage de chaînes de caractères. Cette fonction prend en argument une chaîne de caractères appelée format et des arguments de différents types que l'on souhaite convertir en chaîne de caractère et insérer à une position précise.

Ex. : `printf "Nous sommes en %s %i.\n" mois année ;` permet de remplacer `%s` par la chaîne de caractères liée à l'identificateur `mois` et `%i` par la conversion de `année` au format décimal. Si `mois` est la chaîne de caractères `"Mars"` et `année` est l'entier `2003`, voici ce qui est affichée sur la sortie standard :
`Nous sommes en Mars 2003.`

Lorsque la longueur du texte et le nombre de trous à remplir deviennent important, il devient très difficile de retrouver quel argument correspond à quel trou.

Ex. :
```
printf "Le %02i/%02i/%i, à %i heures %i."
  jour mois année heures minutes ;
```
devient délicat à gérer lorsque le texte inséré entre les trous est beaucoup plus long et que l'on souhaite insérer d'autres trous ultérieurement ou inverser des phrases du texte.

Lors de la génération de code répondant à une syntaxe précise, il est important que le code source réalisant l'affichage soit très lisible. Un autre obstacle à la lisibilité est la protection des caractères spéciaux pour Caml qui se retrouvent fréquemment dans le code que l'on souhaite générer, comme par exemple `"` pour le HTML.

Syntaxe Pour atteindre notre but, il suffit de mettre au point une syntaxe permettant d'inclure les arguments directement dans le texte que l'on souhaite afficher. Camlp4 permet de réaliser de telles extensions syntaxiques pour Caml. Une citation (*quotation*) Camlp4 à été définie. Cette extension a été nommée *Printfer*. Elle a été conçue particulièrement pour être utilisée dans des programmes CGI qui génèrent du code HTML, mais peut servir de remplacement à `printf` dans n'importe quel contexte. Voici un exemple :

```
<< Le $%02i{jour}/$%02i{mois}/$%i{année}, je vous donne rendez-vous,
Monsieur $name{nom}, à $int{heures} heures $int{minutes}.
Je vous invite à consulter la <a href="$doc_url">documentation</a> au
préalable. >>
```

sera remplacé par Camlp4 par le code Caml suivant :

```
Pervasives.print_string "Le ";
Printf.printf "%02i" jour;
Pervasives.print_string "/";
Printf.printf "%02i" mois;
Pervasives.print_string "/";
Printf.printf "%i" année;
Pervasives.print_string ", je vous donne rendez-vous, \nMonsieur ";
print_name nom;
Pervasives.print_string ", \224 ";
print_int heures;
Pervasives.print_string " heures ";
print_int minutes;
Pervasives.print_string ".\nJe vous invite \224 consulter la <a href=\"";
Pervasives.print_string doc_url;
Pervasives.print_string "\">documentation</a> au\npr\233alable."
```

Dans cet exemple, seule la fonction `print_name` doit être définie au préalable, ainsi évidemment que les données insérées dans le texte.

Nous ne donnerons pas ici une description complète de la syntaxe, mais simplement les éléments essentiels. Ces éléments sont rassemblés dans le tableau 3.5 page 123.

Des variantes existent par rapport aux constructions présentées. Si des crochets `[ ]` sont utilisés à la place des accolades `{ }`, alors `output_`*id*

Construction	**Code Caml généré**
`$`*id*	`Pervasives.print_string` *id*`;`
`${...}`	`Pervasives.print_string (...);`
`$`*id*`{`*arg1 arg2 ...*`};`	`print_`*id* *arg1 arg2 ...*`;`
`$%`*format*`{...}`	`Printf.printf "%`*format*`" (...);`

TAB. 3.5 – Constructions principales de l'extension syntaxique Printfer

`Pervasives.stdout` est utilisé à la place de `print_`*id*. Normalement, n'importe quel code Caml valide peut être placé entre accolades ou entre crochets, sauf les commentaires, qui sont traités de façon légèrement simplifiée. L'utilisation de `$>` à la place de `$` permet de vider le tampon de la sortie standard avant d'effectuer l'opération demandée. Un caractère `$` peut être affiché grâce à `$$`. Les chaînes `<<` et `>>` peuvent être générées grâce à `\<<` et `\>>`. Le caractère `\` peut être produit par `\\`. Les espaces en début et fin de citation sont ignorés.

L'extension syntaxique développée est utilisée pour l'affichage de code HTML dans tous les programmes CGI du serveur SuMo.

3.8 Gestion des tâches

Nous verrons dans cette section comment sont réparties physiquement les tâches effectuées par le système SuMo et comment elles sont gérées d'une manière générale.

3.8.1 Système de file d'attente propre

Un utilitaire permettant de gérer une file d'attente au niveau d'une machine unique a été mise en place pour SuMo. Les intérêts de ce programme, nommé `jobqueue` sont les suivants :

- Ne nécessite pas le lancement explicite d'un démon au démarrage de la machine.
- Inclusion dans la distribution de SuMo.
- Priorités des requêtes basées directement sur des estimations de durée.

3.8.1.1 Architecture

3 types de `jobqueue` sont mis en jeu dans la gestion des tâches :

1. Des processus enregistrent les requêtes, c'est-à-dire créent un fichier contenant des directives dans le répertoire assigné à la file d'attente.

Ces processus peuvent lancer le gestionnaire de la file d'attente si ce processus n'existe pas.

2. Un processus central ou *démon* joue le rôle de gestionnaire de la file d'attente. Il examine l'ensemble des requêtes enregistrées lorsqu'un processus lui signale qu'il y a du nouveau, et lance les processus qui vont lancer les tâches. Sa durée de vie est indéterminée.
3. Les processus chargés de lancer les tâches sont générés par le gestionnaire de tâches. Leur rôle est d'assurer le lancement effectif d'une tâche et d'attendre qu'elle se terminent, afin de récupérer la valeur de retour du processus. Ces processus peuvent également envoyer des e-mails lors du lancement de la tâche et une fois qu'elle est terminée.

Les interactions entre ces processus sont schématisées au niveau de la figure 3.17.

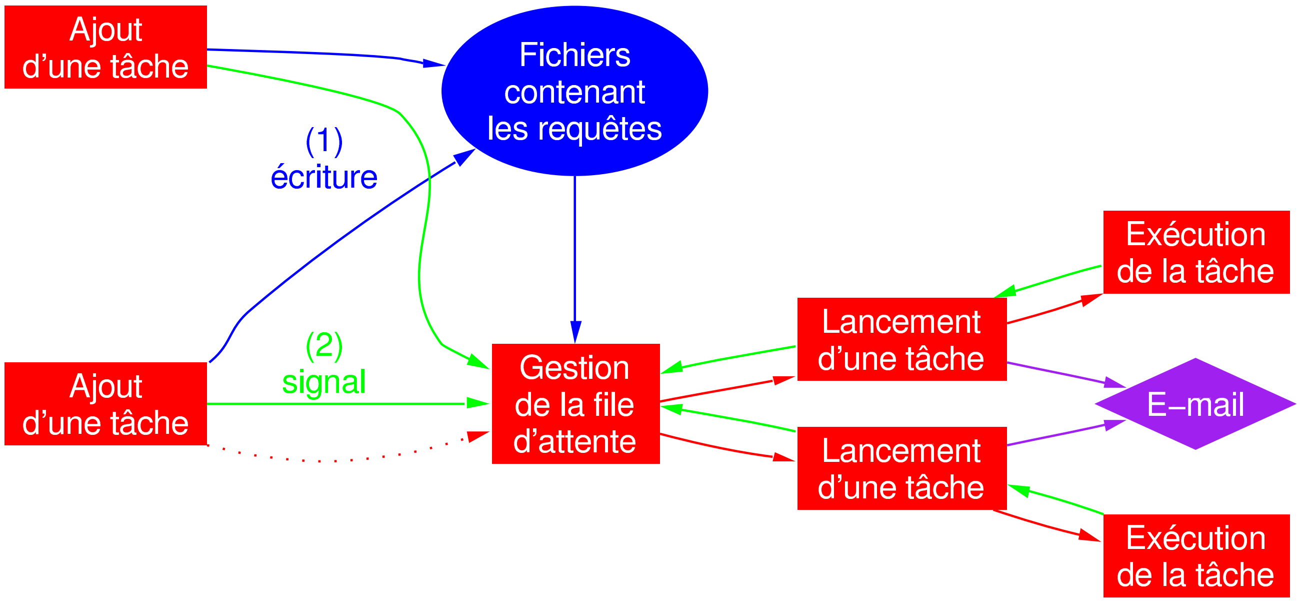


Fig. 3.17 – Système de file d'attente `jobqueue`. Les flèches rouges indiquent le lancement de nouveau processus via fork/exec. Les flèches vertes indiquent les signaux envoyés entre processus.

3.8.1.2 Gestion des priorités

Propriété 3 *Le système de file d'attente est conçu de façon à ce qu'en moyenne, la durée de l'attente soit proportionnelle à la durée de la tâche effectuée.*

Pour que la propriété 3 page 124 soit vérifiée, une estimation de la durée de chaque tâche doit être donnée par l'utilisateur au moment de l'enregistrement de la requête. L'unité de temps est libre, du moment que toutes les requêtes sont formulées en utilisant la même unité pour une file d'attente donnée. A chaque signal reçu, le gestionnaire de file d'attente réévalue les priorités dynamiques, qui dépendent de l'attente déjà effectuée pour chaque requête :

$$\text{priorité} = \frac{\delta}{e}$$

où e est la durée estimée de la tâche et δ est l'attente déjà effectuée. Lorsqu'une tâche peut être lancée, le gestionnaire de tâches va choisir celle dont la priorité est la plus élevée à cet instant donné.

En sous-estimant volontairement la durée d'une tâche, on va donc augmenter sa priorité, inversement si l'on surestime cette durée, on pourra diminuer sa priorité.

Ce système a l'avantage de rendre l'attente des utilisateurs en rapport avec la tâche demandée. Ceci est particulièrement utile pour les requêtes SuMo, où l'estimation de durée est essentiellement basée sur le nombre de comparaisons à effectuer. En effet, lorsqu'un utilisateur indique un sous-ensemble de la base de données à cribler, le nombre de comparaisons est fréquemment réduit d'un facteur 1000 ou plus.

