CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS

CENTRE RÉGIONAL ASSOCIÉ DE PAYS DE LOIRE

MÉMOIRE

présenté en vue d'obtenir

le DIPLÔME d'ingénieur CNAM

SPÉCIALITÉ : INFORMATIQUE

OPTION : Réseaux, systèmes et multimédia (IRSM)

par

Cédric Ramassamy

Quantification Vectorielle Algébrique et Arborescente pour l'analyse de données moléculaires

Soutenu le : 16 décembre 2011

JURY :

Présidente : Isabelle Métais (présidente du jury, professeur Cnam Paris)
Membres : Henri Briand (professeur Ecole Polytechnique Nantes)
Vincent Ricordel (LUNAM Université, Université de Nantes)
Bogdan Cramariuc (IT Center for Science and Technology, Bucarest, Roumanie)
Oana Cramariuc (Dept. of Physics, Tampere University of Technology, Finlande)

Remerciements

«Soyons reconnaissants aux personnes qui nous donnent du bonheur, elles sont les charmants jardiniers par qui nos âmes sont fleuries.[...]»
Marcel Proust dans *Les plaisirs et les jours.*

«Expérimenter, c'est imaginer.»
Friedrich Nietzsche dans *Aurore.*

Ce mémoire clôt sept années d'études, et avant tout, il me faut remercier les femmes de ma vie ; Mon épouse, Thurianne pour son amour et son soutien ainsi que ma mère pour ce *quelque chose* qui a changé... Sans elles, ce but que j'atteins enfin n'aurait été qu'un doux rêve.

Je tiens à exprimer des remerciements chaleureux

À Vincent Ricordel d'avoir accepté d'être mon encadrant lors de ce mémoire. Son enseignement, ses conseils, sa rigueur intellectuelle et son amitié ont été pour moi une véritable source d'inspiration tout au long de ces neufs mois.

À Oana et Bogdan Cramariuc d'avoir accepté de considérer mon travail, la modeste contribution à un projet éminemment spécialisé. Leur expertise et leur amitité furent une aide très précieuse.

Aux membres de l'équipe IVC de l'IRCCyN d'avoir accepté de m'accueillir au sein de leur laboratoire. Je remercie les doctorants et toutes les personnes qui, même indirectement, ont contribué à la rédaction de ce mémoire.

À Erwann Lastennet, de qui je ne sais problablement pas la moitié des exploits qu'il a réalisés pour nous les *auditeurs CNAM* afin d'assurer la continuité du cursus d'ingénieur CNAM.

Liste des abréviations

CP	point critique
DFT	fonctionnelle de la densité
EDM	carte de densité électronique
EQM	erreur quadratique moyenne
GF	groupe fonctionnel
IRCCyN	Institut de Recherche en Communications et Cybernétique de Nantes
IVC	Images et VidéoCommunications
LBG	algorithme LBG (Linde, Buzo, Gray)
LCAO	combinaison linéaire des orbitales atomiques
MEM	mécanique moléculaire
MEQ	mécanique quantique
MM	modélisation moléculaire
PDB	fichier de données moléculaire issue de la Protein Data Bank
QS	quantification scalaire
QV	quantification vectorielle
QVA	quantification vectorielle algébrique
QVAr	quantification vectorielle arborescente
RAM	Random-access memory
QVAA	quantification vectorielle algébrique et arborescente
UML	Unified Modeling Language
VU	volume unitaire
XP	eXtreme Programming

Table des matières

Introduction

Ce sujet est issu des recherches de Vincent Ricordel, Oana et Bogdan Cramariuc. Le projet induit la fusion de deux domaines de recherches qui sont la modélisation moléculaire (MM) et le traitement du signal associé à l'analyse de données.

Les concepts du traitement du signal s'articulent plus précisément autour de méthodes d'analyse, de manipulation et de présentation de l'information. Durant les dernières décennies, de nombreuses approches issues du traitement du signal ont été employées au-delà de leurs domaines d'applications premiers que sont la transmission et les télécommunications. Cette ouverture a contribué aux récentes avancées de la biologie, de la biochimie et de la biomédecine.
Une cause sous-jacente de ce développement est à chercher dans l'augmentation exponentielle du volume de données numériques obtenues à travers la modélisation et les techniques de simulation ou par l'emploi d'outils modernes de recherche expérimental à très haut-débit [Lew03]. Ces outils sont appliqués à l'ADN, les protéines, les micro-puces à anticorps. Une autre cause réside dans la nécessité d'une vision plus large et plus cohérente de la *Nature* à l'aide de modèles systémiques qui décrivent les fonctionnements des organismes vivants sans faire abstraction des processus individuels.

Ce mémoire concernera précisément le domaine de la modélisation moléculaire (MM). Le projet consiste à traiter des données représentant un grand nombre d'informations issues de la synthèse de la densité électronique [Pau08a] à l'échelle moléculaire.
Les informations modélisent des structures de molécules. Il faut accéder à l'organisation de ces données afin d'y détecter des singularités autour des points critiques tels que les points cols (*saddle*). Malheureusement, le volume de donnée est bien trop conséquent pour être exploités directement. Par exemple, les données issues de la modélisation d'une protéine se chiffrent

en téraoctets !
La puissance de calcul des ordinateurs actuels a atteint un niveau acceptable lorsqu'il s'agit de traiter un très grand volume d'atomes. Néanmoins, le domaine de la MM cherche toujours une méthode efficace pour détecter les groupes fonctionnels chimiques composant les molécules [Leh01]. En classification de protéines, par exemple, la *Recherche* s'intéresse tout particulièrement à la caractérisation détaillée de la surface et à la topologie des macromolécules [BMLV08] telles que les bases de l'ADN. Aujourd'hui, la synthèse de médicaments vise à maîtriser les caractéristiques tridimensionnelles des molécules sachant que la difficulté première est la capacité à détecter les groupes fonctionnels au sein des macromolécules. Or les densités électroniques des grands systèmes moléculaires décrivent l'agencement de 10 000 à 50 000 atomes [DAPMGC02]. Leherte [Leh01] s'attache à comparer trois méthodes de traitement de carte de densité électronique (EDM). Le but est d'utiliser le concept des points critiques pour déterminer des *motifs* uniques représentants des groupes fonctionnels chimiques. Ces motifs détectables dans d'autres molécules doivent permettre d'identifier des groupes fonctionnels identiques.

En traitement multimédia, on utilise typiquement la quantification vectorielle (QV) [GG92] au sein de schémas de codage-décodage à des fins de représentation et de compression de l'information. La quantification vectorielle est un outil du traitement de signal qui a révolutionné la compression de données multimédia en simplifiant la représentation des données source et en permettant de hiérarchiser l'information contenue [Ric96]. C'est essentiellement les démarches de simplification et de hiérarchisation [Pau08b] qui concernent le projet.
Les résultats issus des recherches sur la quantification vectorielle avec réseaux réguliers de points [Ric96] peuvent aussi être utilisés comme outil de simplification et d'analyse des données moléculaires. Nous emploierons la quantification vectorielle comme une première étape dans le traitement des densités électroniques moléculaire dans une optique de simplification, d'analyse des similarités moléculaires, d'exploration topologique et de visualisation.
Ce projet doit pouvoir permettre de déterminer des critères (Métriques, Arbres de données, Iso-courbes, $3D$, etc.) de représentation simplifiée des données moléculaires. Les travaux de [Leh01] ont jalonné notre méthode selon un schéma : quantification, détection et caractérisation de points critiques pour identifier des caractéristiques des groupes fonctionnels.
Indépendamment, QV et MM font l'objet de nombreuses publications, mais la fusion des deux domaines est encore à l'état embryonnaire. Des recherches [DAPMGC02] s'attachent à la QV à l'aide de classifieurs. On note les travaux de [Leh01] et [BCCSX05] qui proposent des méthodes de détection de points

critiques à base d'ondelettes.

Á ce jour, la question a fait l'objet de trois projets d'études par des élèves de Polytech'Nantes. Ces travaux ont produits quelques algorithmes et, plus généralement, ont permis de tracer les grandes lignes du projet [Luo10; DM10; Cha11]

Le mémoire est organisé comme suit :

- dans le premier chapitre nous donnons un état de l'art des différentes notions abordées dans ce document. Pour comprendre la nature de nos données source, nous commençons par définir la modélisation moléculaire (MM) en traitant successivement tous les concepts qui s'y rapporte, de la chimie organique à la chimie quantique. Nous présentons ensuite la quantification vectorielle (QV) et particulièrement la quantification vectorielle algébrique et arborescente (QVAA) : la méthode de traitement de nos données. Puis nous terminons par expliquer la méthode d'analyse géométrique à base de points critiques ;

- la modélisation et la réalisation de notre méthode est traitée dans le chapitre 2. Nous établissons un modèle *objet* du projet puis présentons les outils de développement et de test. Ce chapitre s'achève par une explication détaillée de l'implémentation réalisée ;

- le troisième chapitre présente les résultats. Nous y abordons une description succinte de nos données d'entrée. Nous analysons ensuite le résultat post-traitement d'un point de vue visuel puis mathématique ;

- La gestion de projet fait l'objet du chapitre 4, nous exposons notamment l'application des méthodes agiles et particulièrement de l'eXtreme Programming (XP) à notre projet ;

- nous achevons ce document par une conclusion générale résumant l'essentiel de nos travaux. Nous abordons des perspectives de travail pour améliorer nos résultats, et pour envisager de nouvelles stratégies de quantification qui permettront de parfaire la simplification et l'analyse de notre signal moléculaire.

Chapitre 1

État de l'art

1.1 La Modélisation Moléculaire

1.1.1 Introduction

Ce chapitre se propose de présenter les concepts de la modélisation moléculaire (MM) car les données de ce mémoire sont des cartes de densité électronique (EDM) qui représentent des données moléculaires issues des logiciels standards du monde de la MM. Plus généralement, une vision globale des enjeux sous-jacents de cette discipline motive la recherche d'une nouvelle approche en terme de représentation et d'analyse des propriétés chimiques de molécules.
La modélisation moléculaire (MM)[1] est une discipline à cheval sur les domaines de la chimie, la physique quantique, la biologie et l'informatique. Il s'agit d'un ensemble d'outils qui offre des techniques d'études de la structure et du comportement des molécules [Pau08a]. La compréhension et la prévision des phénomènes sous-jacents aux transformations chimiques, physiques, et biologiques sont étroitement liées à la connaissance de la structure tri-dimensionelle des molécules. La MM, grâce à l'informatique, permet de représenter de façon réaliste les structures moléculaires en termes de géométrie et de thermodynamique. La modélisation d'une molécule consiste alors à décrire mathématiquement la position dans l'espace de ses atomes ainsi que déterminer l'énergie de la molécule dans son ensemble. On remarque qu'un modèle réaliste d'une molécule dégage la plus basse "*énergie*". La MM se com-

1. en anglais : Computational Chemistry

pose de deux disciplines : tout d'abord, la mécanique moléculaire (MEM), basée sur la mécanique classique qui utilise les champs de force, les calculs sont effectués sur des petites molécules d'une centaine d'atomes. Ensuite, la mécanique quantique (MEQ) qui s'appuie sur l'**équation de Schrödinger**, dont le principe consiste à exprimer les orbitales moléculaires [JV91] selon différentes méthodes mathématiques différentielles.

Le propos de ce chapitre traitera essentiellement de la mécanique quantique basée sur la méthode de la fonctionnelle de la densité (DFT)[AM00].
Avant d'évoquer les fondements de la modélisation moléculaire, il est important de la resituer dans le contexte de la chimie organique qui en est la principale motivation.

1.1.2 La Chimie Organique

La chimie organique ou *chimie du carbone* est la chimie des organismes vivants. À l'inverse, les molécules minérales, molécules provenant du monde minéral (roche, atmosphère) sont, à quelques exceptions près, dépourvues d'atome de carbone. Ainsi, les molécules organiques contiennent toujours du carbone. Elles proviennent essentiellement des êtres vivants (animaux, végétaux) et sont des combinaisons de peu d'atomes tels que le Carbone (C), l'Hydrogène (H), l'Oxygène (O), l'Azote (N) et plus rarement le Phosphore (P) et le Soufre (S). Les acides aminés, telle que la glycine (figure : 1.1), sont de bons exemples de molécules organiques simples. On a montré ainsi que dans l'univers, les molécules organiques sont à l'origine de l'apparition de la vie et sont donc les molécules fondamentales du monde vivant [MG01].
Les cellules des organismes vivants sont formées de molécules. Celles-ci, bien qu'issues d'un ensemble limité d'atomes, sont extrêmement variées ce qui explique la très grande complexité du fonctionnement des êtres vivants. Le Carbone, commun à toutes les molécules organiques, a la propriété d'être très léger, ce qui, même pour une très grande molécule de l'ordre du millier d'atomes pour les protéines, en limite la masse moléculaire. La figure 1.2 montre la complexité d'une molécule d'insuline humaine.
De plus, le carbone se lie facilement à lui-même par liaison covalente normale parfaite, ce qui lui confère la faculté de composer des molécules avec de longues chaînes de carbone. Par exemple, la figure 1.3 montre une molécule très utilisée dans le domaine des nanotechnologies et exclusivement composée de 60 atomes de carbone : le fullerène (C_{60}).
Par ailleurs, Mercier [MG01] expliquent que le carbone présente la propriété de pouvoir s'associer à de nombreux éléments essentiellement non métalliques

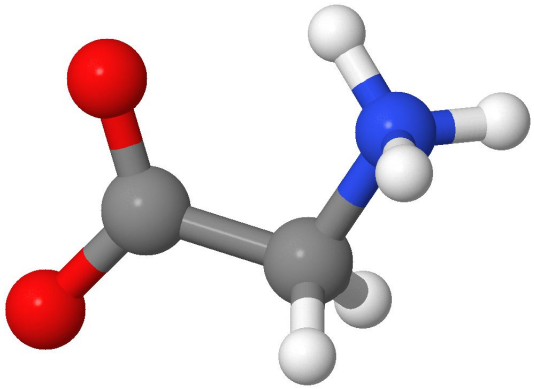


FIGURE 1.1 – Molécule de la Glycine, un Acide Aminé (10 Atomes).

car il est dit *tétravalent*, il est de valence 4 (4 électrons échangés lors d'une liaison ionique).
Ainsi les différentes propriétés du carbone permettent un ensemble de combinaisons quasi-infini de molécules organiques d'où la diversité et la complexité du monde vivant.
Ces combinaisons se matérialisent par des liaisions chimiques qu'il convient de caractériser pour identifier les propriétés chimiques des molécules.

1.1.3 La théorie électronique de la liaison chimique

La théorie électronique de la liaison chimique énonce que les électrons impliqués dans la liaison chimique appartiennent généralement à la couche électronique externe de l'atome qui est la plus faiblement liée au noyau. Ces électrons de *valence* interviennent pour stabiliser la structure électronique de l'atome qui, en partageant ses électrons, en absorbe ou en perd. Cette liaison dite *covalente* est la principale liaison chimique utilisée en chimie organique. L'hydrogène (H) (figure 1.4) ne possède qu'un seul électron et qu'une seule couche électronique qui est dite *saturée* dès qu'elle dispose de deux électrons. Il faut alors, par exemple, lui adjoindre un autre atome d'hydrogène pour que leur couche électronique respective soit stabilisée.