3.8.1.3 Mode de lancement du démon

Le lancement du gestionnaire de tâches est effectué par l'utilisateur de la file d'attente au moment où il ajoute une tâche et seulement si ce processus n'existe pas. Le nombre de tâches s'exécutant simultanément doit être spécifié s'il ne correspond pas à la valeur par défaut. La file d'attente est désignée par un répertoire dans lequel l'utilisateur doit avoir les droits d'écriture et de lecture.

3.8.2 Parallélisation sur une machine multi-processeurs

La fonction de comparaison de SuMo a été parallélisée de façon à tirer profit des machines multi-processeurs. La parallélisation n'utilise pas de bibliothèque particulière. Elle consiste en la répartition des différentes comparaisons à effectuer en un certain nombre de paquets, qui seront traités par un nombre fixe de processus-fils. Ces processus se terminent en écrivant dans un fichier les résultats de comparaison. Le processus-père récolte et fusionne les données de ces fichiers une fois que toutes les sous-tâches ont été effectuées.

Notons que la fonction permettant de paralléliser une requête est polymorphe. Elle peut être utilisée pour n'importe quelle itération d'une fonction sur un tableau. La seule condition est que le résultat puisse être converti par les fonctions du module Marshal d'Objective Caml.

3.8.3 Distribution des tâches sur un parc de machines

Le système de file d'attente `jobqueue` développé ne permet pas de répartir les tâches sur un parc ou *cluster* de machines. L'implémentation de la version 4.4 de SuMo fournit une interface au système PBS. C'est ce système qui est actuellement utilisé par le serveur SuMo public.

3.9 Questions fréquemment posées (FAQ)

3.9.1 Au sujet de la méthode

Question 1 – Pourquoi utiliser des triplets de groupements chimiques ?
Plusieurs considérations ont conduit à préférer des regroupements de trois groupements chimiques plutôt qu'un autre nombre.

- En dimension k, un k-uplet de points non-alignés permet de définir un hyperplan de $\mathbf{R}^k$. Un hyperplan à l'avantage d'être défini à l'aide d'un unique vecteur de $\mathbf{R}^k$. En dimension 3, il s'agira donc d'un plan défini à partir de 3 points.
- L'utilisation de triplets comme sommets du graphe permet d'associer un angle au niveau de chaque arête, ce qui permet la détection de sous-structures superposables. Par contre, en utilisant des groupements chimiques isolés ou des paires de groupements chimiques comme sommets du graphe, il n'est pas possible de garantir directement que des graphes A–B–C–D et A'–B'–C'–D' correspondent à des régions superposables. En utilisant des k-uplets où k est supérieur à 3, il n'y a pas non plus de notion d'angle entre les k-uplets équivalente à l'angle formé entre les plans des triplets.
- Il n'y a pas besoin de définir des k-uplets où k est plus grand que 3 dans la mesure où ils sont représentés par un ensemble de triplets.
- Le nombre de triplets constitués de type différents, c'est-à-dire de constitués de groupements chimiques différents est de l'ordre de 1000. Ceci permet d'effectuer un hâchage dans le but de ne comparer que les paires de triplets de même type.
- En dimension 3, 3 points sont nécessaires et suffisants pour fixer un ligand sans rotation possible (localement). Il semble donc assez peu

restrictif de dire qu'au moins 3 points d'ancrage sont nécessaires à la fixation d'un ligand. La détection de sites fonctionnels non basés sur la fixation d'un ligand n'est pas l'objectif prioritaire de SuMo.

Question 2 – La flexibilité des chaînes latérales est-elle prise en compte ?
Oui, dans la mesure où leurs fluctuations ne dépassent pas certains seuils. SuMo n'est néanmoins pas un outil de dynamique moléculaire. Si plusieurs conformations sont acceptables pour une molécule, chacune doit faire l'objet d'une requête SuMo. Actuellement, seule la première structure d'un faisceau de structures donné dans un fichier PDB est considérée par SuMo.

Question 3 – Peut-on changer les paramètres comme on veut ?
Un des objectifs de SuMo est la simplicité d'utilisation. L'utilisateur de SuMo à travers le serveur web n'a pas la possibilité de modifier les paramètres de SuMo, pour des raisons de simplicité d'interprétation des données. Au niveau de l'outil `sumo` en ligne de commande, de nombreux paramètres sont modifiables ainsi que le fichier de définition des groupements chimiques. Plus le public est large, plus les connaissances que l'on peut exiger de lui sont faibles.

Question 4 – Avez-vous effectué une validation statistique ?
Il ne peut y avoir de validation universelle d'un outil qui traite un problème qui n'est pas formulé en termes mathématiques. Néanmoins, certaines fonctions de score peuvent aider l'utilisateur pour synthétiser les résultats qu'il a obtenus par SuMo en introduisant une notion de spécificité (voir section 3.5 page 77). Une statistique alternative pourrait consister à demander l'avis d'un grand nombre d'utilisateurs à travers des questions comme *SuMo vous est-il utile ?* ou *SuMo vous a-t-il fait gagner du temps dans vos recherches ?*

Question 5 – SuMo peut-il considérer comme similaires des groupements chimiques de types différents ?
Non. Chaque type de groupement chimique correspond à un identificateur. Il ne peut y avoir équivalence qu'entre groupements chimiques de même type. Si l'on souhaite exprimer que l'histidine possède un cycle imidazole aux propriétés catalytiques bien particulières et que ce cycle peut simplement être utilisé pour faire du stacking comme n'importe quel cycle aromatique, il faut définir deux types de groupements chimiques d'identificateurs `aromatic` et `imidazole`, dont la position physique est la même mais dont les paramètres géométriques sont spécifiques de la propriété représentée (voir section 3.2.1.2 page 50).

Question 6 – Prenez-vous en compte les interactions spécifiques, comme un stacking aromatique/guanidinium ?
Oui. Il n'est pas nécessaire de définir un groupement chimique à part entière pour ce type d'interaction, puisque par défaut les groupements chimiques `aromatic` et `guanidinium` sont prédéfinis de façon à prendre en compte ce type d'interactions. D'une manière générale, toute interaction entre groupements chimiques est prise en compte par SuMo à partir du moment où les propriétés chimiques et géométriques permettant l'interaction sont prises en compte au niveau de chacun des partenaires.

Question 7 – Peut-on utiliser SuMo sur des modèles par homologie ?
La pertinence des résultats est laissée à l'appréciation des utilisateurs en fonction de la qualité du modèle. SuMo est un outil d'aide à l'analyse de structures de macromolécules de résolution atomique. Dans bon nombre de cas, il est probablement plus judicieux de travailler sur les structures expérimentales qui ont permis de dériver le modèle que sur le modèle lui-même.

Question 8 – Plusieurs modèles issus d'une dynamique moléculaire donnent-il les mêmes résultats avec SuMo ?
Tout dépend de l'ampleur des fluctuations et du réalisme de la simulation. SuMo détecte des similitudes entre structures de protéines identiques déterminées par des équipes différentes. Si une dynamique n'impose pas des fluctuations plus importantes que les erreurs expérimentales tolérées, les résultats obtenus par SuMo en utilisant des structures prises à différentes dates de la simulation devraient donner des résultats semblables.

3.9.2 Au sujet de l'implémentation

Question 9 – Le choix de Caml comme langage de programmation n'est-il pas un frein à l'industrialisation ?
Poser cette question avant le développement de SuMo est tout-à-fait judicieux et devrait s'appliquer à tous les langages de programmation. En effet, **avant de débuter un projet**, on peut douter en particulier des performances des compilateurs, de la facilité d'utilisation du langage et de la disponibilité de certaines bibliothèques. Néanmoins, poser cette question lorque le projet est déjà abouti et sur le point d'être commercialisé n'a pas beaucoup de sens.

Question 10 – Caml n'est-t-il pas plus lent que le C++ ?
Le lecteur pourra en particulier se reporter à l'adresse

```
http://www.bagley.org/~doug/shootout/
```

pour avoir quelques idées sur cette question. D'autre part, la lenteur d'un programme dépend avant tout des algorithmes mis en oeuvre. La facilité avec laquelle ces algorithmes peuvent être implémentés est donc à prendre en compte lorsque les ressources humaines sont limitées. Rappelons que l'ensemble du système SuMo (version 4.4) a été conçu et implémenté intégralement en trois ans par une seule personne, en utilisant le langage Caml.

Chapitre 4

Résultats de comparaisons

De multiples tests ont été effectués avec SuMo, avec la version 4.4 et donnent des résultats encourageants. Cependant, l'étude de leur pertinence biologique demande un certain investissement. Nous présenterons ici les grandes lignes d'un travail effectué en collaboration avec Anne Imberty du CERMAV et publié [23]. Ensuite, plutôt que de présenter un petit nombre d'exemples biologiques, nous présenterons les résultats obtenus à l'issue de la comparaison de tous les sites de fixation de ligands entre eux.

4.1 Famille des lectines de légumineuses

Les *lectines* sont des protéines qui fixent de façon réversible des oligosaccharides [31, 32]. La famille des lectines de légumineuses comporte à l'époque de l'étude (septembre 2001) 106 membres dans la PDB, dont la plupart (94) ont bien une fonction de lectine au niveau d'un site connu mais dont certaines (12) n'ont pas cette propriété [32]. Il s'agit de :

– l'arceline, une protéine de défense (2 structures),
– des inhibiteurs d'α-amylases (2 structures),
– de lectines démétallisées, c'est-à-dire privées des ions Ca^{2+} et Zn^{2+} qui permettent de stabiliser le site lectine (8 structures).