1.1.4 Structure des molécules organiques

Analyser des molécules, pour ce mémoire, revient à modéliser la disposition spatiale des atomes dans leur représentations. [JV91] expliquent

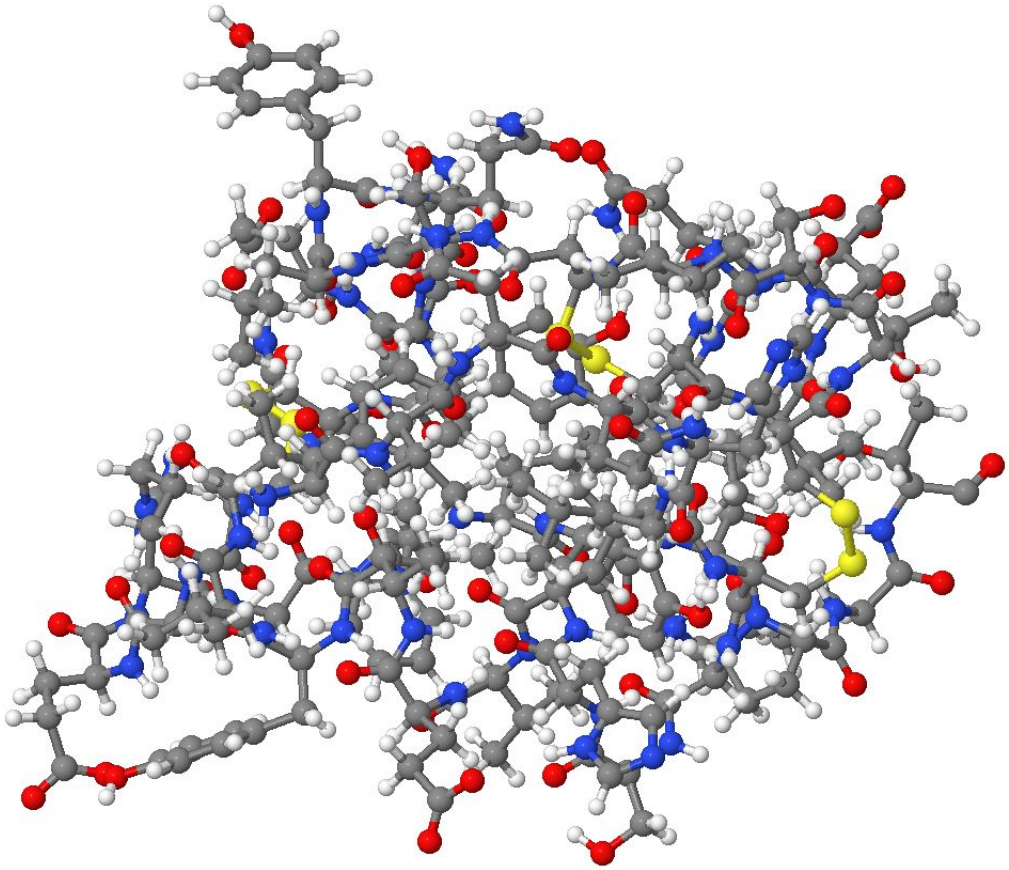


FIGURE 1.2 – Molécule de l'Insuline Humaine (8624 Atomes).

qu'il existe plusieurs manières de représenter les molécules organiques. Tout d'abord les **formules brutes** qui mentionnent la nature et le nombre d'atomes qui composent la molécule sans pour autant donner d'informations spatiales ni renseigner l'enchaînement. Par exemple, la *formule brute* du méthane s'écrit CH_4. Les **formules développées planes** ajoutent l'information concernant l'enchaînement des atomes. Cette représentation est très utilisée bien qu'elle ne donne aucune information spatiale ni sur l'aspect électronique. La figure 1.5 montre la *formule développée plane* du méthane. De nombreuses molécules organiques sont constituées d'une chaîne carbonée principale où sont attachés d'autres groupes de carbones ou des **groupes fonctionnels (GF)**. La **chaîne carbonée** est souvent représentée par une droite alors qu'un zigzag serait plus représentatif de la réalité. Ce diagramme respecte la *tétra-valence* des atomes de carbone qui sont reliés entre eux par de simples,

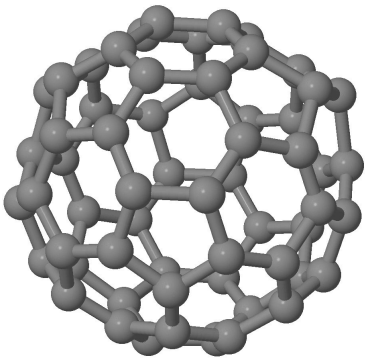


FIGURE 1.3 – Molécule du Fullerène (C_{60}) (60 atomes de carbone).

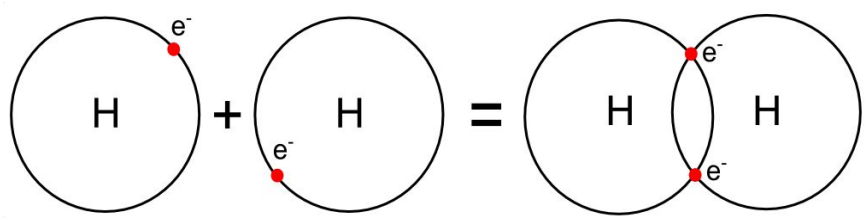


FIGURE 1.4 – Synthèse du dihydrogène (H_2) (Valence = 1, stable à 2).

doubles ou triples liaisons chimiques. On différencie, par ailleurs, les molécules **cycliques** telles que le benzène (figure 1.6) et **acycliques** (figure 1.7) telle que le "1-butène". Le chiffre **1** indique la position du carbone contenant la double liaison.

1.1.5 Les Groupes fonctionnels

Dans le cadre de ce mémoire, à longue échéance, comme expliqué dans [Leh01], la finalité consistera à détecter les groupes fonctionnels qui composent des molécules données. [JV91] enseignent qu'une molécule organique est en général composée d'un squelette constitué de liaisons carbone-carbone simples, qui souvent est chimiquement inerte et sur lequel on peut fixer des **groupe fonctionnel**. Ces derniers ont le contrôle de la réactivité chimique et des propriétés de la molécule. Un tableau des principaux groupes fonctionnels est disponible dans [JV91]. Néanmoins, on peut citer des groupes fonctionnels simples tel que le groupe *hydroxyle* (-OH) qui caractérise les alcools (figure 1.8) ou encore le groupe *carboxyle* (C=O-OH) qui qualifie les acides

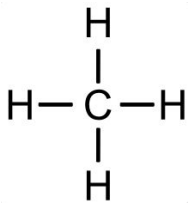


FIGURE 1.5 – Formule développée plane du méthane.

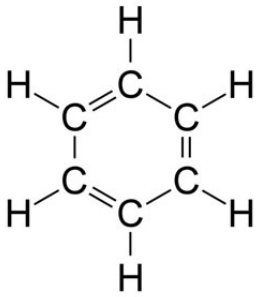


FIGURE 1.6 – Formule développée plane du benzène (C_6H_6).

carboxyliques, par exemple, dans l'acide hexanoïque ($C_6H_{12}O_2$, figure 1.9). D'ailleurs, la condensation d'un alcool sur un acide carboxylique s'appelle *l'estérification* qui par renversement conduit à la *saponification* (la synthèse du savon).
Lorsqu'une molécule possède plusieurs groupe fonctionnel, ses propriétés dépendent alors à la fois de chacun des groupe fonctionnel existants ainsi que de leur position spatiale. Il apparaît alors que la fonction de la molécule vient aussi de sa géométrie. Ainsi, la famille de *hydroxyacides* contient un groupe *hydroxyle* et un groupe *carboxyle*, cela lui permet de se lier autant

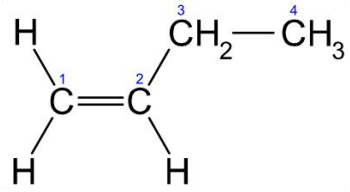


FIGURE 1.7 – Formule développée plane du 1-butène (C_4H_8).

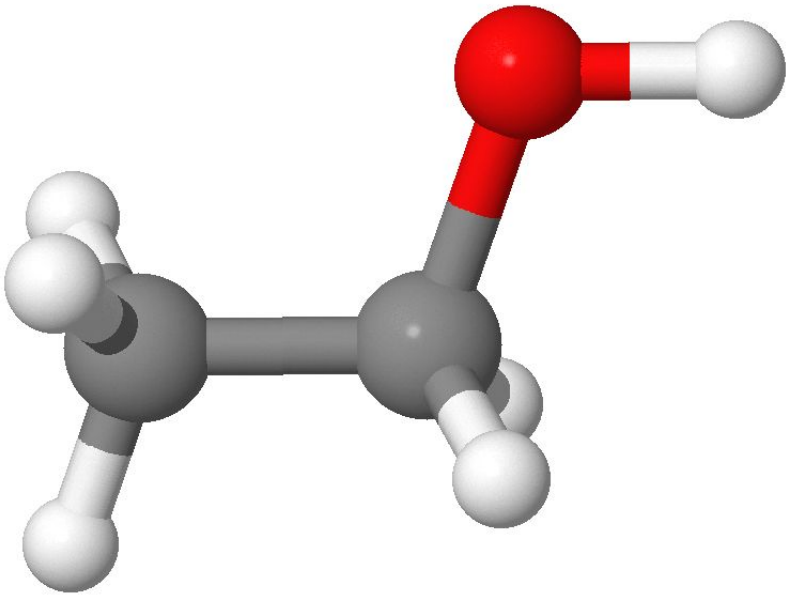


FIGURE 1.8 – Molécule de l'éthanol (C_2H_6O) un alcool.

aux acides qu'aux alcools. On peut évoquer dans ce cas l'*acide tartrique* du raisin, l'*acide citrique* des agrumes.
Certains composés organiques disposent de nombreux groupe fonctionnel différents et sont alors caractérisés par une architecture très complexe difficile à appréhender. L'exemple de la pénicilline G (figure 1.10) nous montre une molécule qui porte des groupe fonctionnels (GFs) *amides* (O=C-NH-), *carbonyle* (O=C=), *thioether* (-S-) et *carboxyle* (C=O-OH). En reprenant la molécule d'insuline humaine de la figure 1.2, entre autres, on distingue très clairement aux extrémités des groupe fonctionnel *hydroxyle* (des groupes isolés formés d'un atome d'oxygène en rouge lié à un atome d'hydrogène en blanc).

1.1.6 Isomérie

Nous avons vu que les *formules brutes* ne permettent pas de caractériser la disposition spatiale d'une molécule. Ainsi, toutes les molécules qui peuvent être représentées par la même formule brute sont des **isomères**. Une molécule simple comme l'éthanol (figure 1.8) dont la formule brute est C_2H_6O peut aussi se représenter sous la forme de la figure 1.11, il s'agit du Diméthyléther, un *isomère structural* de l'éthanol car l'enchaînement des atomes est différent.

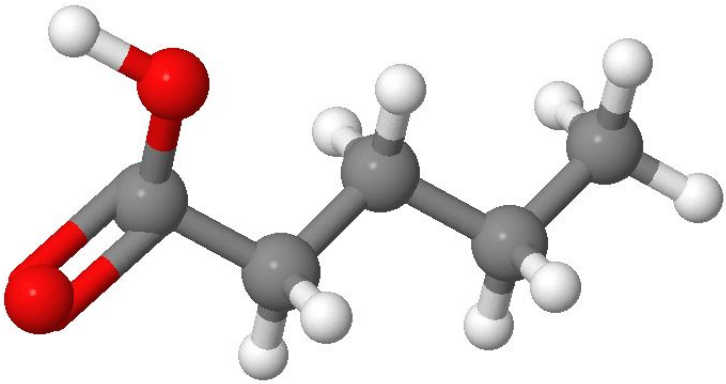


FIGURE 1.9 – Molécule de l'acide héxanoïque ($C_6H_{12}O_2$) un acide carboxylique.

Les molécules dont les atomes sont liés entre eux de la même manière mais différent par l'arrangement spatial de certains atomes sont appelés **stéréo-isomères**, ils présentent des différences de **conformation**.

1.1.7 La chimie quantique

Pour comprendre l'origine et la finalité du traitement des données moléculaires de ce mémoire il faut aborder la théorie quantique de la liaison chimique, qui est issue des travaux de Louis de Broglie [Bro25]. Cette théorie associe une onde à toute particule en mouvement et décrit le comportement de la particule, dans notre cas un électron, par une *fonction d'onde* ψ. Cette théorie ne permet pas de connaître avec précision la position de l'électron dans l'atome. L'électron n'a pas d'orbite déterminée mais il reste la plupart du temps dans une région de l'espace délimitée par une surface où la densité électronique est constante [MG01; Lew03]. Cette densité électronique est exprimée sous la forme d'une probabilité de présence d'électron dans une région donnée. La figure 1.12 montre quatre isosurfaces de densité électronique de la molécule d'eau (H_2O). La surface rouge présente une densité de 0.6 ; la bleue de 0.15 ; la verte de 0.005 et la jaune de 0.0001.
La portion de l'espace où réside majoritairement l'électron est appelée **orbitale**, il s'agit donc d'une région de l'espace délimitée par une surface où la densité électronique est constante.

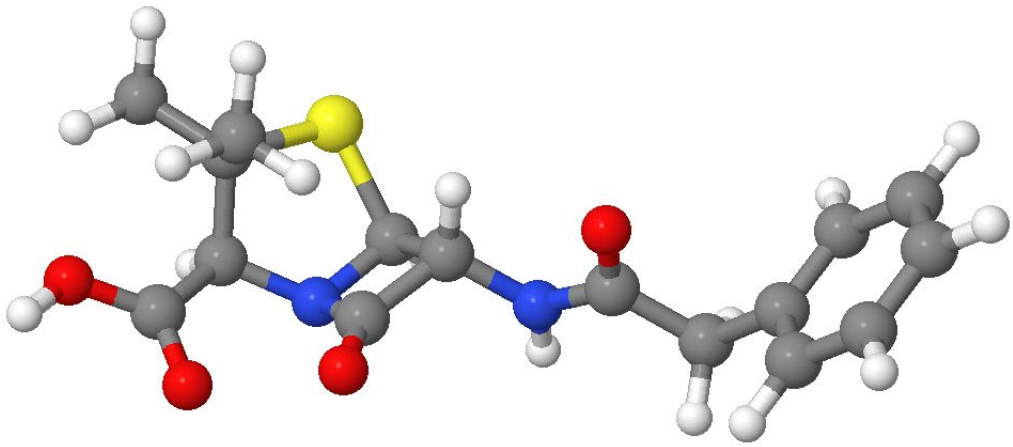


FIGURE 1.10 – Molécule de Pénicilline G pourvue de 4 sortes de GF.

La localisation de la particule s'exprime alors comme la probabilité dp de trouver l'électron dans un volume intermédiaire dV selon la formule suivante :

$$dp = \psi^2 dV$$

où ψ est une fonction d'onde normalisée, c'est à dire que :

$$\int \psi^2 dV = 1$$

ψ^2 est appelée la *densité de probabilité*. Le comportement de la particule est entièrement décrit par sa fonction d'onde, dont la forme générale est donnée par *l'équation de Schrödringer*, une équation fondamentale en physique quantique non relativiste. Au même titre que la relation fondamentale de la dynamique en mécanique classique (dite *newtonienne*), l'équation de Schrödringer décrit l'évolution en fonction du temps du comportement d'une particule massive non-relativiste [Sch26; JV91].
Quant à la liaison chimique, la théorie quantique implique que deux atomes sont liés par leurs électrons qui se déplacent sur des orbitales communes.
Jean et Volatron [JV91] ajoutent que décrire une molécule par la mécanique quantique soulève des difficultés directement liées à la nature polyatomique de la molécule. L'équation de Schrödringer, est alors encore plus difficile à résoudre. En effet, si on prend le cas de l'hydrogène qui ne possède qu'un électron, la résolution exacte de l'équation de Schrödringer est possible. Il en est tout autre lorsque que l'atome est *polyélectronique* et d'autant plus pour les systèmes polyatomiques que sont les molécules. La résolution repose alors

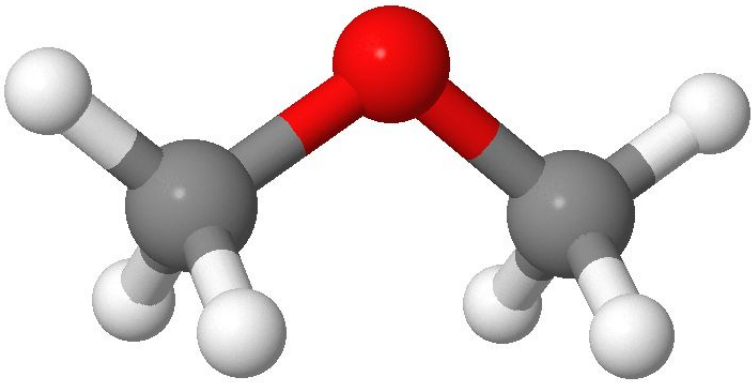


FIGURE 1.11 – Molécule du Diméthyléther, isomère structural de l'éthanol.

sur des hypothèses simplificatrices qui donnent une solution, bien qu'approximative, représentative de la solution exacte.

On utilise au choix l'une des trois techniques d'approximations suivantes :
- *L'approximation de Born-Oppenheimer* qui consiste à considérer que les électrons se meuvent dans un champ où le noyau atomique est immobile.
- *L'approximation orbitalaire* qui revient à écrire une fonction d'onde polyélectronique ψ sous la forme d'un produit de fonctions à un seul électron.
- *La théorie combinaison linéaire des orbitales atomiques* qui revient à exprimer les orbites moléculaires comme des Combinaisons Linéaires des Orbitales Atomiques des différents atomes constituant la molécule étudiée.

1.1.8 La Modélisation Moléculaire

La modélisation moléculaire consiste en un ensemble de techniques pour traiter **informatiquement** les problèmes soulevés par la chimie. [Lew03] expose ces problèmes comme suit :
- *La géométrie moléculaire* détermine la forme des molécules, la longueur des liaisons chimiques, les angles considérés.
- *L'énergie des molécules et les états de transitions* détermine les isomères qui persistent à l'équilibre et à quelle vitesse.
- *La réactivité chimique* détermine où sont regroupés les électrons et localise les faiblesses des molécules.

FIGURE 1.12 – Surfaces d'isodensité électronique de la molécule d'eau (H_2O).

- *Les spectres Infra-rouge, Ultra-Violets et Résonance Magnétique Nucléaire* identifie les molécules par leur signature spectrale.
- *Les interactions d'un substrat avec une enzyme* permettent de voir comment une molécule s'adapte le mieux au site actif d'une enzyme. Il s'agit d'une des approches utilisée pour concevoir de nouveaux médicaments.
- *Les propriétés physiques des substances* dépendent à la fois des propriétés individuelles de chaque molécule et de comment elles inter-réagissent avec l'ensemble.

1.1.9 Les outils de la Modélisation Moléculaire

L'étude des problèmes évoqués précédemment, met à disposition des chimistes cinq classes d'outils différents [Lew03; Jen99] :

1. *La mécanique moléculaire (MEM)* voit la molécule comme un ensemble de boules (atomes) reliées entre elles par des ressorts (liaisons chimiques), elle est régit par des modèles issus de la mécanique newto-

nienne. La figure 1.13 [2] schématise la vision mécanique moléculaire.

2. Le calcul *ab initio* [3] : cette méthode est directement basée sur *l'équation de Schrödringer*. L'équation est résolue pour déduire l'énergie E et fonction d'onde ψ.
3. La méthode quantique *semi-empirique* : cette méthode est aussi basée sur *l'équation de Schrödringer*. Mais contrairement à la méthode *ab initio*, la méthode semi-empirique utilise des données ajustées sur des résultats expérimentaux afin de simplifier les calculs. Les données ajustées sont une sorte de bibliothèque d'intégrales qui donnent les meilleurs résultats sur des entités calculées telles que la géométrie ou l'énergie.
4. *La dynamique moléculaire* : cette méthode applique les lois des mouvement aux molécules. Ainsi on peut simuler les mouvements d'une goutte d'eau autour de la molécule d'un soluté donné.
5. *La théorie de la Fonctionnelle de la Densité (DFT)* : à l'image des méthodes *ab initio* et *semi-empirique*, elle est basée sur *l'équation de Schrödringer*, à la différence néanmoins que la fonctionnelle de la densité ne calcule pas de fonction d'onde ψ. Elle dérive directement la distribution des électrons (la fonction de la densité électronique).

Seule la *théorie de la Fonctionnelle de la Densité (DFT)* sera détaillée dans ce mémoire car elle est le coeur théorique de la source de nos données.
Attacher une molécule au site actif d'une enzyme pour déterminer comment s'adapte et se comporte la relation, est une application extrêmement importante de la modélisation moléculaire. Lewars [Lew03] ajoute que cette discipline revient finalement à manipuler des substrats avec une souris d'ordinateur pour essayer de les intégrer aux sites actifs d'enzymes à l'aide d'un logiciel. Ces expériences de *docking* moléculaire visent à conduire la recherche de médicaments, *i.e.* des molécules qui devront inter-réagir avec certaines enzymes et être ignorées par d'autres. Lewars [Lew03] conclut alors que les points forts de la modélisation moléculaire se trouve dans l'étude des propriétés des matériaux, et plus généralement de la science de la matière. Les semi-conducteurs, les plastiques, les céramiques, la nanotechnologie ; la recherche dans tous ces domaines repose sur la modélisation moléculaire.

2. Source : http ://www.chem.ucla.edu/c125/NIH/MolMechanics.htm
3. La locution latine *ab initio* signifie *depuis le début/commencement*

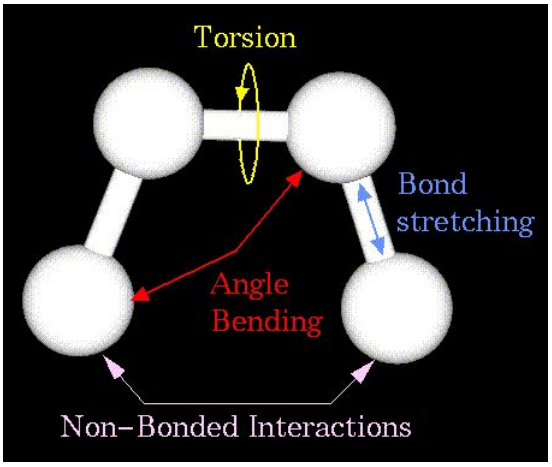


FIGURE 1.13 – Molécule du point de vue de la mécanique moléculaire, "Energy = Bond Stretching Energy + Angle Bending Energy + Torsion Energy + Non-Bonded Interaction Energy"

1.1.10 La théorie de la Fonctionnelle de la Densité

La fonction d'onde ψ n'est une grandeur mesurable que pour un atome ou une molécule d'où la complexité des outils de la modélisation moléculaire qui s'en servent. Au contraire, la théorie de la fonctionnelle de la densité (DFT) n'est pas dépendante de la fonction d'onde ψ, mais plutôt d'une fonction de densité de probabilité électronique qu'on appelle communément : **la densité électronique**. Elle s'écrit ρ(x,y,z) car elle est caractérisée par sa position dans l'espace tri-dimensionel [Lew03; Jen99]. C'est donc une probabilité par unité de volume ; la probabilité d'avoir un certain nombre d'électrons dans un certain volume $dxdydz$ centré sur le point de coordonnée (x,y,z). L'unité de ρ est donc $volume^{-1}$ et puisque les unités $dxdydz$ sont aussi des volumes, ρ(x,y,z)$dxdydz$ est un nombre sans unité, une probabilité. La fonction de densité électronique, contrairement à la fonction d'onde ψ, est mesurable par *diffractométrie de rayons X* [KA54] ou par *diffraction des électrons* [BG01]. Au-delà de cette propriété qui la rend observable et donc facilement appréhendable, la DFT possède un avantage majeur sur toutes les autres méthodes. Elle est fonction de trois variables (x,y,z) alors que les méthodes basées sur une fonction d'onde de n électrons d'une molécule comptera $4n$ variables [Lew03; Jen99]. Argaman *et al.* [AM00] précisent que durant les trente dernières années, la fonctionnelle de la densité n'a cessé de devenir la méthode

de prédilection pour les problèmes de la modélisation moléculaire car elle présente le double avantage d'être simple d'un point de vue calculatoire et d'être capable de traiter de nombreux problèmes tout en conservant un degré de précision plus qu'acceptable. Ils insistent en disant que la validité de la méthode est vérifiée en pratique par son habilité à reproduire des résultats expérimentaux, cela, malgré le fait que toutes les mises en application de la DFT soient basées sur des approximations incontrôlées.
La quantité $|\psi|^2$ a une interprétation physique très importante [Lew03; Jen99]. On l'appelle **l'interprétation de Born** d'une fonction d'onde ψ. Elle décrit la probabilité de trouver un système d'électron dans une région donnée à un instant donné. Dans le cas d'une fonction d'onde ψ à un seul électron, la grandeur $|\psi|^2$ est le densité électronique. Par contre, le calcul se complique pour un système multi-électron. [Lev91] a démontré que ρ(x,y,z) est décomposée spatialement en fonction d'ondes mono-électron ψ_i par :

$$\rho = \sum_{i=1}^{n} n_i |\psi_i|^2$$

La somme est effectuée sur n, le nombre d'orbitales moléculaires ψi occupées. De nos jours, Lewars [Lew03] affirme que les calculs de fonctionnelle de la densité sont basé sur les deux théorèmes de Hohenberg et Kohn. Le premier dit que toutes les propriétés d'une molécule dans un état fondamental sont déterminés par la fonction de densité électronique de l'état fondamental ρ_0(x,y,z). C'est à dire qu'étant donné un ρ_0(x,y,z), en principe, on peut calculer toutes les propriétés de l'état fondamental, par exemple l'énergie E_0 ; qu'on peut représenter ainsi :

$$\rho_0(x, y, z) \rightarrow E_0$$

Cette relation signifie que E_0 est une **fonctionnelle** de $\rho_0(x, y, z)$. À différencier d'une **fonction** qui est une règle de transformation d'un nombre en un autre nombre, une **fonctionnelle** est une règle qui transforme une fonction en un nombre.
Le premier théorème de Hohenberg et Kohn dit par ailleurs, que toute propriété d'un état fondamental d'une molécule est une *fonctionnelle* de la fonction de densité électronique de l'état fondamental. Ce qui revient à dire qu'**il est pertinent de chercher une façon de calculer les propriétés d'une molécule à partir de sa densité électronique** [Lew03].

Les données d'entrée utilisées pour la réalisation de ce mémoire sont produites par un logiciel qui applique les techniques de la *théorie fonctionnelle de la densité*.

1.1.11 Carte de Densité électronique

Les logiciels de modélisation moléculaire implémentent la plupart des outils présentés : le calcul *ab initio*, les méthodes *semi-empirique* et la fonctionnelle de la densité. Dans le cas particulier de la fonctionnelle de la densité, et celui de nombreux fichiers de données chimiques (e.g. : PDB qui n'a rien à voir avec la DFT), les données de densité électronique sont disponibles sous la forme de fichier texte à la mise en forme très simple. Cette forme facilite le traitement car la taille d'un fichier réaliste, contenant les données d'une molécule d'une protéine commune, compte des millier d'atomes. Par exemple le fichier correspondant à la figure 1.2 se compte rapidement en téra-octets. La densité électronique récoltée est présentée dans un volume tri-dimensionnel lui-même composé d'un ensemble de volumes unitaires, ou **voxel**, déterminé lors de la génération des données. Cet ensemble de probabilités de densité électronique est appelé **carte de densité électronique (EDM)** [Leh01]. La figure 1.14 montre une des molécule étudiée dans ce mémoire, l'anticoagulant NAPAP. La figure 1.15 montre sa carte de densité électronique, et un zoom à 300% sur des atomes d'oxygène est visible sur la figure 1.16. Ce zoom permet de prendre la mesure de la discontinuité des données. Chaque boule bleue représente le centre du volume unitaire courant dans lequel est évalué la probabilité d'y trouver un certain nombre d'électrons (i.e : carte de densité électronique).

Par analogie avec le monde du traitement du signal, on peut dire que la carte de densité électronique est la matière première de ce mémoire, notre **signal source**.

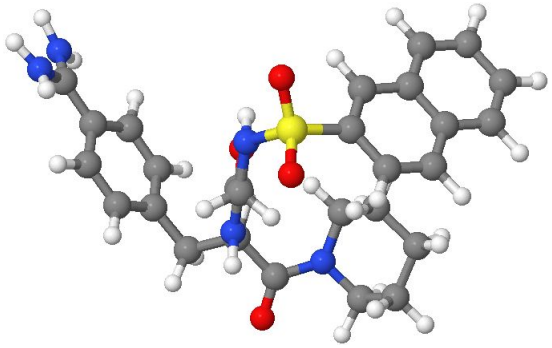


FIGURE 1.14 – Structure de la molécule d'anticoagulant NAPAP.

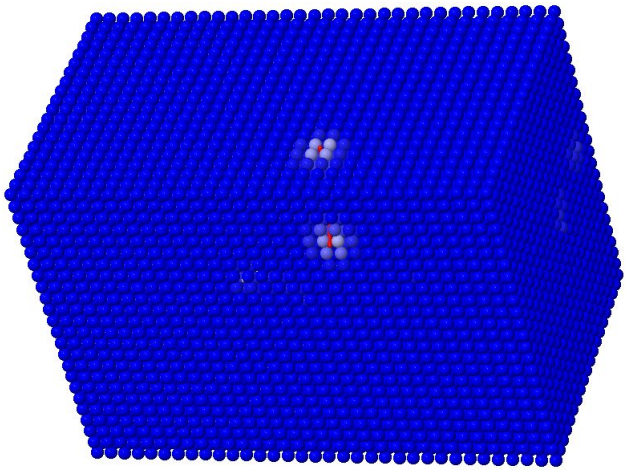


FIGURE 1.15 – *Cube* de densité électronique de la molécule d'anticoagulant NAPAP (résolution $69 \times 48 \times 39$, voxel de $0.762981^3 Å^3$).

1.1.12 Un fichier de densité électronique : Gaussian Cube

Lewars [Lew03] affirme que le logiciel *Gaussian* est le plus utilisé dans le monde de la modélisation moléculaire. Il s'agit en fait d'une suite logicielle qui couvre les méthodes *ab initio*, *semi-empirique* et la *fonctionnelle de la densité*. Toutes ces méthodes de haute-précision sont accessibles par des mots-clés par une ligne de commande. *Gaussian* n'a pas d'interface graphique homme-machine car il existe de nombreux logiciels pour visualiser ses résultats. A l'aide du logiciel **cubegen**, *Gaussian* permet de générer des fichiers *cube* qui sont des supports de cartes de densité électronique ou de potentiel électrostatique. Dans le cadre de ce mémoire nous traiterons des fichiers de type **cube** issus du logiciel *Gaussian*.

Le fichier *cube* s'articule comme suit [Bou03] :

Un entête : Deux premières lignes de commentaires, parfois utilisées comme une étiquette par défaut. La troisième ligne indique le nombre d'atomes n dans le volume ainsi que la position en Ångström (Å) de l'origine des données volumétriques. Les trois lignes suivantes indiquent le nombre de voxels par dimension ainsi que la longueur de l'arête sur la dimension considérée. Les n lignes suivantes, une ligne par atomes, indique le numéro atomique et la position du centre de chacun des atomes en Ångström (Å).

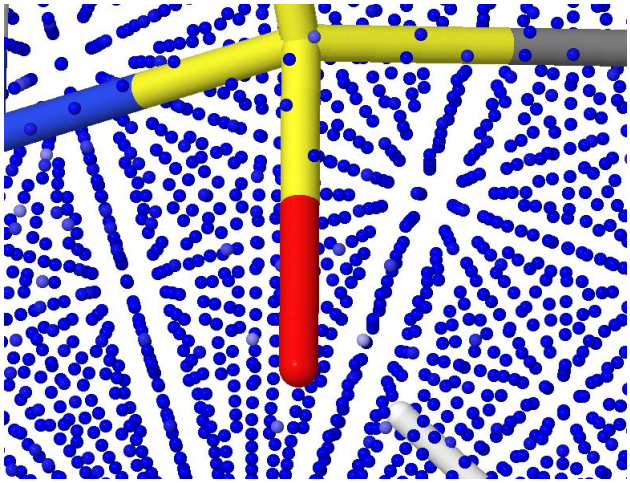


FIGURE 1.16 – *Cube* de densité électronique de la molécule d'anticoagulant NAPAP (résolution $69 \times 48 \times 39$, voxel de $0.762981^3 Å^3$, zoom à 300%).

Les données volumétriques : Ces données sont très simplement ajoutées à la suite en conservant une tabulation entre chaque élément et un saut de ligne tous les 6 éléments s'il y en a plus de 6.