Afin d'examiner les capacités de SuMo, un site 3D de fixation d'oligosaccharide a été délimité à partir de la structure PDB 2PEL. Il a été défini par tous les groupements chimiques possédant au moins un atome distant de moins de 4 Å d'un des atomes du ligand, en l'occurrence un disaccharide, le lactose (LAT). Une illustration de ce site est donnée figure 4.1 page 132. La recherche de sites similaires a été effectuée parmi les 106 structures de la famille. Tout site correspondant avec au moins un des groupements chimiques du site lectine de 2PEL est considéré comme positif. Les résultats de

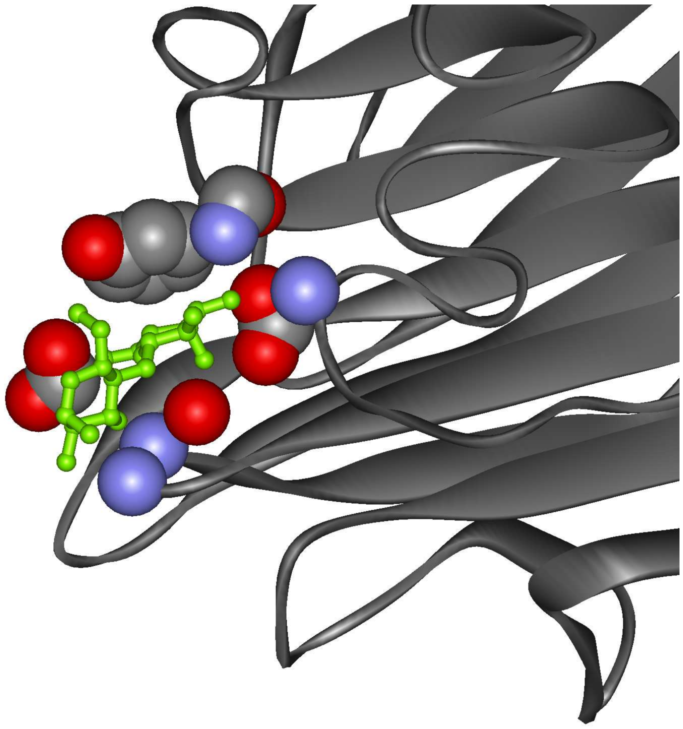


Fig. 4.1 – Illustration du site sélectionné pour le criblage de la famille des lectines de légumineuses. Les atomes en sphères sont ceux qui sont utilisés dans la définition de groupements chimiques sélectionnés. Le ligand est une molécule de lactose, représentée en vert.

ce criblage sont synthétisés sur la figure 4.2. En voici la version chiffrée :
- 90 vrais positifs,
- 12 vrais négatifs,
- 0 faux positif,
- 4 faux négatifs,

soit une réussite de 102/106 par rapport aux données expérimentales, c'est-à-dire 96 %. 4 vraies lectines n'ont pas été détectées par SuMo ; ces résultats sont présentés et discutés en détail dans la publication initiale sur SuMo [23]. Il est important de noter que la version de SuMo utilisée est la version 2.0 datant de juillet 2001. Cette version ne considérait les groupements chimiques par des points sans géométrie particulière, sans distinction entre positions fonctionnelle et physique et dans laquelle les donneurs et accepteurs de liaisons hydrogène libres n'étaient pas pris en compte. De plus, le filtrage final des résultats utilisait le RMSD comme critère de sélection, ce qui a été supprimé à partir de la version 4.0.

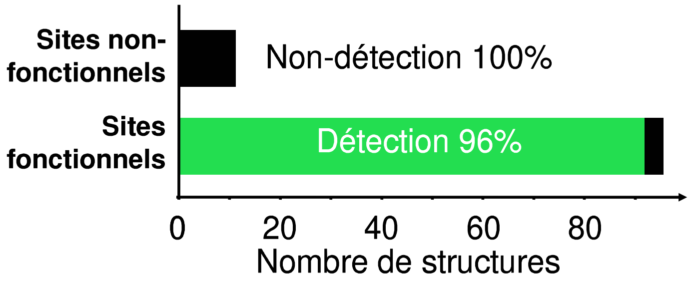


FIG. 4.2 – Résultats de criblage de la famille des lectines de légumineuses par le site lectine de 2PEL. Les sites fonctionnels sont ceux pour lesquels une fixation d'oligosaccharide est observable expérimentalement. Les sites non-fonctionnels sont les autres et proviennent de protéines qui n'ont pas d'activité de lectine connue pour le site considéré dans les conditions considérées.

4.2 Comparaison systématique des sites

4.2.1 Données générales

La comparaison 2 à 2 de tous les sites de fixation de ligands dont le protocole est décrit section 3.5.4 page 80 a été effectuée sur les 11292 sites disponibles au moment du calcul. 1836 ligands différents sont impliqués. Nous avons vu que la fonction spécificité peut être calculée seulement lorsque le site considéré est détecté comme similaire avec au moins 10 autres sites de la base de données appartenant à la même famille. Ceci restreint de façon importante le nombre de sites et de ligands considérés. Ainsi sont considérés 3299 sites répartis sur 37 familles de ligands.

4.2.2 Définition de familles de ligands

Seules 2 familles comprenant plusieurs ligands ont été définies pour l'instant. La première est appelée *monosaccharide*, la seconde *ATP-family*. La définition exacte de ces familles est donnée figure 4.3. Ces définitions ont été réalisées manuellement, en fonction de la ressemblance entre les différents ligands, à l'aide notamment des illustrations données par les serveurs web PDBsum [28, 27] et HIC-Up [24]. De nombreuses familles pourraient judicieusement être définies, mais ceci ne peut pas être réalisé simplement de façon automatique. Rappelons cependant que les familles sont chevauchantes et que tout ligand génère automatiquement une famille à un membre, désignée par le même nom que le ligand.

ATP-family :	ATP, ATP-MG, ATP-MN, ADP, ADP-MG, ADP-MN, GTP, GTP-MG, GTP-MN, GDP, GDP-MG, GDP-MN, GDP-CA, ANP, ANP-MG, ANP-MN, GNP, GNP-MG, GNP-MN, GSP, GSP-MG, GSP-MN, 3AN, 3AT, ABP, ADI, ATR, DAD, DAT, DG3, DGT, DTP, GPX, M7G, MDG, MGP, MGT, SAP
monosaccharide :	2DG, 2FG, ALL, ARA, ARB, BMA, FCA, FCB, FRU, FUC, G2F, G6D, GAL, GLA, GLB, GLC, GLT, LXC, MAN, RAM, RIB, RNS, XUL, XYP, XYS

FIG. 4.3 – Les 2 familles de ligands non-triviales définies actuellement

4.2.3 Résultats

La table 4.1 page 135 donne la liste des 37 familles de ligands pour lesquelles on peut calculer la spécificité pour au moins un site. La spécificité moyenne des sites d'une famille est une moyenne logarithmique, c'est-à-dire

$$\exp \frac{\sum_{i=1}^{n} \log x_i}{n}$$

Cette moyenne est préférable à une moyenne arithmétique lorsque l'on considère que l'espacement entre 2 valeurs x_i et x_j est donné par le rapport x_i/x_j plutôt que $x_i - x_j$. Par exemple, la moyenne logarithmique de l'ensemble $\{0,1; 0,01; 0,001\}$ est 0,01 alors que sa moyenne arithmétique est 0,037. Cette moyenne est calculée uniquement sur les spécificités finies, c'est-à-dire que les spécificités infinies apparaissant dès lors qu'il n'y a aucun faux positif ne sont pas prises en compte dans le calcul de la moyenne.

> *La moyenne séparée des dénominateurs et des numérateurs des éléments exprimés sous forme fractionnaire (voir page 97) serait une solution permettant de prendre en compte également les dénominateurs nuls, mais ceci a pour effet de favoriser les sites les plus redondants. Il faudrait alors appliquer des coefficients pondérateurs de façon à donner autant d'importance à toutes les sites d'une famille.*

Ainsi, la spécificité moyenne présentée pour chaque famille de ligands du tableau est une moyenne de la spécificité apparente des sites de fixation de ligands de la famille considérée. La spécificité apparente d'un site S est donnée par la fonction Φ_S définie page 80. Le tableau s'interprète de la façon suivante :

- La spécificité d'une famille de ligands doit être interprétée *comme un rapport entre le nombre de groupements chimiques vrais positifs et le nombre de groupements chimiques faux positifs, en faisant comme s'il y avait autant de sites internes à la famille que de sites externes à la famille*.
- La spécificité moyenne ne peut pas prendre en compte les spécificités infinies. Le nombre de sites concernés, s'il y en a, est indiqué dans la dernière colonne.

TAB. 4.1: Spécificité moyenne des sites de fixation de ligands. N indique le nombre total de sites pour chaque famille. N_{inf} indique le nombre de sites pour lesquels la spécificité est infinie et qui ne sont donc pas pris en compte dans le calcul de la spécificité moyenne. Résultats obtenus en mars 2003, avec la version 4.4 de SuMo.

Famille de ligands	**Code**	**Spécificité**	N	N_{inf}
HRACEN-4-ONE GUANOSINE DINUCLEOTIDE	PGD	13091,30	2	
HYPOXANTHINE	HPA	1897,69	16	

Famille de ligands	**Code**	**Spécificité**	N	N_{inf}
THYMIDINE-3',5'-DIPHOSPHATE	THP	1308,54	7	
S-ADENOSYL-L-HOMOCYSTEINE	SAH	1186,81	16	
CARBONATE ION	CO3	1067,26	36	
NICOTINAMIDE-ADENINE-DINUCLEOTIDE	NAD	707,13	63	
2'-DEOXYURIDINE 5'-MONOPHOSPHATE	UMP	524,69	37	
BIOTIN	BTN	473,34	19	
2,5-ANHYDROGLUCITOL-1,6-BIPHOSPHATE	AHG	434,05	15	
GUANOSINE-2'-MONOPHOSPHATE	2GP	422,93	35	
NADP NICOTINAMIDE-ADENINE-DINUCLEOTIDE PHOSPHATE	NAP	349,82	32	
MALTOSE	MAL	329,06	11	
monosaccharide		328,72	7	1
FRUCTOSE-6-PHOSPHATE	F6P	242,23	13	
PHOSPHATE ION	PO4	165,53	10	
PHOSPHOAMINOPHOSPHONIC ACID-GUANYLATE ESTER	GNP	145,83	7	
COPPER (II) ION	CU	140,44	108	2
NICKEL (II) ION	NI	114,74	16	2
PROTOPORPHYRIN IX CONTAINING FE	HEM	95,33	382	87
COBALT (II) ION	CO	69,62	3	1
ATP-family		63,89	93	24
FE (II) ION	FE2	63,18	23	

Famille de ligands	**Code**	**Spécificité**	N	N_{inf}
ADENOSINE-5'-DIPHOSPHATE	ADP	54,38	2	
GUANOSINE-5'-DIPHOSPHATE	GDP	49,65	7	
FLAVIN MONONUCLEOTIDE	FMN	38,96	89	43
MANGANESE (II) ION	MN	35,71	123	43
CALCIUM ION	CA	31,65	1293	92
FE (III) ION	FE	29,80	57	11
SULFATE ION	SO4	19,57	14	
ZINC ION	ZN	18,75	683	53
MAGNESIUM ION	MG	12,82	14	1
HEME D	DHE	7,94	13	10
FE2/S2 (INORGANIC) CLUSTER	FES	3,16	13	11
IRON/SULFUR CLUSTER	FS4	1	16	16
ACETATE ION	ACT	1	14	14
DIMETHYL SULFOXIDE	DMS	1	9	9
POTASSIUM ION	K	1	34	34
Moyenne		81,56		

4.2.4 Commentaires

Les résultats présentés auraient pu être synthétisés de multiples façons, notamment en trouvant un moyen de ne pas éliminer les spécificités infinies. Néanmoins, le but de cette approche est simplement de fournir des données de taille raisonnable qui permettent d'avoir une vue d'ensemble de la qualité de SuMo.

4.2.4.1 Représentativité des ligands

Puisque la spécificité apparente d'un site pour une famille de ligands n'est calculée que lorsqu'au moins 10 sites de la même famille ont été détectés comme similaires, de nombreux ligands ne sont pas pris en compte dans le tableau :

- soit parce que ces ligands sont présents moins de 10 fois dans la PDB,
- soit parce que ces ligands ont moins de 10 sites de fixations utilisant un même mode d'ancrage,
- soit parce que SuMo n'est pas capable de détecter des similitudes au sein de la même famille de ligands.

Les ligands peu fréquents mais présentant de multiples variantes dans la PDB peuvent être regroupés en familles. Si les sites de fixation de ligands d'une famille donnée se ressemblent suffisamment, alors le problème de l'abondance de ces sites pourra être réglé, tout en conduisant à une spécificité apparente plus élevée.

4.2.4.2 Problème des spécificités infinies

Certains sites de fixation de ligands d'une famille donnée ont une spécificité infinie, ce qui est très bon signe, mais ceci n'est pas pris en compte dans la moyenne effectuée. Par exemple, les sites de fixation des 4 derniers ligands du tableau ont tous une spécificité apparente infinie, ce qui implique que la spécificité moyenne calculée pour ces ligands n'a pas de sens.

4.2.4.3 Problèmes liés à la structure de la PDB

La PDB est couramment considérée comme redondante par les gens qui s'intéressent à la classification et à l'évolution des protéines. Cependant, nous considérerons que si les structuralistes ont pris la peine de cristalliser plusieurs fois la même protéine, il est fort possible que les structures 3D présentent des différences de structure locales cruciales pour la fonction des protéines.

Nous nous contenterons de remarquer que la distribution des protéines de la PDB n'est pas représentative de l'abondance des protéines dans des conditions physiologiques données, et qu'il est essentiel de faire la distinction entre la notion biochimique de spécificité et celle de spécificité apparente utilisée ici pour avoir une idée de la qualité de SuMo.

Chapitre 5

Bilan

Le système SuMo est, avec le serveur PINTS [44] le seul qui propose de cribler une base de données de sites 3D de fixation de ligands. SuMo, par rapport à PINTS introduit plusieurs concepts novateurs et implémentés avec succès :

- l'abandon de la notion de chaîne principale et de chaînes latérales d'acides aminés,
- la prise en compte des donneurs et accepteurs de liaisons hydrogènes libres une fois les ligands ignorés,
- l'attribution de formes géométriques et de paramètres spécifiques à chaque type de groupements chimiques
- la distinction entre position physique, position fonctionnelle et positions-cibles pour un groupement chimique donné,
- le concept de l'interaction entre un ligand très flexible et une macromolécule de dynamique beaucoup plus lente et non perturbée par la fixation du ligand, par opposition à l'interaction entre 2 macromolécules,
- l'utilisation d'une fonction de score basée sur le modèle protéine rigide/ligand flexible, en remplacement du classique RMSD,
- le notion d'importance accordée à chaque groupement chimique et de rayon d'influence.

Par contre, SuMo ne prétend pas donner une validation statistique de ses résultats. Aucune solution permettant de donner une signification probabiliste aux résultats de SuMo n'a été trouvée pour le moment. D'autre part, les résultats de SuMo permettent de restreindre de façon importante l'espace de recherche, par exemple pour l'identification de sites fixation de ligands sur des protéines, mais ne sont pas un aboutissement. De tels résultats pourront être judicieusement complétés par des simulations de dynamique moléculaire ou d'ancrage de ligands basés sur des modèles physiques empiriques.

Le développement du système SuMo a débuté en janvier 2000, sans la volonté de construire un logiciel aussi complexe et volumineux. En effet, le but initial était simplement de mettre en place un système de comparaison des propriétés de surface des protéines. Cette approche se devait d'être indépendante des notions propres aux protéines que sont la séquence en acides aminés et le repliement de la chaîne principale.

Les premiers résultats convaincants ont été obtenus en mars 2000, dès lors que l'algorithme central de comparaison a été utilisé sur des graphes de triplets de groupements chimiques plutôt que des groupements chimiques seuls. Dans cette première version, les groupements chimiques étaient simplement modélisés par des points correspondant à la moyenne de certains atomes appartenant à un même acide aminé. L'algorithme a été conçu pour identifier des régions similaires de taille quelconque entre des structures de protéines. En effet, il a été conçu de façon symétrique, c'est-à-dire que les eux objets comparés sont du même type : les sous-structures sélectionnées n'ont pas un type particulier. Le criblage de la base de sites de fixation de ligands réalisable actuellement n'est donc pas une recherche de sites parfaits dans une structure requête, mais une recherche de fragments de sites fonctionnels similaires.

Les développements successifs qui ont eu lieu suite aux premiers résultats intéressants obtenus avec SuMo ont eu pour objet autant de rendre l'outil plus facile d'utilisation que d'améliorer la qualité de l'heuristique de comparaison. En effet, SuMo est un système permettant d'analyse de données expérimentales non formel : il permet de sélectionner les données de façon à rendre visible ce qui ne l'est pas pour un observateur humain, selon une heuristique qui se rapproche des intuitions des utilisateurs. C'est pour cela que l'heuristique de SuMo est de plus en plus complexe et sera probablement appelée à évoluer. Cependant, les avantages majeurs d'une approche comme SuMo sur une observation purement humaine des données sont

1. la reproductibilité des résultats,
2. le temps d'analyse.

SuMo se place en amont des simulations de *docking*, c'est-à-dire les calculs basés sur des modèles physiques les plus réalistes possibles qui vont permettre de caractériser l'affinité d'un ligand prédéfini pour un site donné. Dans un processus de *conception de médicament* ou *drug design* à partir de structures 3D connues, SuMo permet d'aider à choisir les sites-cibles, à caractériser partiellement ou totalement la nature des ligands qui sont susceptibles de s'y fixer. La position de SuMo dans un tel processus est schématisée par la figure 5.1 page 141.

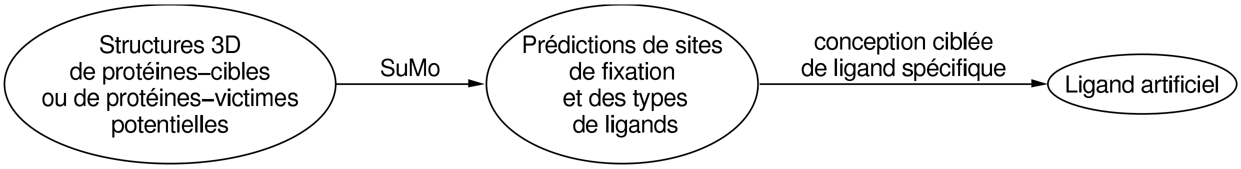


Fig. 5.1 – Situation de SuMo dans un processus de conception de ligand artificiel

Dans les sections suivantes nous discuterons de l'évolution de SuMo, c'est-à-dire de ses possibilités actuelles, des problèmes qui se posent et de la nature des obstacles rencontrés à leur résolution.

5.1 Avenir du logiciel

Le logiciel SuMo n'est actuellement pas distribué mais utilisable directement sur le site web `http://sumo-pbil.ibcp.fr` pour une utilisation non commerciale. Toute décision de restrictions concernant les droits d'accès à ce serveur peut être librement prise par les administrateurs du site et aucune aide aux utilisateurs n'est garantie.

Pour une utilisation intensive ou commerciale de SuMo, les utilisateurs doivent contacter les personnes compétentes. Un brevet est en cours de dépôt par le CNRS. En avril 2003, des négociations entre le CNRS, la société MEDIT et les inventeurs sont en cours pour donner les droits de développement, d'utilisation et de distribution du logiciel à MEDIT. MEDIT a été fondée en 2003 par François Delfaud et ses associés.

À l'avenir, le développement du système SuMo devrait donc être poursuivi. Nous allons examiner différents aspects actuels de SuMo qui imposent des contraintes ou des spécificités pour son maintien et son développement futurs.

5.1.1 Conditions d'utilisation

L'architecture du système SuMo comporte essentiellement 3 niveaux (voir section 3.1.1 page 36). L'utilisation normale et pratique du logiciel se fait grâce à un serveur HTTP. Ceci a l'avantage d'éviter l'installation de logiciels spécifiques sur chaque poste client. En contrepartie, les restrictions d'accès devront être gérées autrement que par la distribution physique du logiciel, par filtrage d'adresses IP ou par la mise en place de mots de passe.

Le système d'exploitation utilisé pour faire tourner le serveur SuMo est GNU/Linux, mais il devrait être directement portable sur tous les systèmes POSIX. Un portage du programme `sumo` pour Cygwin, une émulation d'Unix pour Microsoft Windows a été effectué sans difficultés en 2001. Néanmoins, aucun autre effort de portage n'a été effectué depuis que SuMo utilise une interface web, le rendant accessible depuis n'importe quel système d'exploitation.