[Bou03] propose l'algorithme C suivant qui sert à lire les données volumétriques d'un fichier *cube* :

```
// NX = nombre de voxels selon l'axe des X
// NY = nombre de voxels selon l'axe des Y
// NZ = nombre de voxels selon l'axe des Z
for ( ix = 0; ix < NX; ix++ )
{
        for ( iy = 0; iy < NY; iy++ )
        {
                for ( iz = 0; iz < NZ; iz++ )
                {
                        printf( "%g ", data[ix][iy][iz] );
                        if ( iz % 6 == 5 )
                                printf("\n");
                }
                printf("\n");
        }
}
```

Exemple : Le deux premiers plan selon l'axe Z de l'EDM de la molécule d'eau à la résolution $31 \times 31 \times 31$ et un voxel de $0.333333^3 Å^3$.

```
My test program: Water -- single point energy density=SCF
Electron density from Total SCF Density
    3    -4.970736    -4.761272    -4.970736
   31     0.333333     0.000000     0.000000
   31     0.000000     0.333333     0.000000
   31     0.000000     0.000000     0.333333
    8     8.000000     0.000000     0.209464     0.000000
    1     1.000000     1.481184    -0.838004     0.000000
    1     1.000000    -1.481184    -0.837705     0.000000
 5.77681E-11  1.62339E-10  4.24674E-10  1.03416E-09  2.34435E-09  4.94721E-09
 9.71848E-09  1.77718E-08  3.02520E-08  4.79350E-08  7.06991E-08  9.70561E-08
 1.24012E-07  1.47476E-07  1.63224E-07  1.68131E-07  1.61178E-07  1.43802E-07
 1.19407E-07  9.22818E-08  6.63799E-08  4.44433E-08  2.76976E-08  1.60678E-08
 8.67682E-09  4.36176E-09  2.04110E-09  8.89138E-10  3.60559E-10  1.36108E-10
 4.78284E-11
 1.29614E-10  3.64278E-10  9.53070E-10  2.32130E-09  5.26330E-09  1.11098E-08
 2.18307E-08  3.99338E-08  6.80003E-08  1.07786E-07  1.59027E-07  2.18381E-07
 2.79108E-07  3.31986E-07  3.67484E-07  3.78545E-07  3.62872E-07  3.23705E-07
 2.68733E-07  2.07628E-07  1.49302E-07  9.99281E-08  6.22547E-08  3.61027E-08
 1.94898E-08  9.79458E-09  4.58229E-09  1.99572E-09  8.09159E-10  3.05411E-10
 1.07311E-10
 [...]
```

1.1.13 Les anticoagulants MQPA, NAPAP et 4-TAPAP.

Pour valider l'efficacité de notre méthode nous avons besoin de données simples bien qu'étant assez riches pour permettre de mettre en évidence des GFs connus. Leherte [Leh01] rappelle qu'il est prouvé que lors de traitements destinés à identifier des similarités ou à détecter des GFs chimiques au sein de molécules organiques, les anticoagulants MQPA (figure 1.18), NAPAP (figure 1.19) et 4-TAPAP (figure 1.20) sont tout à fait qualifiés. En effet, pour des molécules organiques, elles sont assez petites - respectivement 71, 69 et 59 atomes - et pour autant, riches en GFs. Leherte [Leh01] remarque

en commun sur chacune des molécules, des groupes *Carboxyle* (CO-OH), *Carbonyle* (O=C=), *Sulfonyle* (SO_2), *Amidine* (-C(=NH)-NH_2) et *Pipéridine* ($C_5H_{11}N$). Elle ajoute que ces trois molécules adoptent toutes une structure en étoile à trois branches terminées par une fonction *Sulfonyle*, un anneau *Pipéridine* ou un groupe *Amidine*.

Pour finir, ces molécules sont assez simples pour en appréhender de façon aisée la géométrie, et assez riches pour proposer des variations de densités électroniques remarquables quelqu'en soit la résolution et le volume étudié. Elles sont les trois molécules *"étalon"* de ce mémoire.

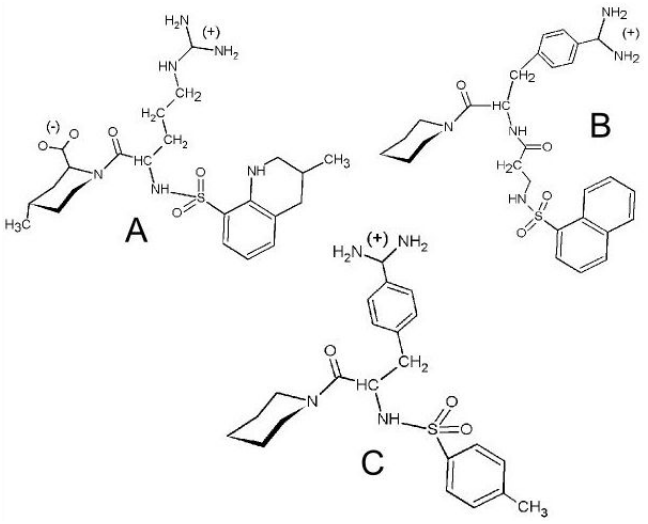


FIGURE 1.17 – Structure plane des molécules d'anticoagulant MQPA (A), NAPAP (B) et 4-TAPAP (C).

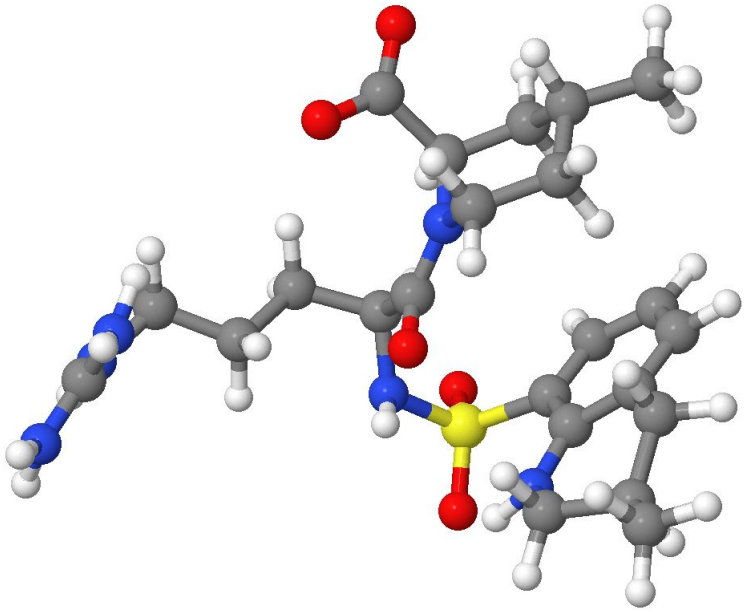


FIGURE 1.18 – Structure de la molécule d'anticoagulant MQPA (71 atomes).

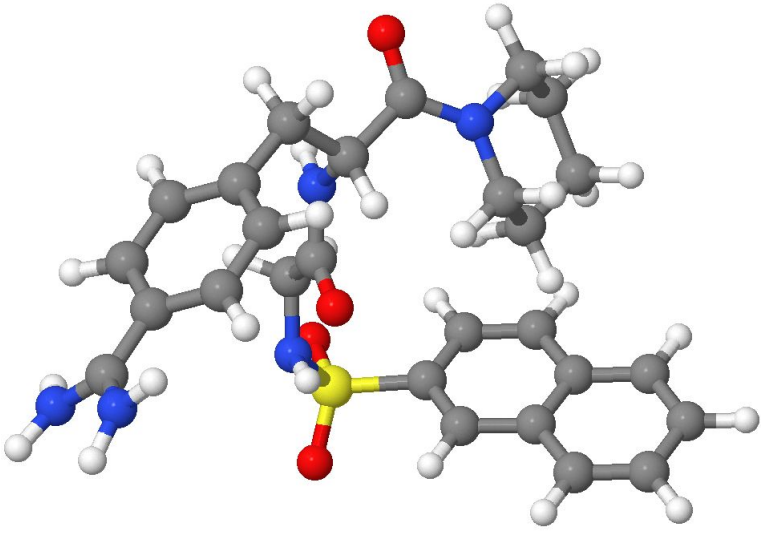


FIGURE 1.19 – Structure de la molécule d'anticoagulant NAPAP (69 atomes).

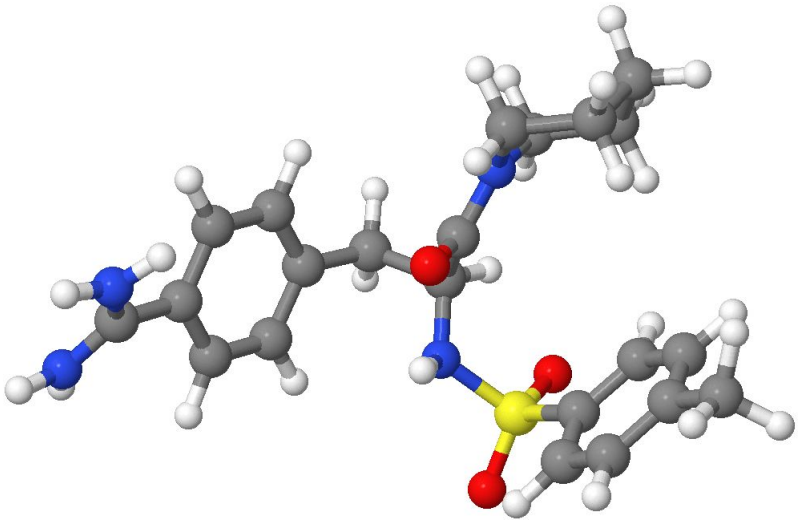


FIGURE 1.20 – Structure de la molécule d'anticoagulant TAPAP (59 atomes).

1.1.14 Conclusion

Le sujet de cette étude, bien qu'essentiellement orienté informatique du traitement du signal, porte sur la simplification et l'analyse multirésolution de données moléculaires. Celles-ci, issues de la modélisation moléculaire (MM), décrivent des cartes de densité électronique (EDM) de molécules organiques. Ces EDM sont le résultat concret de l'application des théories de la physique quantique et notamment la théorie de la fonctionnelle de la densité (DFT). Dans notre approche nous nous basons sur des fichiers d'EDM de type *cube* générés par le logiciel *Gaussian*. La disposition spatiale des groupes fonctionnels (GF) chimiques joue un rôle majeur pour les propriétés des molécules, ainsi notre méthode vise à simplifier les données pour détecter ces GF. Pour mener à bien cette tâche nous nous proposons d'étudier trois molécules organiques, plus précisément MQPA, NAPAP et 4-TATAP, des anticoagulants bien connus des chimistes.

1.2 La Quantification Vectorielle

1.2.1 Introduction

La méthode évoquée dans ce mémoire est basée sur la transposition des concepts et des outils de quantification vectorielle à des données autres qu'un signal multimédia. L'application des champs de ce domaine, alternatif au monde de la chimie, est proposée à des fins de simplification, de compression et d'analyse multirésolution de données moléculaires.
Une description générale de la quantification s'énonce comme la détermination mathématique d'une approximation pour un signal donné [GG92]. La quantification repose sur la représentation d'un ensemble de nombres (ou vecteurs) par un ensemble plus réduit appelé le **dictionnaire** [4] [GG92; Ric96]. Il s'agit donc d'un outil d'analyse qui permet de limiter le volume d'information tout en ayant l'assurance de conserver l'information pertinente. La quantification est communément appliquée en codage source afin de réduire le volume de données nécessaires pour représenter l'information [Ric96], la compression et transmission du son, de l'image et de la vidéo sont ses principales cibles.

1.2.2 La Quantification Vectorielle

La quantification vectorielle (QV) consiste selon [GG92; Ric96] à exprimer tout vecteur x de dimension k par un autre vecteur y_i de même dimension. Avec y_i, appelé **vecteur représentant** [5], appartenant à l'ensemble fini du dictionnaire D de L vecteurs.
Un cas particulier de quantification vectorielle (QV) en dimension 1 est donc

4. codebook en anglais
5. codeword en anglais

la quantification scalaire.

On peut définir un quantificateur vectoriel de dimension k et de taille L comme une application Q de R^k vers D tel que :

$$Q : \begin{array}{lll} R^k & \rightarrow & D \\ x & \rightarrow & Q(x) = y_i \end{array}$$

avec

$$D = \left\{ y_i \in R^k / i = 1, 2, \ldots, L \right\}$$

L'application Q partitionne l'espace R^k en L sous-espace C_i appelés **régions de *Voronoï*** (figure 1.21). Les **code-vecteurs** sont aussi les centres des régions ou **cellules** de *Voronoï* (figure 1.21), les duals des réseaux réguliers de point.

La QV se compose de trois opérations (figure 1.22) qui sont la création du dictionnaire D, l'encodage et le décodage :

- *Le calcul du dictionnaire* est une opération très complexe et gourmande en ressource, elle procède souvent à partir d'une séquence d'apprentissage représentant la source. L'algorithme classique, l'algorithme LBG (Linde, Buzo, Gray) (LBG), construit un dictionnaire dit *optimal* en termes de reconstruction de la séquence. Il s'agit d'une extension de l'algorithme de Lloyd-Max au cas vectoriel [GG92; LBG80].
- *L'encodage* revient pour tout vecteur x à chercher son représentant y_i dans le dictionnaire D le plus proche. On parle de recherche exhaustive qui est aussi très coûteuse en terme de calcul. L'indice i de y_i est appelé **index**. Il s'agit de la phase de compression de donnée car c'est souvent l'index i de y_i qui est stocké ou transmis.
- *Le décodage* est voué à reconstruire le vecteur source à l'aide d'une copie du dictionnaire D. Le décodeur restitue le représentant correspondant à l'index qu'il a reçu ou stocké, il est l'acteur de la décompression.

Lors d'une opération de quantification, il y a souvent une différence (erreur) entre le vecteur y_i attribué et le vecteur d'origine. Cette erreur iest mesurée par la distance $d(x, y_i)$. L'erreur moyenne faite en comprimant le signal est souvent appréciée à travers le calcul de l'erreur quadratique moyenne (EQM) qu'on ne détaillera pas ici.
Dans le cadre de ce mémoire, seul l'aspect **analyse** de la quantification est utilisé car notre but est d'extraire des points critiques qui caractérisent les groupes fonctionnels chimiques. Ainsi, le lecteur intéressé est invité à consulter [GG92; Ric96] afin d'obtenir plus de détails sur la quantification vectorielle.

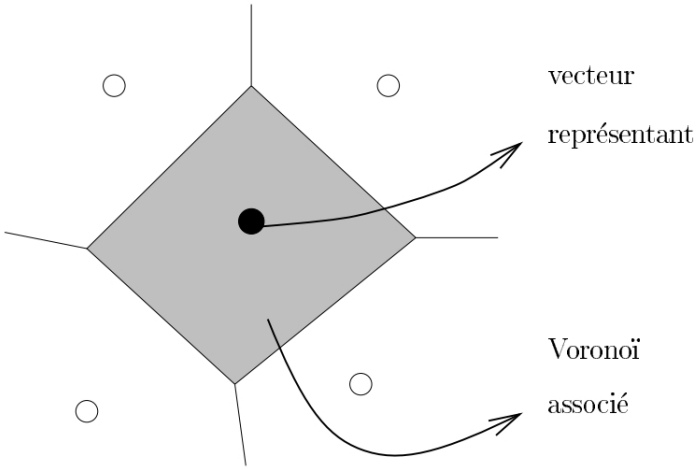


FIGURE 1.21 – Principe de la quantification vectorielle.

1.2.3 La Quantification Vectorielle Algébrique

La quantification vectorielle algébrique (QVA) est le résultat de nombreux travaux de recherche qui ont eu pour objectif d'accélérer la construction du dictionnaire et l'encodage. La QVA consiste à structurer fortement le dictionnaire afin d'en simplifier la conception [Ric96]. La QVA ne nécessite

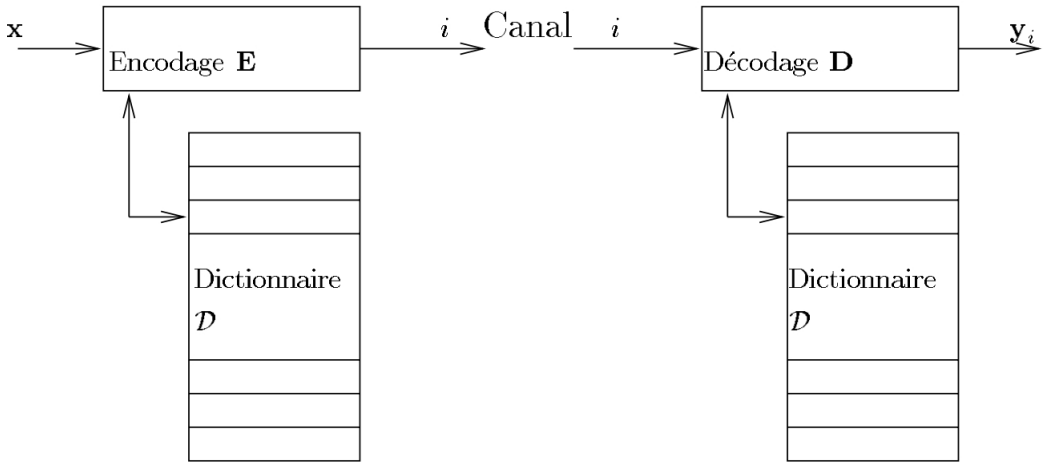


FIGURE 1.22 – Schéma du quantificateur vectoriel dans un contexte de codage/décodage.

pas d'étape d'apprentissage ni de recherche exhaustive pour construire le dictionnaire. En effet, les vecteurs représentants ou **code-vecteurs** sont les points d'un **réseau régulier de point** (ou *Lattice* en anglais) [CSB93]. La compacité de l'occupation (l'empilement) d'un réseau régulier de point dans l'espace déterminera ses propriétés. Contrairement à une méthode classique basée sur l'algorithme LBG (Linde, Buzo, Gray) (LBG) [GG92; LBG80], en QVA l'encodage est lié aux coordonnées du vecteur. Ainsi, il est simplifié car basé sur des opérations d'arrondi et de mise à l'échelle. Par ailleurs, la quantification est indépendante de la taille du réseau. Á cela s'ajoute l'inutilité de transmettre le réseau, il est naturellement connu à la fois du codeur et du décodeur. Pour finir, le critère de choix d'un réseau pour la QVA réside dans l'existence d'algorithmes de quantification rapide en fonction du réseau choisi. Toutes ces propriétés rendent le codage simple en termes de calcul [CSB93; Ric96]. Une des origines mathématique des réseaux réguliers de point est la recherche de l'empilement optimal (c'est-à-dire le plus dense)

de sphère dans l'espace.

Conway *et al.* et Ricordel [CSB93; Ric96] énoncent que le **rayon** ρ caractérise des sphères identiques empilées et chaque sphère est inscrite au sein d'un polytope au centre du *Voronoï* caractéristique du réseau [Ric96](figure 1.23). Il est aussi démontré que la qualité de la quantification est directement dépendante de la géométrie du polytope qui décrit la cellule de *Voronoï*. Une synthèse de [Ric96] montre qu'en dimension 3, il n'existe que trois réseaux réguliers de point, dont un algorithme de quantification rapide et optimal est connu. Il s'agit des réseaux $Z3$, réseau à la cellule de *Voronoï* Cubique ; $D3$ dont la cellule de *Voronoï* est un dodécaèdre rhombique ; $D3^*$, un octaèdre tronqué (figure 1.24).

Nos données impliquent de travailler en dimension 3, ce sont donc ces trois réseaux qui vont nous intéresser au cours de ce mémoire.

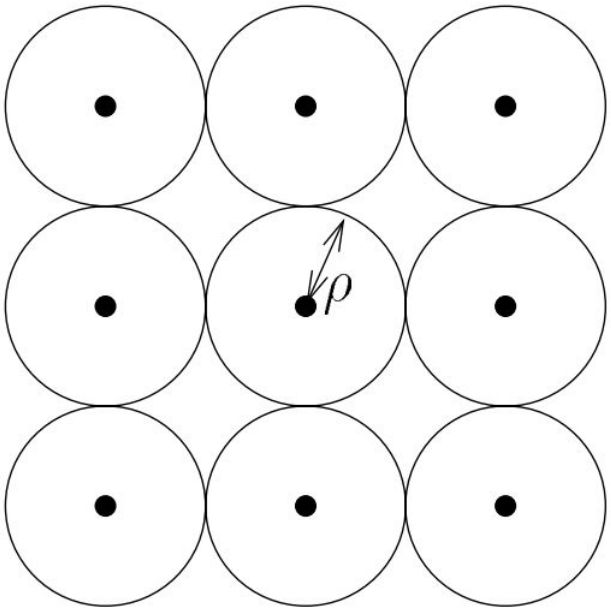


FIGURE 1.23 – Un empilement de sphères dans l'espace bidimensionnel.

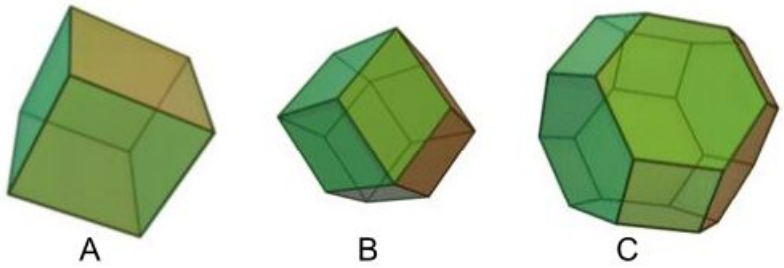


FIGURE 1.24 – Cube (A), Dodécaèdre rhombique (B), Octaèdre tronqué (C).

1.2.4 La Quantification Vectorielle Arborescente

Un autre aspect de la QV propose de structurer l'espace de la source à l'aide d'une arborescence. Ceci à l'intérêt de faciliter l'analyse, et par exemple de réduire la complexité de la recherche exhaustive au sein du dictionnaire.
La quantification vectorielle arborescente (QVAr) est une combinaison de plusieurs approches de quantification où le traitement est basé sur un arbre de décision [GG92]. Parmi les avantages de ce type de QV, on distingue une charge de calcul réduite grâce à l'usage de dictionnaires intermédiaires simplifiés. On remarque par ailleurs la conception d'une structure de représentation progressive et une structure appropriée pour le codage à débit variable [GG92; Ric96]. Cette propriété sera mise en avant dans ce mémoire, car la possibilité de faire une description structurée et multirésolution de l'espace source nous sera précieuse.
La QVAr propose, par ailleurs, deux visions antagonistes pour la construction de l'arbre de données : L'**élagage**, qui consiste a créer un arbre complet dont les branches seront supprimées par un processus itératif d'élagage, et le **découpage** qui par un processus itératif construit l'arbre au fur et à mesure de son parcours [GG92; Ric96]. Ce mémoire s'intéressera au processus de

découpage pour des raisons évidentes d'économie de ressource.

1.2.5 La Quantification Vectorielle Algébrique et Arborescente

[Ric96] propose de conjuguer la quantification vectorielle algébrique (QVA) et quantification vectorielle arborescente (QVAr) via son schéma de la quantification vectorielle algébrique et arborescente (QVAA) qui use d'une hiérarchie de réseaux réguliers de point, de façon à pouvoir emboîter un réseau régulier de point tronqué de plus petite échelle dans la cellule (de *Voronoï*) de plus grande échelle du niveau suivant dans la hiérachie. Un facteur d'échelle doit être déterminé (figure 1.25).

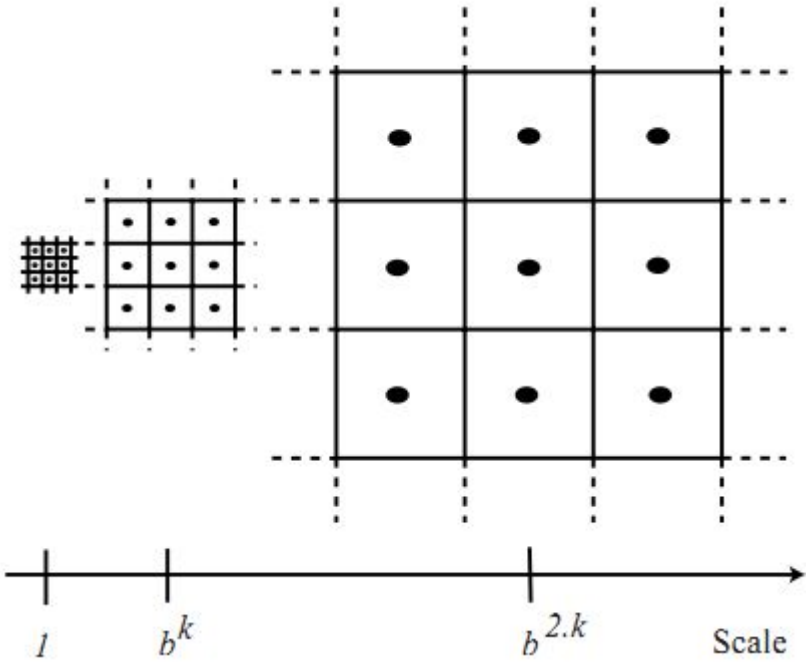


FIGURE 1.25 – Hiérarchie de réseaux cubiques en 2D. Facteur d'échelle b=3

D'après [RL95], le principe de la QVAA consiste à :

1. Projeter un vecteur source dans un premier réseau régulier de points tronqué ;
2. Affiner la quantification, en emboîtant un autre réseau identique mais à une d'échelle moindre, dans la cellule de *Voronoï* qui contient le vecteur source ;
3. Répéter l'opération précédente jusqu'à atteindre la résolution souhaitée.

Il est plus facile de travailler avec l'échelle du vecteur d'entrée que de considérer plusieurs réseaux à des échelles différentes. La figure 1.26 présente le principe du codage [Ric96].

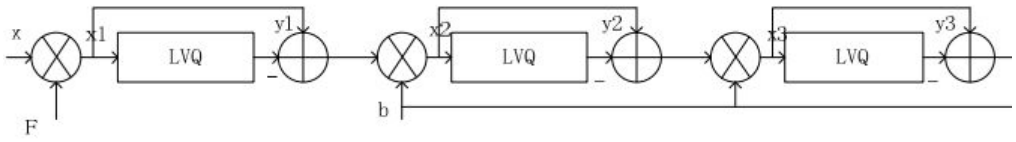


FIGURE 1.26 – Principe de la Quantification Vectorielle Algébrique et Arborescente ; x est le vecteur source, yn le vecteur de reproduction suivant, F le premier facteur d'échelle et b le facteur d'échelle suivant

On utilise le facteur d'échelle F pour projeter le vecteur source x dans le premier réseau régulier de point tronqué tel que :

$$F = \frac{b \times \rho}{L_{2max}}$$

Où ρ est le rayon de d'empilement, et L_{2max} la norme *euclidiènne* de x maximale. Ainsi, tous les vecteurs sont projetés dans une hyper-boule dont le rayon est : $b \times \rho$.

Dans un espace normalisé, b est le facteur d'échelle utilisé pour projeter chaque vecteur, aussi translaté du y_j précédent, dans le prochain réseau tronqué de la hiérarchie. Le vecteur de reproduction du $j^{ème}$ niveau est y_j. La

valeur finale du vecteur de reproduction y associé au vecteur source initial est alors :

$$y = \frac{1}{F} \times \sum_{j} \frac{y_j}{b^{j-1}}$$

Notons qu'à chaque itération j (ou étage de quantification), on utilise le même algorithme de quantification rapide (noté LVQ sur la figure 1.26).

1.2.6 Conclusion

La quantification vectorielle algébrique et arborescente (QVAA) est méthode d'analyse de la carte de densité électronique (EDM). Elle nous permet en effet de disposer d'un outil qui conjugue l'analyse hiérarchique (multirésolution) de l'espace source et une quantification rapide sur réseau algébrique. Les carte de densité électronique (EDM) sont des grilles tridimensionnelles cubiques, ainsi grâce à l'usage de la QVAA, il sera possible d'accéder à la structure des données selon une description arborescente, de parcourir l'arbre en multirésolution, de compresser les données tout en conservant l'information pertinente, et par-dessus tout, en accélérant de façon drastique l'analyse.

1.3 Les Points Critiques

1.3.1 Introduction

[Leh01] affirme que la forme tridimensionnelle de la molécule, c'est-à-dire la disposition des groupe fonctionnels (GFs) permet de déduire le comportement de la molécule. Elle introduit, par ailleurs, la notion de **point critiques (CPs)** pour qualifier la représentation spatiale de la molécule étudiée. Détecter et trouver des liaisons entre ces CP est alors décrit comme nécessaire et suffisant pour appréhender les propriétés de la molécule.
Caractériser des formes géométriques dans l'espace requiert l'usage d'outils mathématiques issue du domaine de la **topologie mathématique**. Une technique classique consiste à détecter et particulariser les **CPs** d'une forme. Ces points remarquables permettent d'identifier les **pics**, les **creux** et les **cols** (ou selles) (figure 1.27) de l'objet étudié. La théorie de Morse est l'origine de cet outil. [Mil63; Gab01; Esc11; Nak03].

1.3.2 Notions de topologie mathématique

Avant d'énoncer les principes de la détection des points critiques, il convient de définir un ensemble de notions de topologie mathématique. Tout d'abord, [Nak03] simplifie les concepts d'*espaces vectoriels* et de *topologie* comme deux visions différentes de la notion de *variété* mathématique. Nakahara [Nak03] définit alors une *variété* comme un espace *localement similaire* à $\mathbb{R}^n$ (ou $\mathbb{C}^n$) (une petite région autour d'un point d'une surface peut être approximée par le plan tangent à ce point) ; c'est la vision de l'espace vectoriel. Du point de vue de la topologie, on s'intéresse à la *variété* comme un tout. Il est alors utile d'étudier les propriétés des *variétés* pour les classifier à l'aide de *mesure*. On définit une **carte** f comme une règle qui associe deux ensembles X et Y.

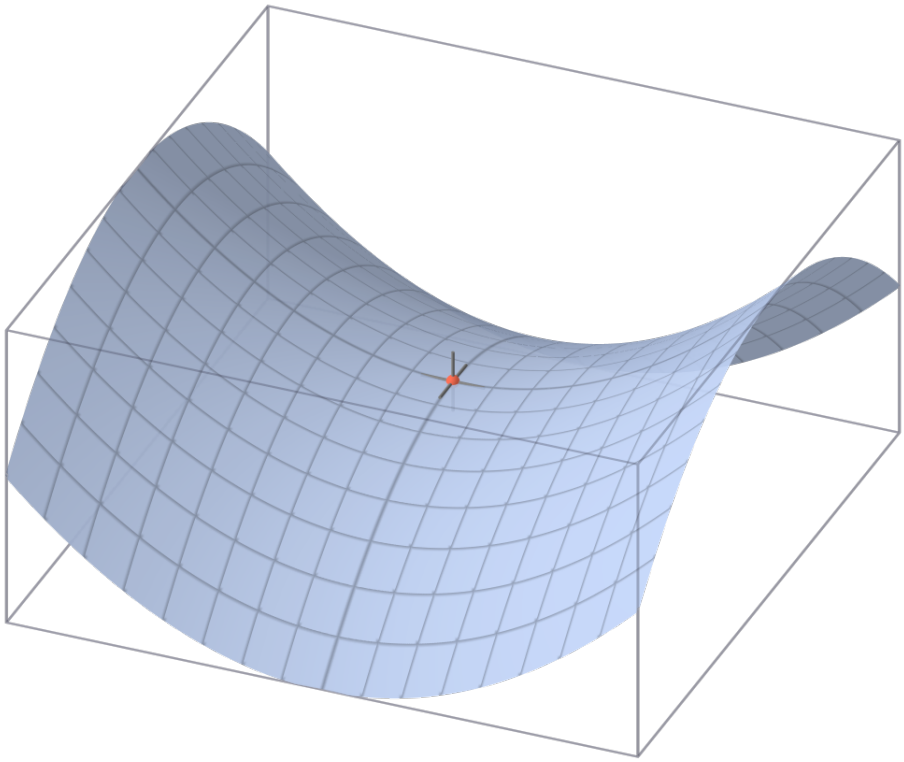


FIGURE 1.27 – Le point col est un point critique. (Source wikipédia)

Si $y \in Y$ et $x \in X$, alors :

$$f : X \to Y$$

Il peut y avoir plus de deux éléments de X correspondant au même $y \in Y$. Les couples (X, f) sont appelés **cartes**. Au lieu de *carte* on dit parfois aussi *système de coordonnées*. X est alors le *domaine* de la carte alors que Y est son *étendue*.

L'objectif de classer des espaces nécessite de définir les notions de *égal* et *différent*. En topologie, deux figures sont équivalentes s'il est possible de déformer une figure en une autre à l'aide d'une *déformation continue*. Le formalisme mathématique permet de définir la notion d'**homéomorphisme**. Soit $X1$ et $X2$, deux espaces topologiques. Une carte $f : X1 \to X2$ est un

homéomorphisme si elle est continue et a un inverse $f^{-1} : X2 \rightarrow X1$, lui aussi continue. $X1$ et $X2$ sont alors dit **homéomorphiques**
On caractérise alors les classes d'équivalences des homéomorphismes en démontrant que si deux espaces ont des **invariants topologiques** différents, ils ne sont pas *homéomorphiques*. L'*invariant topologique* peut être un nombre, par exemple le nombre d'objets connectés dans l'espace, un structure algébrique telle qu'un anneau ou même les *connectivités* et *la compacité*. La figure 1.28 présente un polyèdre et un tore qui ont la particularité d'être homéomorphiques. La déformation continue du polyèdre permet d'obtenir un tore.