5.1.2 Pérennité

Pour pouvoir être installé et tourner correctement, SuMo nécessite à ce jour les composantes suivantes :

- Objective Caml,
- un serveur HTTP,
- des clients HTTP,
- un système d'exploitation de type Unix.

Le système d'exploitation n'est pas une contrainte forte puisqu'Objective Caml peut tourner sur la plupart des systèmes d'exploitation actuels, dont Microsoft Windows et MacOS. L'installation d'Objective Caml nécessite un compilateur C. Les différentes bibliothèques utilisées par SuMo sont écrites en Objective Caml et pourront le cas échéant être modifiées pour s'adapter aux éventuelles modifications du langage Caml. Tous les composants nécessaires à l'installation de SuMo sont *open source*. Leur utilisation est libre de droits sauf MolScript et Raster3D, utilisés uniquement pour générer des images.

5.1.3 Réutilisabilité

5.1.3.1 Heuristiques

Différentes heuristiques développées pour SuMo sont pleinement réutilisables. Elles sont localisées au niveau de modules indépendants et peuvent être appliquées aux types de données les plus polymorphes possibles, du fait du polymorphisme de Caml.

Néanmoins, nombre d'entre elles ont été développées justement parce qu'elles répondaient à un problème particulier posé par SuMo dans des conditions de calcul bien précises. Par exemple, l'heuristique de comparaison de forme locale peut être utilisée avec n'importe quel objet 3D modélisé par des points équivalents à des atomes. Néanmoins, les objets doivent être préalablement superposés selon des paires de points données, et la région d'intérêt doit être définie. Par contre, cette heuristique repose sur un calcul de volume défini de façon spéciale. Cette définition et ce calcul pourront être réutilisés dans un cadre beaucoup plus large, notamment dès que l'on a besoin d'estimer rapidement le volume occupé par des objets modélisés par des points pondérés.

5.1.3.2 Langages

Langages spécialisés La plupart des langages développés pour SuMo n'ont pas pour but d'être réutilisés. En effet, le but d'un langage de programmation est de permettre à un être humain de passer des commandes à un système automatique de la façon la plus facile et la plus sûre possible. C'est pour cela que des langages aussi complexes que celui utilisé pour la définition des groupements chimiques ont été mis en place. Ces langages sont dits spécialisés : ils comportent un grand nombre de mots-clés et de constructions syntaxiques prédéfinies dont le but est de rendre le programme particulièrement lisible.

Langages génériques Deux langages génériques ont été développés dans le cadre de SuMo. Il s'agit de l'extension syntaxique Printfer qui facilite la programmation pour la génération de code HTML, et de la syntaxe arborescente générique développée utilisée notamment pour SuMoQ. Ces deux outils sont pleinement réutilisables dans des programmes et des systèmes n'ayant rien à voir avec la biologie ou la chimie. Cependant, ce ne sont que des syntaxes permettant de mettre en forme facilement des données possédant leur propre sémantique, telles que le HTML ou les requêtes SuMoQ.

5.1.4 Développements futurs

Le développement futur de la méthode et du logiciel nécessitent une bonne compréhension des différentes approches mises en jeu. Il s'agit de différencier les portions de code selon leur rôle et les algorithmes qu'elles implémentent. Nous pouvons grossièrement les classifier en utilisant les critères suivants :
- algorithmes exacts nécessaires ?
- coût des algorithmes
- coût relatif des calculs devant les autres calculs effectués conjointement
- quantité d'affichage réalisé
- mémoire utilisée
- espace-disque utilisé
- temps d'accès aux données
- réutilisabilité nécessaire ?
- longueur du code
- niveau de sécurité souhaité

De nombreux développements de la méthode et de son implémentation peuvent être effectués. Ces développements seront effectués essentiellement à la demande des utilisateurs et en fonction des problèmes rencontrés. Cependant, la conception de base de SuMo impose certains obstacles qui pourront plus judicieusement être franchis par des approches externes et complémentaires de SuMo.

5.2 Limites actuelles du système

Les besoins de SuMo en ressources de calcul sont modérées. Un PC récent est suffisant pour effectuer les calculs les plus courants. Le tableau 5.1 donne l'ordre de grandeur de la durée de différents types de comparaisons effectuées.

Type de comparaisons	**Durée du calcul**
1 structure / 11000 sites	5–30 min
1 site / 20000 structures	30–90 min
11000 sites / 11000 sites	8 jours

TAB. 5.1 – Ordre de grandeur de la durée des comparaisons. Processeur de type Intel Pentium III, 2 GHz.

L'implémentation actuelle de SuMo n'impose donc pas l'utilisation d'un nombre élevé de machines, ni d'un espace de stockage hors du commun, pour l'utilisateur qui s'intéresse à un site ou une protéine particulière. Les comparaisons demandant plus de calculs nécessiteront des moyens de calculs plus importants. En effet, il peut être intéressant d'effectuer toutes les prédictions de sites de fixation de ligands sur chacune des protéines de structure 3D disponibles, et de stocker les résultats obtenus. Ainsi, les résultats seraient directement utilisables par tout utilisateur, sans délai. Si l'on compte 20 min par protéine, l'ensemble de la PDB comptant 20000 structures serait traitée en 278 jours. En utilisant 10 CPU simultanément, le tout serait effectué en 1 mois, une fois pour toutes pour une version donnée de SuMo.

Le temps de calcul n'est donc pas un facteur réellement limitant. Par contre, il est important de s'interroger sur les limites concernant la qualité des résultats de SuMo. SuMo est basé depuis qu'il existe sur les points suivants :

1. représentation d'une structure 3D par un ensemble discret d'objets représentant chacun une propriété localisée,
2. incertitudes sur les positions des objets (groupements chimiques) négligeables devant les distances entre ces positions.

Le premier point impose une contrainte que nous appellerons *problème de la localisation* tandis que le second impose une contrainte que nous appellerons *problème de l'échelle*.

5.2.1 Le problème de la localisation

La représentation des structures 3D de molécules dans SuMo consiste en un ensemble de groupements chimiques. Ces groupements chimiques sont des objets positionnés dans l'espace, de forme et de taille prédéfinie pour un type de groupements chimiques donné. Seuls les groupements chimiques de même type peuvent être comparés et le cas échéant être considérés comme équivalents.

Néanmoins, certaines propriétés qu'il serait intuitivement intéressant de prendre en compte comme l'hydrophobie ne sont pas considérées directement par SuMo. La taille et la forme indéterminées de régions que l'on considérerait comme hydrophobes empêche leur prise en compte en tant que groupements chimiques SuMo.

La solution qui a été adoptée et qui permet partiellement de s'affranchir de cette contrainte est la comparaison de forme de l'environnement (heuristique décrite page 83). Actuellement, aucune propriété particulière n'est projetée au niveau de chaque atome permettant de définir l'environnement, mais cette extension pourrait être envisagée. Ainsi, il suffirait d'accorder à chaque atome de l'environnement un poids ou masse propre (voir page 88) proportionnel à la propriété que l'on souhaite cartographier.

5.2.2 Le problème de l'échelle

Le problème de l'échelle est lié à l'incertitude variable sur la localisation des groupements chimiques de types différents. Plus le groupement chimique modélise une région de grande taille, plus l'incertitude absolue sur sa position fonctionnelle va être élevée. Ceci n'est pas un problème du moment que les distances entre groupements chimiques sont suffisamment grandes devant l'incertitude sur leurs positions.

Cependant, il est intéressant de définir des types de groupements chimiques ayant des rayons d'influence différents. Par exemple :

	rayon d'influence	**incertitude absolue**
donneur de liaison H	faible	faible
phényl	moyen	moyenne

La notion de rayon d'influence est représentée par le coefficient associé à chaque type de groupement chimique SuMo défini page 50.

Le problème de l'échelle apparaît lors du regroupement de groupements chimiques dans des triplets hétérogènes. Une sélection sur la taille des arêtes des triangles physiques est effectuée selon les coefficients des groupements chimiques connectés (voir page 59). Cette sélection permet d'éviter de connecter des groupements chimiques trop proches ou trop éloignés en fonction de leur rayon d'influence. Cependant, le regroupement dans un même triplet de groupements chimiques de rayons d'influence trop élevés est néfaste pour l'obtention de résultats convenables lors de la comparaison, puisque les distances entre groupements chimiques sont comparées. Dans ce cas de figure, les résultats ont tendance à être :

- trop restrictifs pour les rayons d'influence importants,
- trop larges pour les rayons d'influence faibles.

Néanmoins, tous les groupements chimiques doivent être connectés, au moins indirectement, pour qu'ils apparaissent dans la même liste de correspondances finale.

Une solution envisageable est de n'accepter dans un même triplet que des groupements chimiques aux rayons d'influence ou coefficients assez proches. Par exemple, si l'on utilise des groupements chimiques des types A, B et C, dont les rayons d'influence sont a, b, et c de sorte que a et c sont incompatibles, seuls les triplets de types suivants seraient possibles : (A, A, A), (A, A, B), (A, B, B), (B, B, B), (B, B, C), (B, C, C), (C, C, C), (B, B, C). Par contre, si seuls des groupements de type A ou C sont présents dans une région donnée de la molécule, ceux-ci ne seront pas connectés et n'apparaîtront pas dans les même listes de correspondances constituant les résultats de comparaison.

5.3 Conclusion

La figure 5.2 page 147 présente par un schéma simple la place des traitements informatiques tels que ceux effectués par SuMo sur des données expérimentales. L'informatique appliquée aux sciences expérientales permet de transformer des données expérimentales difficiles à interpréter visuellement, en un résultat plus facilement interprétable par le chercheur.