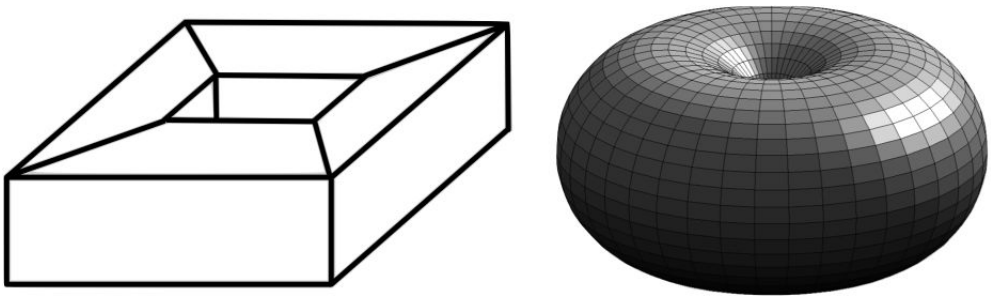


FIGURE 1.28 – Polyèdre homéomorphique à un tore.

La **caractéristique d'Euler** est l'*invariant topologique* le plus utilisé [Nak03]. Pour l'exemple, on se restreint à $\mathbb{R}^3$, dont on considère un sous-ensemble X, qui est homéomorphique au polyèdre K. Alors la **caractéristique d'Euler** $\chi(X)$ de X est définie par :

$$\chi(X) = (Nbre\ de\ sommets\ de\ K) - (Nbre\ d'arrêtes\ de\ K) + (Nbre\ de\ faces\ de\ K)$$

L'**homologie** est une version affinée de la *caractéristique d'Euler*, intuitivement, elle compte pour chaque dimension n, les n-dimensionnels trous dans l'espace. La **cohomologie** présente sur l'*homologie* l'avantage d'être structurée en anneau.

1.3.3 La théorie de Morse

[Mil63; Gab01] considèrent la **La théorie de Morse** comme une méthode pour caractériser la topologie d'une *variété* finie ou infinie à partir des CPs issus d'une seule fonction adéquate de la *variété*. Cette théorie a une portée considérable au sein de nombreuses disciplines, de la géodésie à la physique de matières en passant par la cristallographie. La théorie de Morse s'énonce ainsi :

Soit M une variété compacte de dimension n et une fonction f dite *fonction de Morse* $f : M \to \mathbb{R}$. On appellera $p \in M$ un *point critique (CP)* de f où les coordonnées locales de p ont la propriété suivante : $\frac{\partial f}{\partial x^1} = \ldots = \frac{\partial f}{\partial x^n} = 0$. Ce calcul se nomme le **gradient**. Par ailleurs, on détermine la nature d'un *point critique (CP)* en analysant les valeurs propres de la **matrice hessienne**. Le nombre de valeurs propres négatives se nomme **index de Morse**. [Esc11] décrit les *points critiques* de la théorie de Morse comme une application de la *cohomologie*.

1.3.4 Le gradient

Le **gradient** en mathématique noté ∇f, au même titre qu'une dérivée, représente les variations d'une fonction par rapport à la variation de ses paramètres. L'étude du gradient permet de détecter les points critiques d'une fonction donnée. Cela consiste à déterminer le valeurs de f où $\nabla f = 0$ [Mil63; Gab01; Esc11].

Si f est une fonction différentiable des variables $(x_1, x_2, \ldots, x_n)$, le gradient de f est la fonction :

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

1.3.5 La matrice Hessienne

La **matrice hessienne** (ou le Hessien) est la matrice *carrée* des dérivées partielles secondes d'une fonction f. Elle permet dans de nombreux cas de déterminer la nature des points critiques de la fonction f, c'est-à-dire des points d'annulation du *gradient* [Mil63; Gab01; Esc11].
Soit une fonction f différentiable des variables $(x_1, x_2, \ldots, x_n)$, la matrice hessienne $H(f)$ est la matrice carrée :

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

1.3.6 Nature des points critiques

Un point critique de f est dit **dégénéré** lorsque la valeur propre de la *matrice hessienne* est nulle. Le point critique reste alors indéterminé. Par contre si les valeurs propres sont toutes non nulles en un point critique donné, il est dit **non-dégénéré** et le signe des valeurs propres du hessien détermine la nature de ce point. On parle alors de **point d'extremum local** [Mil63; Gab01; Esc11]. Les conditions nécessaires d'un extremum local sont alors :
Si le point critique est **non-dégénéré** et ;

- Si p est un point de minimum local de f, alors c'est un point critique et le hessien en p est positif (toutes les valeurs propres sont positives). Le point critique est un **point creux**.
- Si p est un point de maximum local de f, alors c'est un point critique et le hessien en p est négatif (toutes les valeurs propres sont négatives). Le point critique est un **point pic**.

En particulier, si le hessien en un point critique admet au moins une valeur propre strictement positive et une valeur propre strictement négative, le point critique est un **point col** (ou *point selle*).

1.3.7 Conclusion

Les données d'entrée de notre méthode implique des données tridimensionnelles qui ont la particularité de décrire des objets géométriques. Cela sera d'autant plus vrai après l'étape de quantification, où l'objet sera composé de formes correspondant au *Voronoï* issu du *réseau régulier de point* à différentes échelles. Le monde de la science (physique, géologie, géodésie, etc) est déjà bien habitué à utiliser la notion de CP pour caractériser les formes géométriques tridimensionnelles. Dans notre cas, la connectivité des CPs nous permettra de modéliser de façon simple les zones riches en information de la molécule étudiée, en procédant donc à la façon de [Leh01].

1.4 La Quantification Vectorielle appliquée à la Modélisation Moléculaire

1.4.1 Introduction

Les travaux de [Leh01] et [BCCSX05] à base d'ondelette ainsi que les recherches de [DAPMGC02] focalisées sur la quantification vectorielle à l'aide de classifieur se sont attachés à appliquer les techniques du traitement du signal à la modélisation moléculaire. La démarche de ces deux méthodes vise à trouver un moyen de créer un mécanisme de *descripteurs* uniques (comme une signature) associé à chaque molécule et plus simplement à chaque groupe fonctionnel. La finalité est donc de créer une base de données des propriétés moléculaires sous forme de *descripteur* basés sur des diagrammes de connectivité pour Leherte [Leh01] et sur des *alpha-shapes* (un objet géométrique bien connu des biologistes moléculaires pour [DAPMGC02]).
Le propos de ce mémoire introduit tout d'abord une méthode pour développer de nouveaux descripteurs basés sur la quantification vectorielle algébrique et arborescente (QVAA) suivit de la détection de points critiques et par la suite sera conduite une comparaison avec le travail de [Leh01].

1.4.2 Détection de points critiques à base d'ondelettes

[Leh01] s'attache à comparer trois méthodes de traitement de carte de densité électronique (EDM). Le but est d'utiliser le concept des points critiques pour déterminer des *motifs* uniques représentants des groupes fonctionnels chimiques. Ces motifs détectables dans d'autres molécules doivent permettre d'identifier des groupes fonctionnels identiques. Les données décrivent trois anti-coagulants (MQPA, NAPAP et 4-TAPAP, C.f. §1.1.13).

Un enchaînement (workflow) de tâches simplifie les EDM en utilisant trois approches différentes ; la Cristallographique (XTAL) et le Lissage analytique (ASA) proche de la modélisation moléculaire classique et les *ondelettes (WMRA)* issues du domaine du traitement du signal. Leherte [Leh01] voit les points critiques comme des représentants d'un ensemble localisé de points de densité électronique qui, une fois agglomérés par des mécanismes de connectivité, se doivent de représenter des groupes fonctionnels. La multirésolution est un critère central offert par le définition même des ondelettes. Cet avantage est aussi un élément fondamental de notre méthode à base de quantification vectorielle algébrique et arborescente (QVAA). La détection de similarité se fait par superposition de graphes de points critiques. Les trois approches sont testées sur des cubes de données de dimension $128 \times 128 \times 128$. [BCCSX05] présentent une méthode d'analyse basée sur les ondelettes en accès temps-réel (traitement très rapide). Les données, fortement compressées selon une hiérarchie inhérente aux mécanismes des ondelettes, sont localement décodées pour être analysées et visualisées. Leherte [Leh01] constate qu'en employant les ondelettes, les premiers niveaux de simplification créent beaucoup de bruit et que seul un certain seuil de résolution permet une représentation des données acceptable. Les données sont donc simplifiées de façon destructive, les données négligeables sont mise à 0. Néanmoins, les ondelettes proposent un mécanisme de reconstruction approximatif qui est requis par Leherte [Leh01] pour conserver le nombre initial des valeurs. Ainsi quatre niveaux de décomposition en ondelettes suivit d'une reconstruction ont été nécessaires pour générer des images contenant les informations accessibles fournies par les autres approches plus classiques. Pour finir, elle remarque des temps de traitement 100 fois inférieurs aux approches classiques. [BCCSX05] propose un schéma très robuste aux erreurs de compressions grâce à des on-

delettes en 3D adapté à la géométrie des données disponibles, une facilité d'implémentation et un haut degré de précision ainsi que de compression.

1.4.3 Quantification Vectorielle de données moléculaires

[DAPMGC02] montre l'usage de la densité de probabilité, la quantification vectorielle et les **alpha shapes** [Kav07] pour créer des modèles synthétiques, au mieux, uniques, représentatifs et légers de la macromolécule. La méthode utilise d'abord la QV sous le contrôle de la densité de probabilité pour simplifier les données d'entrées. Les données simplifiées apparaissent sous forme de "pseudo-atomes" caractérisés par trois paramètres (position, densité, erreur de quantification). Puis les données originales sont alors considérées en 3D (voxel) pour en détecter les contours. Á l'aide des *alpha shapes*, on combine alors les données quantifiées ainsi que les contours en 3D pour construire une forme en 3D. Des "complexes Alpha" sont alors calculés pour donner une représentation juste mais compacte de la molécule. On compare alors, les similarités entre les données d'entrées et celles de sortie afin de prouver qu'on a limité la quantité d'information sans pour autant perdre en qualité d'information. Le but avoué est alors de remarquer que l'*alpha shape* est très caractéristique, voire unique pour une molécule donnée. Á tel point, qu'elle permet de donner une signature (un descripteur) archivable et réutilisable dans une base de données. De-Alarcon *et al.* [DAPMGC02] ont déterminé que le critère de fin de quantification est la similarité des densités de probabilité avant-après quantification. La similarité se détermine par le calcul erreur de quantification moyenne (distance moyenne entre le pseudo-atome et le vecteur source).

1.4.4 Conclusion

Il apparaît donc que l'application de la quantification vectorielle à des données moléculaires est encore à l'état embryonnaire. La fusion des domaines du traitement du signal et de la modélisation moléculaire est encore une activité à développer. Néanmoins, même si elles sont peu nombreuses, les différentes méthodes évoquées dans ce chapitre seront des références pour évaluer et comparer notre nouvelle approche de quantification vectorielle algébrique et arborescente (QVAA) appliquée à la modélisation moléculaire (MM).

Chapitre 2

Modélisation et réalisation

2.1 Introduction

Oana Cramariuc, spécialiste en physique quantique, Vincent Ricordel et Bogdan Cramariuc, spécialistes en traitement du signal ont formalisé l'idée que la recherche en caractérisation structurale d'une macromolécule pourrait être grandement simplifiée par l'utilisation de la quantification vectorielle. En effet, la physique quantique nous apprend que dans chaque atome, la localisation des électrons n'est possible que sous la forme de probabilité appelée **densité électronique**. Or cartographier des niveaux de densité électronique permet de représenter géométriquement une molécule dans l'espace. Cette forme tridimensionnelle est une donnée critique pour identifier les propriétés d'une molécule. Malheureusement, le volume des données d'une carte de densité électronique (EDM) est rapidement très important (de l'ordre du tera-octet pour une protéine simple). Il faut donc analyser et simplifier les données pour qu'elles soient exploitables.

Ce mémoire présente les résultats obtenus en appliquant la quantification vectorielle algébrique et arborescente (QVAA) à des données consommées à

partir de fichier de *cube*. Les qualités intrinsèques de la QVAA induisent des fonctions de simplification et d'analyse multirésolution.
La démarche, conceptualisée par les trois chercheurs, nous conduit à effectuer plusieurs traitements successifs de QVAA sur les données jusqu'à obtenir une vision simplifiée, mais riche en information, de la représentation moléculaire de la carte de densité électronique (EDM).
Nous allons détailler les choix techniques ainsi que les méthodes que nous avons appliquées. Cette démarche a été guidée par la nécessité d'obtenir un fichier *cube* à l'issue du traitement.

2.2 Paramétrisation

La littérature nous appprend que la définition de la quantification vectorielle algébrique et arborescente (QVAA) impose de faire des choix pour le traitement, notamment sur la méthode de quantification vectorielle arborescente (QVAr) (élagage ou découpage), la résolution visée, le réseau régulier de points utilisé, ainsi que la valeur du facteur d'emboîtement pour la quantification vectorielle algébrique (QVA) et pour finir le nombre de quantifications successives.

2.2.1 Mode de quantification arborescente

La quantification vectorielle arborescente (QVAr) dispose de deux modes de traitement pour créer des structures d'arbre non équilibrées. Le premier, l'**élagage**, impose de créer toutes les feuilles de l'arbre complet et de parcourir à nouveau tout l'arbre afin de le simplifier en supprimant les feuilles non pertinentes au regard de notre application. L'autre mode, le **découpage**, consiste à créer les feuilles et les branches de l'arbre au fur et à mesure du

traitement.

Nous avons fait le choix de l'approche en **découpage**, c'est à dire la construction de l'arbre *planté à l'envers* de façon descendante à partir de la racine avec un procédé de fusion en lien avec les données initiales pour suivre le schéma de la QVAA (figure 2.1).

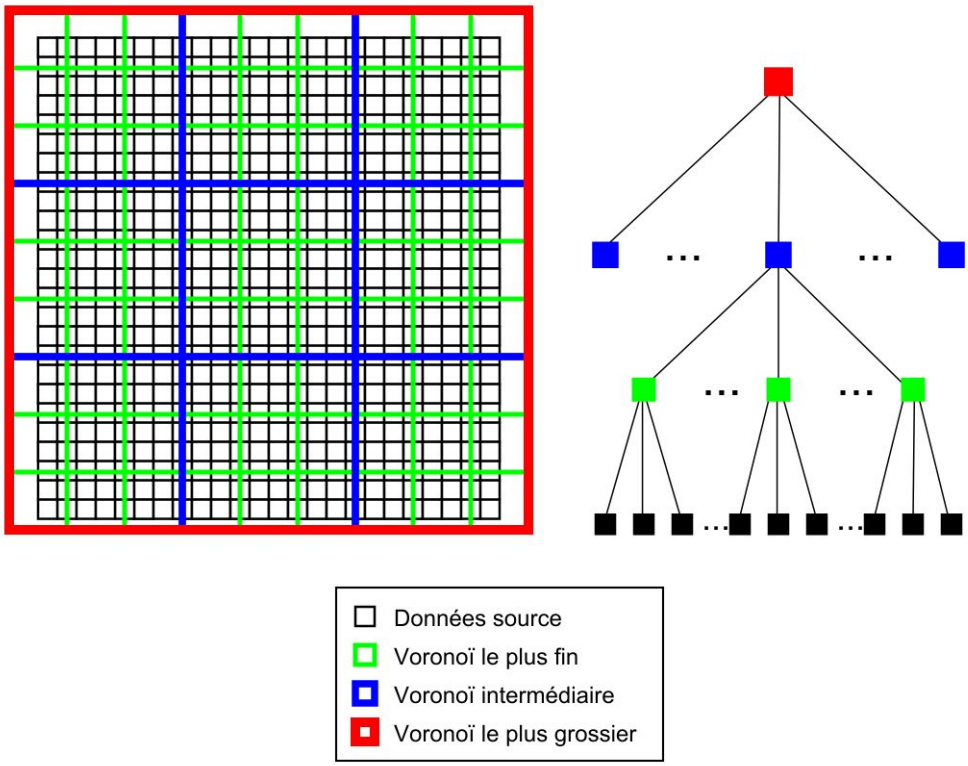


FIGURE 2.1 – Le découpage arborescent organise les données du plus fin (données source en noir) au plus grossier (en rouge).

2.2.2 Réseau régulier de points

La contrainte tridimensionnelle limite le nombre de réseaux réguliers de points utilisable car on ne connait d'algorithme de quantification rapide que

pour trois d'entre eux ; $Z3$, le réseau cubique ; $D3$, le réseau basé sur une cellule de *Voronoi* en dodécaèdre rhombique ; et $D3^*$ à la cellule de *Voronoi* en octaèdre tronqué.
Par ailleurs, le réseau cubique $Z3$ est moins performant que d'autres réseaux dont les polytopes des cellules de *Voronoi* sont plus proches de la forme d'une boule car ces derniers assurent une plus grande densité de la représentation $3D$. $Z3$ est néanmoins un réseau très *pédagogique* qui permet de facilement ressentir les effets de la quantification. Il s'agit du réseau qui assure l'emboîtement optimal des réseaux tronqués. En effet, le réseau $Z3$ tronqué est un cube qu'il est possible d'emboîter, après changement d'échelle (ou réduction), en un autre cube (Cellule de *Voronoï* d'un réseau de résolution inférieure). Un autre aspect plaidant pour $Z3$ réside dans le fait que le fichier **cube** source est défini par une matrice régulière cubique.

2.2.3 Facteur d'emboîtement

La donnée d'entrée, la carte de densité électronique (EDM) est composée de valeurs positives de densités électroniques pour un volume donné. On peut donc représenter nos données comme un ensemble de cubes. Pour chaque cube est associé en son centre sa valeur de densité. La parité du facteur d'emboîtement influe sur les cellule de Voronoï (figure 2.2). Notre méthode s'intéressera aux facteurs d'emboîtement 2 et 3.

2.2.4 Résolution du fichier cube

La carte de densité électronique (EDM) représente des données moléculaires où les distances s'expriment en *Angström* (Å). Le pouvoir simplificateur de la QVAA n'a d'effet que lorsque les dimensions du réseau régulier de points sont supérieures à la distance entre deux valeurs de densité électro-

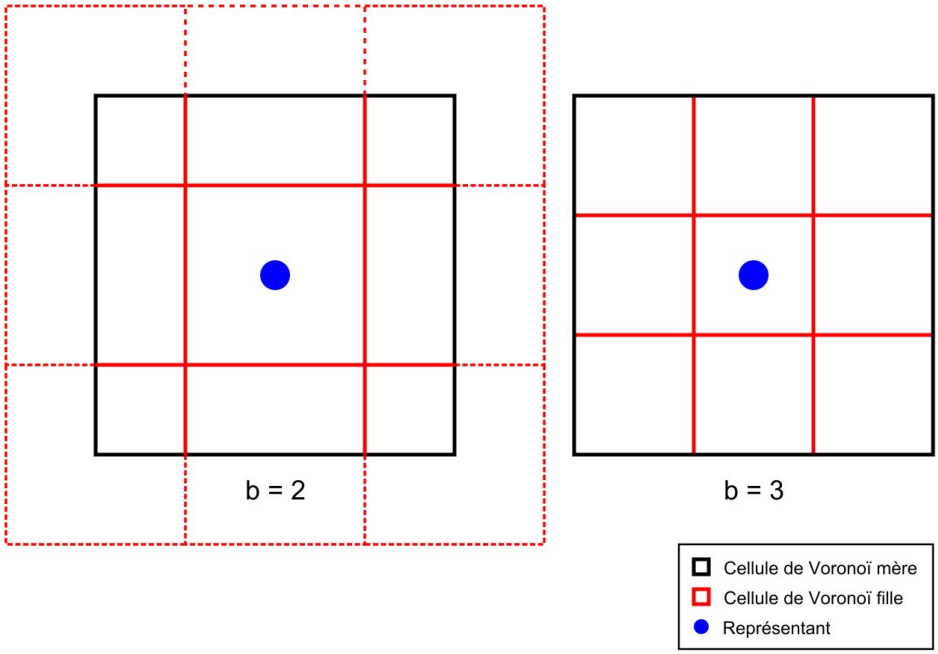


FIGURE 2.2 – Facteur d'emboîtement **b**, **pair** (à gauche) et **impair**(à droite).

nique contigües. Ainsi, il est très facile de déterminer une condition d'arrêt de la quantification, c'est à dire quand elle n'a plus d'effet (figure 2.3).

2.3 La modélisation

2.3.1 Introduction

Le but de ce mémoire implique la conception du prototype d'une interface (ou *framework*), qui permette la création de combinaisons de paramètres pour la recherche attenante à la quantification vectorielle algébrique et arborescente (QVAA) appliquée à des données moléculaires. La modélisation de ce projet est envisagé dans un contexte objet ce qui a naturellement porté notre choix sur un langage de modélisation tel que l'UML. Nous utiliserons

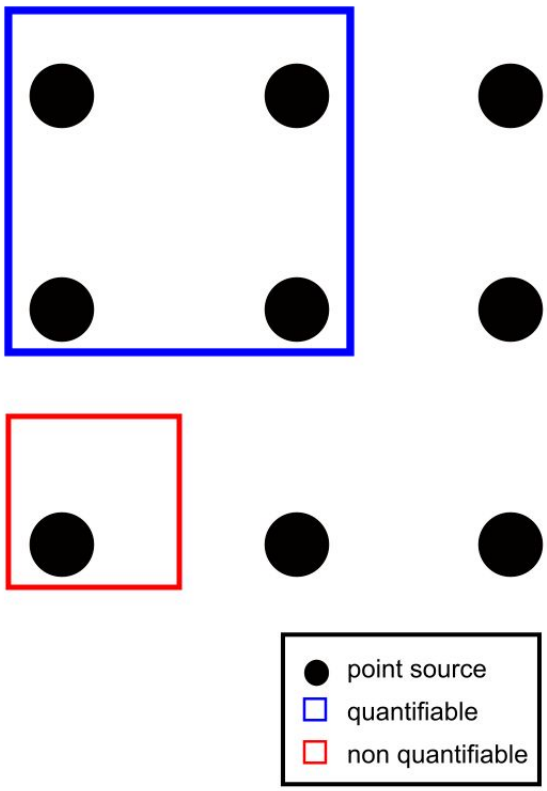


FIGURE 2.3 – Lorsque l'arête de la cellule de *Voronoï* est plus petit que la distance qui sépare deux points contigües, la quantification n'a plus aucun effet, il faut donc la stopper.

MATLAB pour le prototypage.

2.3.2 Conception globale

Une première approche du projet permet de formaliser le diagramme de cas d'utilisation 2.5. La première information à noter est que la lecture du fichier de donnée doit être dynamique. Un cas réaliste de fichier *cube* implique des téra-octets de donnée. Un processus indépendant de fourniture de donnée est nécessaire d'où la clause <*thread*> du cas **read .cube file** pour autoriser

un fonctionnement *multiprocessing*. La notion de *simplification* au sens de la quantification doit être implémentée de la façon le plus générique possible, la clause <*abstract*> sur le cas **simplify data** spécifie cette contrainte. En effet, il est évidement impensable de devoir redévelopper toute la démarche pour, par exemple, simplement modifier le type de quantification. Ici nous exécutons une QVAA représentée par le cas **simplify with TSLVQ**. Ce cas *hérite* du cas abstrait **simplify data**.
Le cadre du bas du diagramme de cas d'utilisation 2.5 décrit les cas d'interprétation purement chimique associés aux groupes fonctionnels qui ne seront pas traités dans le cadre de ce mémoire.

L'analyse du diagramme de cas d'utilisation 2.5 permet de dégager quatre activités globales. La figure 2.6 montre l'enchaînement de ces activités ; Tout d'abord **Data Loading** qui est tenu de charger et préformater les données des fichiers *cubes*. On poursuit avec l'activité de **Simplification** qui prend en charge la QVAA. Une étape intermédiaire demande alors, la mise en place d'une activité de mise en forme des données (un ou plusieurs fichiers *cubes*) à des fins de contrôle. Le cas **Human Control** offre à l'utilisateur le pouvoir de juger de la pertinence de la poursuite de la quantification. Le traitement se termine par **Critical Points detection**, la phase de détection de points critiques et de mise à disposition.
On peut alors étudier en détail les activités (figure 2.7) et les *packages* (figure 2.4) pour enfin identifier un flux de traitement, véritable *fil rouge* pour la modélisation et plus tard le développement.

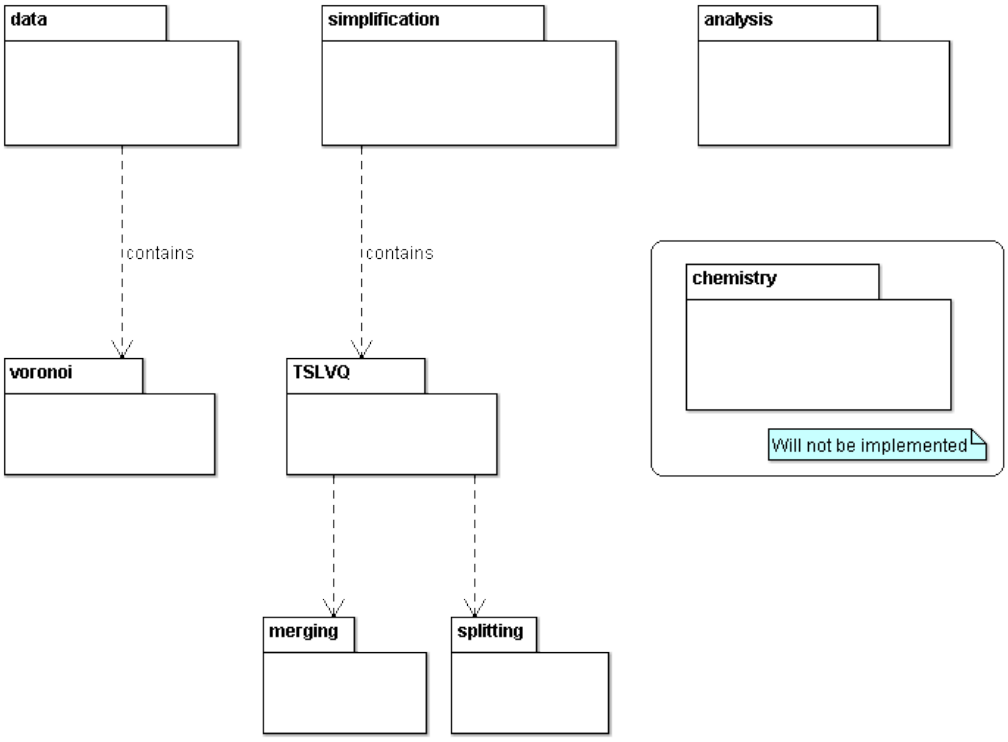


FIGURE 2.4 – Diagramme de package global.

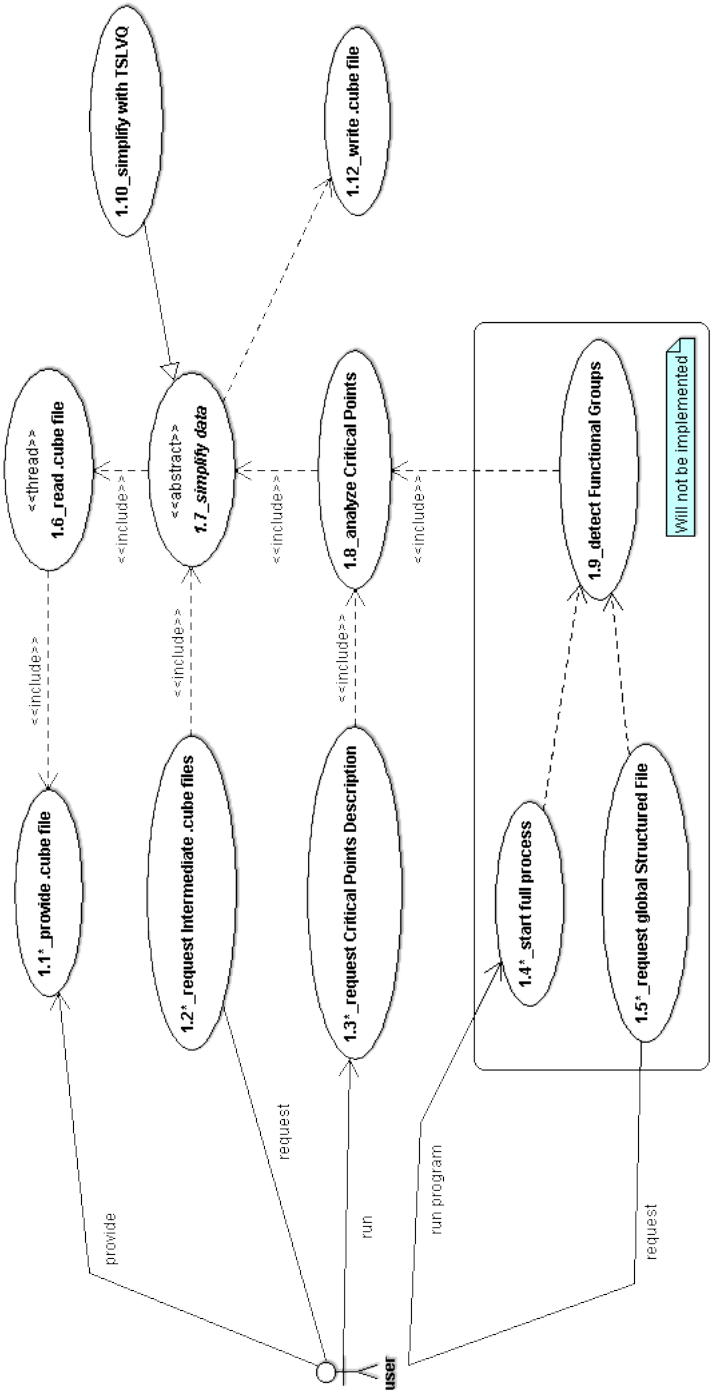


FIGURE 2.5 – Diagramme de cas d'utilisation du projet global.

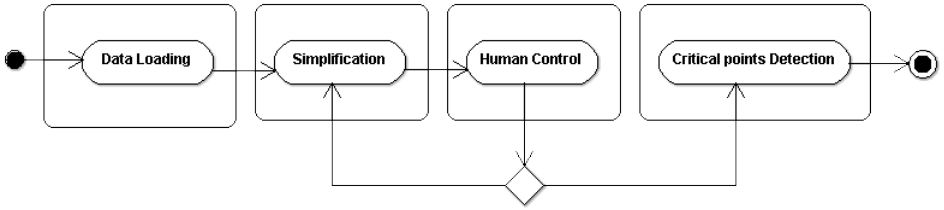


FIGURE 2.6 – Diagramme d'activité des étapes globales.

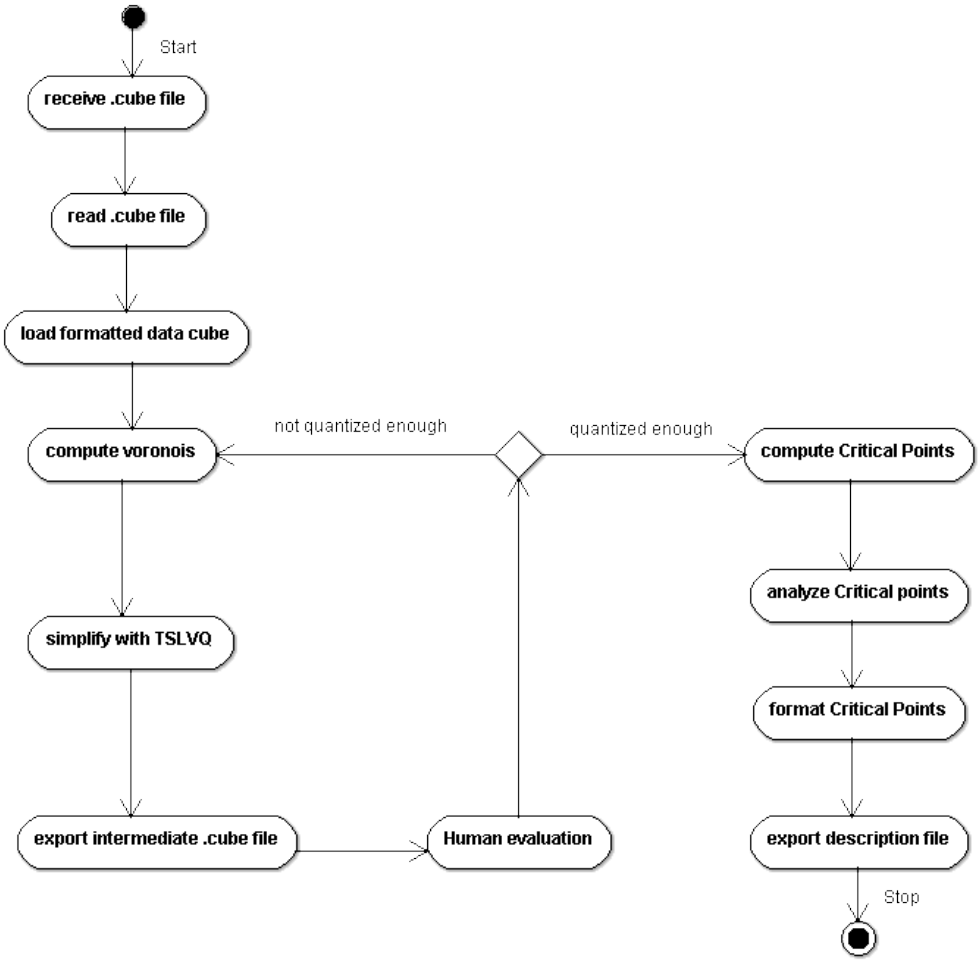


FIGURE 2.7 – Diagramme d'activité du projet global.