Selon ce schéma, SuMo traite des structures 3D de protéines, difficiles à analyser à l'oeil nu, pour les transformer en un résultat beaucoup plus simple à analyser. Nous pouvons considérer que SuMo joue le rôle d'un canal sensoriel supplémentaire. Il permet de voir ce que l'oeil humain ne peut pas voir avec les représentations usuelles des structures 3D de molécules. Ce canal sensoriel est purement objectif puisque les résultats qu'il renvoie sont parfai-

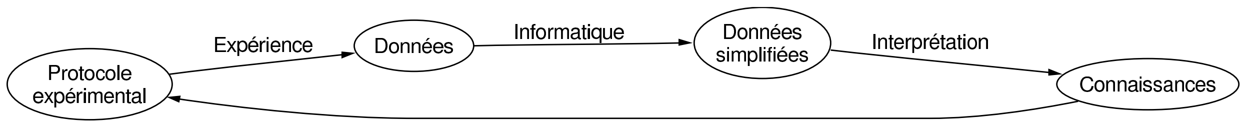


FIG. 5.2 – Informatique et sciences expérimentales

tement reproductibles. Les données reçues finalement par l'oeil humain sont donc moins complètes que les données initiales, mais permettent rapidement d'émettre des hypothèses beaucoup plus élaborées.

Chapitre 6

Participation à d'autres projets

Au cours de ma thèse, j'ai été impliqué dans d'autres projets qui ont conduit à des publications. En voici une brève description.

6.1 Protéine anti-apoptotique Nr-13

Une modélisation moléculaire par analogie de la protéine Nr-13 a fait l'objet de mon stage de magistère durant l'été 1998. La protéine Nr-13 est une protéine anti-apoptotique ou protéine de survie. Suite à cette modélisation, différentes mutations ont été effectuées au sein de cette protéine. Ce travail a été réalisé par l'équipe de Germain Gillet à l'IBCP. Les protocoles et les résultats obtenus ont fait l'objet d'une publication [26].

6.2 Geno3D

Le protocole de modélisation moléculaire par analogie utilisé pour la prédiction de la structure 3D de Nr-13 a été complètement automatisé et mis à disposition des utilisateurs grâce au serveur web Geno3D, accessible à l'adresse suivante :

```
http://geno3d-pbil.ibcp.fr
```

Ce travail a été mené par Christophe Geourjon et fait l'objet d'une publication [13]. Environ 100 requêtes Geno3D sont effectuées par jour, dont 15–20 aboutissent à un modèle 3D.

Soit une séquence de protéine S pour laquelle on souhaite obtenir une prédiction de structure 3D. Le principe de Geno3D est le suivant :

1. recherche de structures 3D connues homologues à S par comparaison des séquences en acides aminés et des prédictions de structures secondaires,
2. génération de contraintes géométriques portant sur les distances et les angles supposés conservés entre les structures connues et S,
3. génération d'un jeu de modèles en utilisant les contraintes géométriques et le logiciel de simulation CNS.

Chapitre 7

Publications

7.1 Articles

- M. Jambon, A. Imberty, G. Deleage, and C. Geourjon. A new bioinformatic approach to detect common 3D sites in protein structures. *Proteins*, 52(2) :137–45, 2003.
- C. Combet, M. Jambon, G. Deleage, and C. Geourjon. Geno3D : automatic comparative molecular modelling of protein. *Bioinformatics*, 18(1) :213–4, 2002.
- P. Lalle, A. Aouacheria, A. Dumont-Miscopein, M. Jambon, S. Venet, H. Bobichon, P. Colas, G. Deleage, C. Geourjon, and G. Gillet. Evidence for crucial electrostatic interactions between Bcl-2 homology domains BH3 and BH4 in the anti-apoptotic Nr-13 protein. *Biochem J*, 368(Pt 1) :213–21, 2002.

7.2 Brevet

- M. Jambon, G. Deléage, and C. Geourjon. Process for identifying similar 3D substructures onto 3D structures and its applications. Dépôt par le CNRS d'une demande de brevet européen numéro 02291407.1 en date du 6 juin 2002.

7.3 Communications orales

- M. Jambon. Communication (1 hour) : Un système de prédiction de sites fonctionnels dans les structures 3d de protéines. Seminar for Hybrigenics, Paris, France, 12 June 2002.

- M. Jambon. Communication (20 min) : Sumo, une méthode de comparaison des propriétés de surface des protéines. 5^{e} journée de l'EDISS (Ecole doctorale), Lyon, France, 29 May 2001.
- M. Jambon. Communication (30 min) : Sumo, une méthode de comparaison des propriétés de surface des protéines. Séminaire de ■ Bio-Informatique Structurale ■, Montpellier, France, 4 October 2001.
- M. Jambon, C. Combet, G. Deléage, and C. Geourjon. Communication (20 min) : Sumo, une méthode de comparaison des propriétés de surface des protéines. 12^{e} rencontres du GGMM (groupe de graphisme et modélisation moléculaire), Nîmes, France, 9-11 May 2001.
- M. Jambon, A. Imberty, G. Deléage, and C. Geourjon. Communication (20 min) : Sumo : a software that detects 3d sites shared by protein structures. JOBIM 2002 (Journées ouvertes biologie informatique mathématiques), Saint Malo, France, 12 June 2002.

7.4 Posters

- C. Combet, M. Jambon, G. Deléage, and C. Geourjon. Poster : Geno3d un serveur web automatique de modélisation moléculaire. 12^{e} rencontres du GGMM (groupe de graphisme et modélisation moléculaire), Nîmes, France, 9-11 May 2001.
- M. Jambon, G. Deléage, and C. Geourjon. Poster : Sumo : logiciel intégré de caractérisation de sites actifs de protéines au service du drug design. 9^{e} carrefours de la fondation Rhône-Alpes Futur, Lyon, France, 17 November 2001.
- M. Jambon, M. Errami, Gilbert Deléage, and C. Geourjon. Poster : Development of the sumo method to detect 3d sites in proteins. 12^{e} rencontres du GGMM (groupe de graphisme et modélisation moléculaire), Nîmes, France, 9-11 May 2001.
- M. Jambon, M. Errami, Gilbert Deléage, and C. Geourjon. Poster : Development of the sumo method to detect 3d sites in proteins. Summer school on protein folding, Cargèse, France, 20-26 May 2001.
- M. Jambon, M. Errami, Gilbert Deléage, and C. Geourjon. Poster : Development of the sumo method to detect 3d sites in proteins. 5^{e} journée de l'EDISS (Ecole doctorale), Lyon, France, 29 May 2001.
- M. Jambon, M. Errami, Gilbert Deléage, and C. Geourjon. Poster : Development of the sumo method to detect 3d sites in proteins. JOBIM 2001 (Journées ouvertes biologie informatique mathématiques) Toulouse, France, 30 May-1 June 2001.

Annexe A

Définition des groupements chimiques

```
(* $Id: amino_acids,v 1.32 2003/02/24 16:08:23 martin Exp $ *)

(* These are the current default definitions for chemical groups in
 * SuMo. The database must be recreated every time the definitions
 * are modified (addition/removal of groups, new identifiers,
 * new parameter, ...)
 *)

(* Recent changes:
   - version 4.4: many changes in parameters
   - version 4.3: new ‘target’ parameter for aromatic
   - version 4.2: important changes in the coefficients
*)

Export (acyl amide aromatic hydroxyl histidine guanidium
        thioether thiol
        delta_plus delta_minus
        (*proline glycine*)
        (*negative positive*));

Not_special ("ALA" "CYS" "ASP" "GLU" "PHE" "GLY" "HIS" "ILE" "LYS" "LEU"
             "MET" "ASN" "PRO" "GLN" "ARG" "SER" "THR" "VAL" "TRP" "TYR"
             "MSE");

Not_special ("HOH");


ala = <<"ALA">>;
arg = <<"ARG">>;
asn = <<"ASN">>;
asp = <<"ASP">>;
```

```
cys = <<"CYS">>;
gln = <<"GLN">>;
glu = <<"GLU">>;
gly = <<"GLY">>;
his = <<"HIS">>;
ile = <<"ILE">>;
leu = <<"LEU">>;
lys = <<"LYS">>;
met = <<"MET">>;
mse = <<"MSE">>;
phe = <<"PHE">>;
pro = <<"PRO">>;
ser = <<"SER">>;
thr = <<"THR">>;
trp = <<"TRP">>;
tyr = <<"TYR">>;
val = <<"VAL">>;

aa = <<"ALA" "CYS" "ASP" "GLU" "PHE" "GLY" "HIS" "ILE" "LYS" "LEU"
       "MET" "MSE" "ASN" "PRO" "GLN" "ARG" "SER" "THR" "VAL" "TRP" "TYR">>;
aaa = <<"ALA" "CYS" "ASP" "GLU" "PHE" "GLY" "HIS" "ILE" "LYS" "LEU"
        "MET" "MSE" "ASN" "GLN" "ARG" "SER" "THR" "VAL" "TRP" "TYR">>;


(***** H-bonding groups *****)
delta_plus {0.6} =

| Delta_plus [backbone] (aaa.<"N">,
                         aa[-1].<"C"> aaa.<"CA">,
                         aaa.<"N">,
                         target = 0.0,
                         functional_shift = 2.8, angle = 140.)

| Delta_plus_plan [d22]
                  (asn.<"ND">, (* position *)
                   asn.<"CG">, (* v_start *)
                   asn.<"ND">, (* v_stop *)
                   asn.<"CG">, (* plan1 *)
                   asn.<"ND">, (* plan2 *)
                   asn.<"OD">, (* plan3 *)
                   60,
                   target = 0.0,
                   functional_shift = 2.8, angle = 140.)

| Delta_plus_plan [d21]
                  (asn.<"ND">,
                   asn.<"CG">,
                   asn.<"ND">,
                   asn.<"CG">,
```

```
                   asn.<"ND">,
                   asn.<"OD">,
                   -60,
                   target = 0.0,
                   functional_shift = 2.8, angle = 140.)

| Delta_plus_plan [e22]
                  (gln.<"NE">,
                   gln.<"CD">,
                   gln.<"NE">,
                   gln.<"CD">,
                   gln.<"NE">,
                   gln.<"OE">,
                   60,
                   target = 0.0,
                   functional_shift = 2.8, angle = 140.)

| Delta_plus_plan [e21]
                  (gln.<"NE">,
                   gln.<"CD">,
                   gln.<"NE">,
                   gln.<"CD">,
                   gln.<"NE">,
                   gln.<"OE">,
                   -60,
                   target = 0.0,
                   functional_shift = 2.8, angle = 140.)

| Delta_plus [e]
             (arg.<"NE">,
              arg.<"CD"> arg.<"CZ">,
              arg.<"NE">,
              target = 0.0,
              functional_shift = 2.8, angle = 140.)

| Delta_plus_plan [h12]
                  (arg.<"NH1">,
                   arg.<"CZ">,
                   arg.<"NH1">,
                   arg.<"CZ">,
                   arg.<"NH1">,
                   arg.<"NH2">,
                   60,
                   target = 0.0,
                   functional_shift = 2.8, angle = 140.)

| Delta_plus_plan [h11]
                  (arg.<"NH1">,
                   arg.<"CZ">,
```

```
                 arg.<"NH1">,
                 arg.<"CZ">,
                 arg.<"NH1">,
                 arg.<"NH2">,
                 -60,
                 target = 0.0,
                 functional_shift = 2.8, angle = 140.)

| Delta_plus_plan [h21]
                (arg.<"NH2">,
                 arg.<"CZ">,
                 arg.<"NH2">,
                 arg.<"CZ">,
                 arg.<"NH1">,
                 arg.<"NH2">,
                 60,
                 target = 0.0,
                 functional_shift = 2.8, angle = 140.)

| Delta_plus_plan [h22]
                (arg.<"NH2">,
                 arg.<"CZ">,
                 arg.<"NH2">,
                 arg.<"CZ">,
                 arg.<"NH1">,
                 arg.<"NH2">,
                 -60,
                 target = 0.0,
                 functional_shift = 2.8, angle = 140.)

| Delta_plus (trp.<"NE1">,
             trp.<"CD1"> trp.<"CE2">,
             trp.<"NE1">,
             target = 0.0,
             functional_shift = 2.8, angle = 140.)

| Delta_plus [d1]
            (his.<"ND1">,
             his.<"CG"> his.<"CE1">,
             his.<"ND1">,
             target = 0.0,
             functional_shift = 2.8, angle = 140.)

| Delta_plus [e2]
            (his.<"NE2">,
             his.<"CD2"> his.<"CE1">,
             his.<"NE2">,
             target = 0.0,
             functional_shift = 2.8, angle = 140.)
```

```
| Delta_plus_multiple (lys.<"NZ">,
                       lys.<"CE">,
                       lys.<"NZ">,
                       3,
                       71,
                       target = 0.0,
                       functional_shift = 2.8, angle = 140.)

| Delta_plus_multiple (ser.<"OG">,
                       ser.<"CB">,
                       ser.<"OG">,
                       1,
                       71,
                       target = 0.0,
                       functional_shift = 2.8, angle = 140.)

| Delta_plus_multiple (thr.<"OG1">,
                       thr.<"CB">,
                       thr.<"OG1">,
                       1,
                       71,
                       target = 0.0,
                       functional_shift = 2.8, angle = 140.)

| Delta_plus_multiple (tyr.<"OH">,
                       tyr.<"CZ">,
                       tyr.<"OH">,
                       1,
                       71,
                       target = 0.0,
                       functional_shift = 2.8, angle = 140.)
;


delta_minus {0.6} =
| Delta_minus [backbone] (aa.<"O">,
                          aa.<"C">,
                          aa.<"O">,
                          1,
                          target = 1.)

| Delta_minus [d1] (asp.<"OD1">,
                    asp.<"CG">,
                    asp.<"OD1">,
                    1,
                    target = 1.)

| Delta_minus [d2] (asp.<"OD2">,
```

```
                     asp.<"CG">,
                     asp.<"OD2">,
                     1,
                     target = 1.)
| Delta_minus [e1] (glu.<"OE1">,
                     glu.<"CD">,
                     glu.<"OE1">,
                     1,
                     target = 1.)
| Delta_minus [e2] (glu.<"OE2">,
                     glu.<"CD">,
                     glu.<"OE2">,
                     1,
                     target = 1.)
| Delta_minus (asn.<"OD1">,
                asn.<"CG">,
                asn.<"OD1">,
                1,
                target = 1.)
| Delta_minus (gln.<"OE1">,
                gln.<"CD">,
                gln.<"OE1">,
                1,
                target = 1.)
| Delta_minus (ser.<"OG">,
                ser.<"CB">,
                ser.<"OG">,
                1,
                target = 1.)
| Delta_minus (thr.<"OG1">,
                thr.<"CB">,
                thr.<"OG1">,
                1,
                target = 1.)
| Delta_minus (tyr.<"OH">,
                tyr.<"CZ">,
                tyr.<"OH">,
                1,
                target = 1.)
;

Hbond (delta_plus, delta_minus);



(***** Other groups *****)

acyl {0.75} =
| Plan (asp.<"CG"> asp.<"OD1"> asp.<"OD2">,
```

```
        asp.<"CG">,
        asp.<"OD1"> asp.<"OD2">,
        asp.<"OD1">,
        angle1 = 60,
        angle2 = 90,
        target = 1.)
| Plan (glu.<"CD"> glu.<"OE1"> glu.<"OE2">,
        glu.<"CD">,
        glu.<"OE1"> glu.<"OE2">,
        glu.<"OE1">,
        angle1 = 60,
        angle2 = 90,
        target = 1.)
;

amide {0.75} =
| Chiral (asn.<"CG"> asn.<"OD1"> asn.<"ND2">,
          asn.<"CG">,
          asn.<"OD1"> asn.<"ND2">,
          asn.<"OD1">,
          angle1 = 60,
          angle2 = 90,
          target = 1.)
| Chiral (gln.<"CD"> gln.<"OE1"> gln.<"NE2">,
          gln.<"CD">,
          gln.<"OE1"> gln.<"NE2">,
          gln.<"OE1">,
          angle1 = 60,
          angle2 = 90,
          target = 1.)
;

positive {0.75} =
| Point (arg.<"NH1"> arg.<"NH2">)
| Point (lys.<"NZ">)
| Point (his.<"NE2">)
;

negative {0.75} =
| Point (asp.<"OD1"> asp.<"OD2">)
| Point (glu.<"OG1"> glu.<"OG2">)
;

aromatic {0.9} =
| Biplan (phe.<"CG"> phe.<"CD1"> phe.<"CD2">
           phe.<"CE1"> phe.<"CE2"> phe.<"CZ">,
          phe.<"CG">,
          phe.<"CE1">,
          phe.<"CE2">,
```

```
          angle = 45, (* was 60 *)
          target = 4.5)

| Biplan (tyr.<"CG"> tyr.<"CD1"> tyr.<"CD2">
           tyr.<"CE1"> tyr.<"CE2"> tyr.<"CZ">,
          tyr.<"CG">,
          tyr.<"CE1">,
          tyr.<"CE2">,
          angle = 45,
          target = 4.5)

| Biplan (his.<"CG"> his.<"ND1"> his.<"CE1"> his.<"CD2"> his.<"NE2">,
          his.<"CG">,
          his.<"ND1">,
          his.<"NE2">,
          angle = 45,
          target = 4.5)

| Biplan [penta]
         (trp.<"CG"> trp.<"CD1"> trp.<"NE1">
           trp.<"CD2"> trp.<"CE2">,
          trp.<"CG">,
          trp.<"CD1">,
          trp.<"CE2">,
          angle = 45,
          target = 4.5)

| Biplan [hexa]
         (trp.<"CD2"> trp.<"CE2"> trp.<"CZ2"> trp.<"CH2">
           trp.<"CE3"> trp.<"CZ3">,
          trp.<"CD2">,
          trp.<"CZ2">,
          trp.<"CZ3">,
          angle = 45,
          target = 4.5)
;

hydroxyl {0.65} =
| Polar (ser.<"OG">, ser.<"CB">, ser.<"OG">, angle = 120, target = 0.)
| Polar (thr.<"OG1">, thr.<"CB">, thr.<"OG1">, angle = 120, target = 0.)
| Polar (tyr.<"OH">, tyr.<"CZ">, tyr.<"OH">, angle = 120, target = 0.)
;

thiol {0.65} = Polar (cys.<"SG">,
               cys.<"CB">,
               cys.<"SG">,
               angle = 120,
               target = 0.)
;
```

```
histidine {0.9} = Chiral (his.<"CG"> his.<"ND1"> his.<"CE1"> his.<"CD2"> his.<"NE2">,
                his.<"CG">,
                his.<"CE1"> his.<"NE2">,
                his.<"CD2">,
                angle1 = 90,
                angle2 = 90,
                target = 0.)
;

proline {0.75} = Point (pro.<"CA"> pro.<"CB"> pro.<"CD"> pro.<"N">);

glycine {0.75} = Polar (gly.<"N">,
                 aa[-1].<"C"> gly.<"CA">,
                 gly.<"N">,
                 angle = 60,
                 target = 0.)
;

guanidium {0.85} = Biplan (arg.<"CZ">,
                    arg.<"NE">,
                    arg.<"NH1">,
                    arg.<"NH2">,
                    angle = 45,
                    target = 2.)
;

thioether {0.65} = Point (met.<"SD">)
                | Point (mse.<"SE">); (* selenium from seleno-methionine *)
```

Annexe B

Aide du serveur web SuMo

Annexe C

Copie des publications

Bibliographie

[1] P.J. Artymiuk, A.R. Poirrette, H.M. Grindley, D.W. Rice, and P. Willett. A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structures. *J Mol Biol*, 243(2) :327–44, 1994.

[2] D.J. Bacon and W.F. Anderson. A fast algorithm for rendering space-filling molecule pictures. *Journal of Molecular Graphics*, 6 :219–220, 1988.

[3] D. Bagley. The great computer language shootout. `http://www.bagley.org/~doug/shootout`

[4] A. Bairoch. Prosite : a dictionary of sites and patterns in proteins. *Nucleic Acids Res*, 19 Suppl :2241–5, 1991. 0305-1048 Journal Article.

[5] W.C. Barker, J.S. Garavelli, H. Huang, P.B. McGarvey, B.C. Orcutt, G.Y. Srinivasarao, C. Xiao, L.S. Yeh, R.S. Ledley, J.F. Janda, F. Pfeiffer, H.W. Mewes, A. Tsugita, and C. Wu. The protein information resource (PIR). *Nucleic Acids Res*, 28(1) :41–4, 2000.

[6] W.C. Barker, J.S. Garavelli, P.B. McGarvey, C.R. Marzec, B.C. Orcutt, G.Y. Srinivasarao, L.S. Yeh, R.S. Ledley, H.W. Mewes, F. Pfeiffer, A. Tsugita, and C. Wu. The PIR-International Protein Sequence Database. *Nucleic Acids Res*, 27(1) :39–43, 1999.

[7] A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S.R. Eddy, S. Griffiths-Jones, K.L. Howe, M. Marshall, and E.L. Sonnhammer. The Pfam protein families database. *Nucleic Acids Res*, 30(1) :276–80, 2002.

[8] H.M. Berman, T. Battistuz, T.N. Bhat, W.F. Bluhm, P.E. Bourne, K. Burkhardt, Z. Feng, G.L. Gilliland, L. Iype, S. Jain, P. Fagan, J. Marvin, D. Padilla, V. Ravichandran, B. Schneider, N. Thanki, H. Weissig, J.D. Westbrook, and C. Zardecki. The Protein Data Bank. *Acta Crystallogr D Biol Crystallogr*, 58(Pt 6 No 1) :899–907, 2002.

[9] C. Blanchet. *Logiciel MPSA et ressources bioinformatiques client-serveur Web dédiés à l'analyse de séquences de protéine.* PhD thesis, Université Claude Bernard, Lyon 1, 1999.

[10] B. Boeckmann, A. Bairoch, R. Apweiler, M.C. Blatter, A. Estreicher, E. Gasteiger, M.J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res*, 31(1) :365–70, 2003.

[11] E. Chailloux, P. Manoury, and B. Pagano. *Développement d'applications avec Objective Caml*. O'Reilly, 2000.

[12] C. Combet, C. Blanchet, C. Geourjon, and G. Deléage. Nps@ : Network protein sequence analysis. *TIBS*, 25 :147–150, 2000.

[13] C. Combet, M. Jambon, G. Deleage, and C. Geourjon. Geno3D : automatic comparative molecular modelling of protein. *Bioinformatics*, 18(1) :213–4, 2002.

[14] F. Corpet. Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res*, 16(22) :10881–90, 1988.

[15] D. de Rauglaudre. Camlp4 : Pre-processor-pretty-printer for objective caml. `http://caml.inria.fr/camlp4/`

[16] D. de Rauglaudre. Ioxml. `http://cristal.inria.fr/~ddr/IoXML`

[17] Altschul S. F., Madden T. L., Schaffer A. A., Zhang J., Zhang Z., Miller W., and Lipman D. J. Gapped blast and psi-blast : a new generation of protein database search programs. *Nucleic Acids Res*, 25(17) :3389–402, 1997. 0305-1048 Journal Article Review Review, Tutorial.

[18] D. Fischer, O. Bachar, R. Nussinov, and H. Wolfson. An efficient automated computer vision based technique for detection of three dimensional structural motifs in proteins. *J Biomol Struct Dyn*, 9(4) :769–89, 1992. 0739-1102 Journal Article.

[19] M. Gribskov, A. D. McLachlan, and D. Eisenberg. Profile analysis : detection of distantly related proteins. *Proc Natl Acad Sci U S A*, 84(13) :4355–8, 1987. 0027-8424 Journal Article.

[20] L. Holm and C. Sander. The fssp database : fold classification based on structure-structure alignment of proteins. *Nucleic Acids Res*, 24(1) :206–9, 1996. 0305-1048 Journal Article.

[21] L. Holm and C. Sander. Touring protein fold space with dali/fssp. *Nucleic Acids Res*, 26(1) :316–9, 1998. 0305-1048 Journal Article.

[22] M. Jambon, G. Deléage, and C. Geourjon. Process for identifying similar 3D substructures onto 3D structures and its applications. Dépôt par le CNRS d'une demande de brevet européen numéro 02291407.1 en date du 6 juin 2002.

[23] M. Jambon, A. Imberty, G. Deleage, and C. Geourjon. A new bio-informatic approach to detect common 3D sites in protein structures. *Proteins*, 52(2) :137–45, 2003.

[24] G.J. Kleywegt. Recognition of spatial motifs in protein structures. *J Mol Biol*, 285(4) :1887–97, 1999.

[25] P.J. Kraulis. Molscript : A program to produce both detailed and schematic plots of protein structures. *Journal of Applied Crystallography*, 24 :946–50, 1991.

[26] P. Lalle, A. Aouacheria, A. Dumont-Miscopein, M. Jambon, S. Venet, H. Bobichon, P. Colas, G. Deleage, C. Geourjon, and G. Gillet. Evidence for crucial electrostatic interactions between Bcl-2 homology domains BH3 and BH4 in the anti-apoptotic Nr-13 protein. *Biochem J*, 368(Pt 1) :213–21, 2002.

[27] R.A. Laskowski. PDBsum : summaries and analyses of PDB structures. *Nucleic Acids Res*, 29(1) :221–2, 2001.

[28] R.A. Laskowski, E.G. Hutchinson, A.D. Michie, A.C. Wallace, M.L. Jones, and J.M. Thornton. PDBsum : a Web-based database of summaries and analyses of all PDB structures. *Trends Biochem Sci*, 22(12) :488–90, 1997.

[29] X. Leroy, J. Vouillon, D. Doligez, and coworkers. The objective caml system. software and documentation on the web, `http://caml.inria.fr/ocaml/`, 1996.

[30] S. L. Lin, R. Nussinov, D. Fischer, and H. J. Wolfson. Molecular surface representations by sparse critical points. *Proteins*, 18(1) :94–101, 1994. 0887-3585 Journal Article.

[31] H. Lis and N. Sharon. Lectins : carbohydrate-specific proteins that mediate cellular recognition. *Chem. Rev.*, 98 :637–674, 1998.

[32] R. Loris, T. Hamelryck, J. Bouckaert, and L. Wyns. Legume lectin structure. *Biochim. Biophys. Acta*, pages 9–36, 1998.

[33] E.A. Merritt and D.J. Bacon. Raster3d : Photorealistic molecular graphics. *Methods in Enzymology*, 277 :505–24, 1997.

[34] E.A. Merritt and M.E.P. Murphy. Raster3d version 2.0 : A program for photorealistic molecular graphics. *Acta Cryst.*, D50 :869–73, 1994.

[35] M. Mottl. Ocamlmakefile : Automated compilation of complex ocaml-projects. `http://www.ai.univie.ac.at/~markus/home/ocaml_sources.html`

[36] C. Notredame, D.G. Higgins, and J. Heringa. T-Coffee : A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*, 302(1) :205–17, 2000.

[37] W. R. Pearson. Rapid and sensitive sequence comparison with fastp and fasta. *Methods Enzymol*, 183 :63–98, 1990. 0076-6879 Journal Article.

[38] R. Preissner, A. Goede, and C. Frommel. Dictionary of interfaces in proteins (dip). data bank of complementary molecular surface patches. *J Mol Biol*, 280(3) :535–50, 1998. 0022-2836 Journal Article.

[39] R. Preissner, A. Goede, and C. Frommel. Homonyms and synonyms in the dictionary of interfaces in proteins (dip). *Bioinformatics*, 15(10) :832–6, 1999. 1367-4803 Journal Article.

[40] R. B. Russell. Detection of protein three-dimensional side-chain patterns : new examples of convergent evolution. *J Mol Biol*, 279(5) :1211–27, 1998. 0022-2836 Journal Article.

[41] B. Sandak, R. Nussinov, and H. J. Wolfson. An automated computer vision and robotics-based technique for 3-d flexible biomolecular docking and matching. *Comput Appl Biosci*, 11(1) :87–99, 1995. 0266-7061 Journal Article.

[42] I.N. Shindyalov and P.E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, 11(9) :739–47, 1998.

[43] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1) :195–7, 1981.

[44] A. Stark, S. Sunyaev, and R.B. Russell. A model for statistical significance of local similarities in structure. *J Mol Biol*, 326(5) :1307–16, 2003.

[45] G. Stolpmann. Ocamlnet. `http://sourceforge.net/projects/ocamlnet`

[46] Thomas Cormen, Charles Leiserson, and Ronald Rivest. *Introduction à l'algorithmique*. Dunod, 1994.

[47] J. D. Thompson, D. G. Higgins, and T. J. Gibson. Clustal w : improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22) :4673–80, 1994. 0305-1048 Journal Article.

[48] A. Via, F. Ferre, B. Brannetti, and M. Helmer-Citterich. Protein surface similarities : a survey of methods to describe and compare protein surfaces. *Cell Mol Life Sci*, 57(13-14) :1970–7, 2000. 1420-682x Journal Article Review Review, Tutorial.

[49] D. Voet and J. Voet. *Biochemistry*. Wiley and Sons, 1997. pages 389-400.

[50] A. C. Wallace, N. Borkakoti, and J. M. Thornton. Tess : a geometric hashing algorithm for deriving 3d coordinate templates for searching

structural databases. application to enzyme active sites. *Protein Sci*, 6(11) :2308–23, 1997. 0961-8368 Journal Article.

[51] A. C. Wallace, R. A. Laskowski, and J. M. Thornton. Derivation of 3d coordinate templates for searching structural databases : application to ser-his-asp catalytic triads in the serine proteinases and lipases. *Protein Sci*, 5(6) :1001–13, 1996. 0961-8368 Journal Article.

[52] A. C. Wallace and J. M. Thornton. Procat, a database of 3d enzyme active site templates `http://biochem.ucl.ac.uk/bsm/procat/procat.html`, 1996.

[53] P. Weis and X. Leroy. *Le langage Caml*. Dunod, 1999.

[54] Xavier Leroy, Damien Doligez, Jacques Garrigue, Didier Rémy, and Jérôme Vouillon. *The Objective Caml system*. Institut National de Recherche en Informatique et en Automatique, release 3.06 edition.

[55] E.M. Zdobnov and R. Apweiler. InterProScan–an integration platform for the signature-recognition methods in InterPro. *Bioinformatics*, 17(9) :847–8, 2001.

Index