2.3.3 Démarrage du traitement

L'analyse globale précédente conduit à concevoir une architecture modulaire et réutilisable qui a la propriété d'être assez simple pour permettre aux chercheurs amenés à s'en servir de se focaliser sur le métier (ici la modélisation moléculaire) plutôt que sur le côté purement informatique de l'interface. C'est pourquoi, à l'exception de l'analyse de points critiques, chaque activité précédemment identifiée dispose d'un point d'entrée sous la forme d'une classe générique ou *abstraite*.

Le diagramme de classe 2.8 décrit **runMe**, une classe qui est le point d'entrée de tout le flux de traitement. On doit voir cette dernière comme la *tour de contrôle* du programme, elle est chargée d'enchaîner les étapes s'il y a lieu de le faire.

Issue du *package* **data**, le rôle de la classe abstraite **DataFileLoader** est de charger en mémoire, sans temps mort, le fichier *cube* quelle que soit sa taille. Cette classe est un conteneur sur le modèle d'un *vecteur* en Java. Elle fournit donc un itérateur qui permet, de façon continue, d'alimenter en donnée toute fonction métier qui le requiert.

La classe abstraite **Simplifier** membre du *package* **simplification** porte en elle toutes les fonctions de la QVAA, notamment, à l'évidence, la quantification vectorielle, la notion d'arbre de donnée et la gestion des cellules de *Voronoï* associées aux réseaux réguliers de points.

La classe **CriticalPointAnalyzer** issue du *package* **analysis** traite la détection, l'analyse et la valorisation des points critiques.

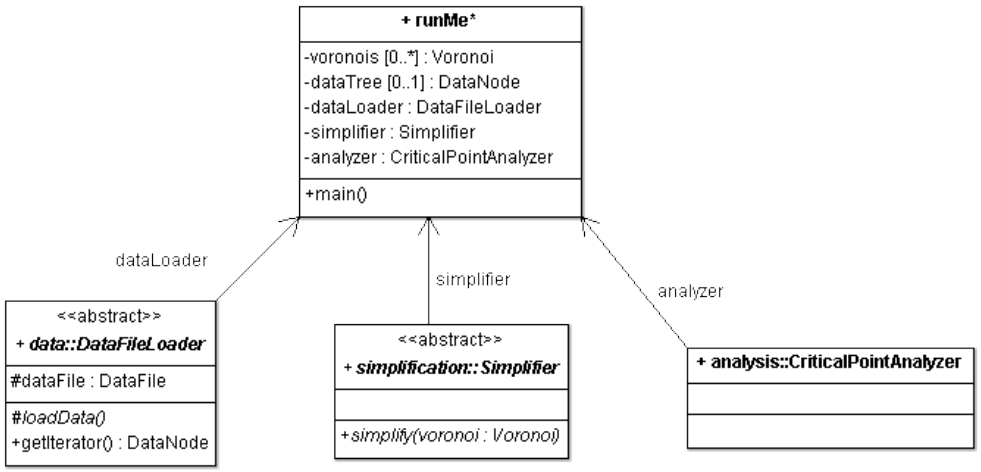


FIGURE 2.8 – Diagramme de classe de l'exécution du programme.

2.3.4 Lecture des données d'entrée

L'extraction des données sources passe par la lecture en flux continu de fichier *cube*. Le diagramme de cas d'utilisation 2.9 décrit ce processus multi-tâche dont la vitesse d'exécution influera directement sur celle du traitement du programme de façon critique. Le fil d'exécution (*thread*) **read .cube file** contrôle l'ouverture et la fermeture du fichier grâce aux sous-tâches **open .cubeFile** et **close .cubeFile**. En effet, la gestion de ces fonctions systèmes doit être transparente pour fournir des données de façon fluide. Un pointeur virtuel de donnée du cas d'utilisation **read DensityBuffer** fourni les données lues, sachant que cela implique d'avoir préalablement récupéré l'entête du fichier (**extract Header**) et chargé le tampon de lecture (**fill DensityBuffer**) avec des blocs de densité électronique (**load DensityData**).

Le diagramme de classe 2.10 permet d'aborder les détails de la fonction de lecture de donnée. On remarque de prime abord la stucture de donnée qui sera la véritable colonne vertébrale du programme. il s'agit d'une liste chaînée

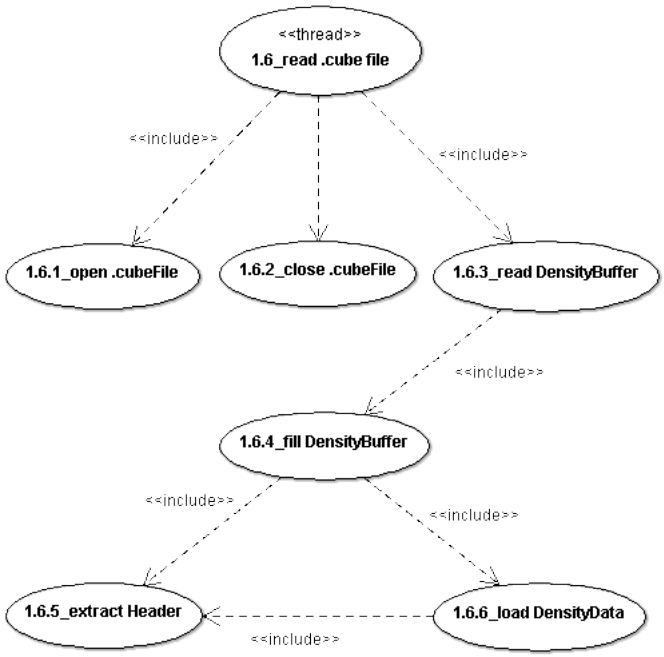


FIGURE 2.9 – Diagramme de cas d'utilisation de la lecture d'un fichier cube.

d'élément de la classe **DataNode** (*noeud de donnée*). Chaque noeud représente un point de nos données localisé dans l'espace (*location*). Il est caractérisé par un **volume** (le volume cubique dans lequel rayonne la densité), une **population** (le nombre d'électron dans le volume) et une valeur de densité électronique (**density**) (c.f. chapitre 2.5.4, page 77). De part leur conception, et notamment grâce aux attributs **parent** et **children**, ces noeuds agrégés en liste migrent très facilement vers une structure arborescente. Un dernier attribut complète cette description, **owner**, la cellule de *Voronoï* à laquelle appartiendra ce point après l'étape de la quantification.

Ainsi, nous découvrons la classe *abstraite* **Voronoi** au sein des diagrammes 2.10 et 2.11. Tout point de densité quantifié appartient à une cellule de *Voronoï*. Cette dernière est régulière et de part sa structuration arborescente,

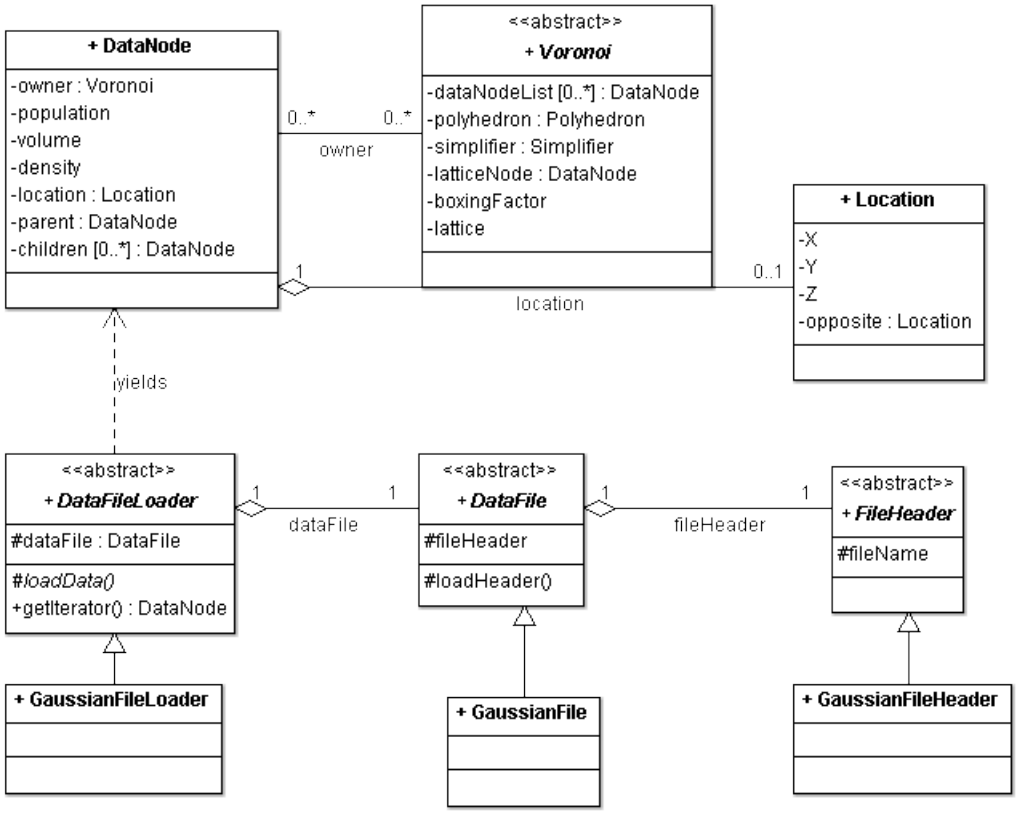


FIGURE 2.10 – Diagramme de classe du chargement des données.

elle devient le support du caractère multi-résolution de notre méthode. Elle contient donc une liste de noeuds de donnée (**dataNodeList**), le polyèdre (sa forme et sa taille) de la cellule de *Voronoï* (**polyhedron**). Cette classe est par ailleurs pourvue d'un outil de simplification (ou quantification) (**simplifier**). Ce dernier membre et **polyhedron** sont *abstraits* pour permettre d'étendre facilement la bibliothèque des quantificateurs et des cellules de *Voronoï*. On termine cette description en indiquant le dual de la cellule de *Voronoï*, à savoir son centre, le représentant du réseau régulier de points correspondant (**latticeNode**).

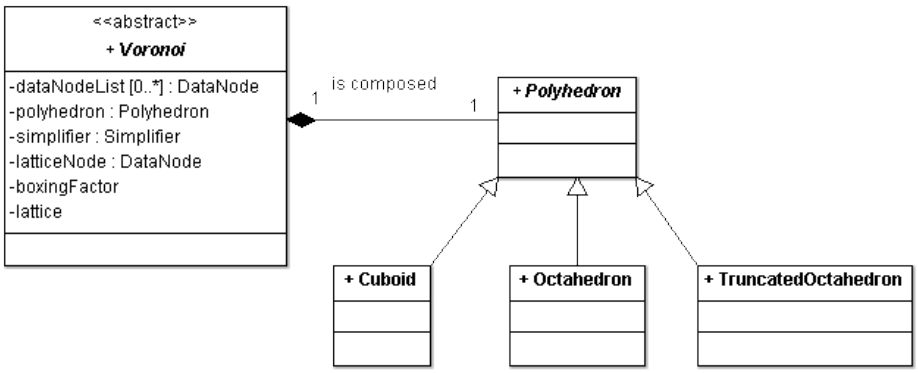


FIGURE 2.11 – Diagramme de classe de la gestion des *Voronoïs*.

La lecture du fichier proprement dite est conçue de façon *générique* pour ne pas verrouiller le programme sur l'usage seul de fichier **cube** (aussi appelé **GaussianFile** car issu du logiciel *Gaussian*). Le processus se découpe en trois classes *abstraites* ; La classe **DataFileLoader** qui charge la mémoire avec les données et met à disposition un *itérateur*, la classe **DataFile** qui représente le fichier et la classe **FileHeader** qui symbolise l'entête du fichier lu.

2.3.5 Simplification avec la QVAA

La phase d'exécution de la QVAA est modélisée sur le diagramme de cas d'utilisation 2.12. Notre démarche impose d'implémenter la QVAA (ou TLSVQ pour Tree-Structured Vector Quantization) en mode *découpage* (splitting). C'est à dire que la construction et l'optimisation de l'arbre de donnée sont réalisés au fur et à mesure de la quantification.
Le cas d'utilisation **run Splitting** se charge d'exécuter le traitement de quantification vectorielle. Lorsqu'un point (ou son volume associé) est quantifié, il est ajouté dans une structure arborescente (**design dataTree**) pour signifier les caractéristiques de la représentation rencontrée. En effet, la position, la valeur et le volume représentés sont des propriétés de l'arbre de donnée ainsi créé. Cet arbre est composé d'éléments de la classe **DataNode** comme décrit sur le diagramme de classe 2.13). Les méta-données de l'arbre (**population**, **volume**, **density**, **location**) sont traitées par le cas **compute dataTree meta-data**.
Ainsi la position quantifiée devient celle du centre de la cellule de *Voronoï*, c'est à dire la position du représentant (élément du réseau régulier de points choisi). Le volume quantifié est la somme des volumes symbolisés par les points sources intégrés à la cellule de *Voronoï* en cours. De même pour la densité électronique que l'on quantifie par la moyenne des valeurs des densités électroniques de tous les points sources (c.f. chapitre 2.5.5, page 78).
Lorsque toutes les données ont été quantifiées jusqu'à atteindre la résolution souhaité ou pertinente, on dispose d'un arbre complet qui, très probablement, ne sera pas *pas équilibré*. On sauvegarde alors l'arbre de donnée complet sous la forme d'un fichier **XML**. Ce fichier sera la description de notre volume tridimensionnel. On exporte alors sous forme d'un fichier **cube** standard, chaque niveau de quantification, c'est à dire pour un niveau donné, tous les

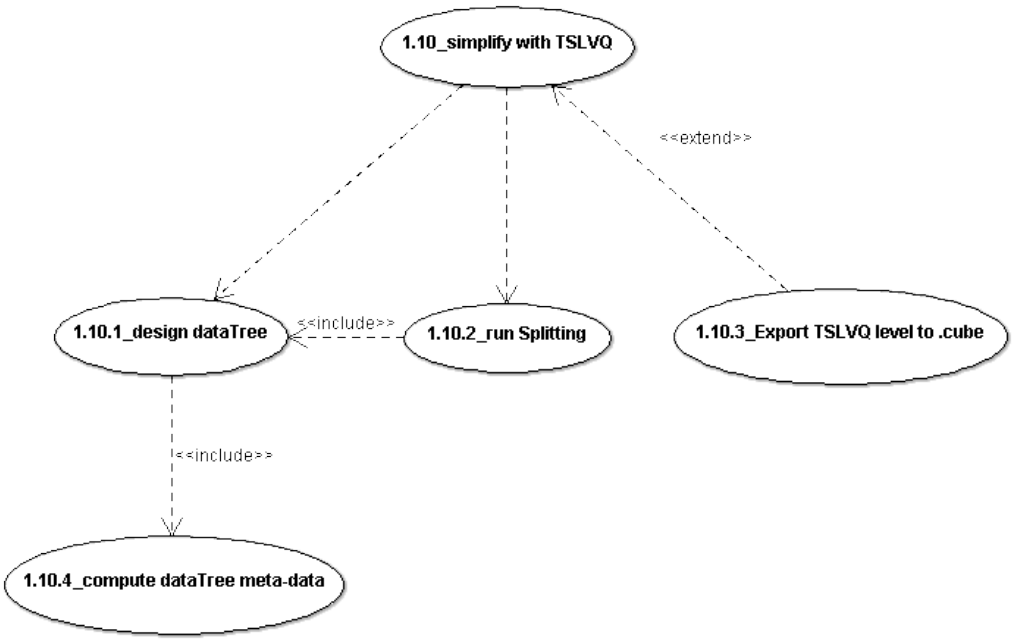


FIGURE 2.12 – Diagramme de cas d'utilisation de la QVAA.

noeuds de l'arbre qui sont au même niveau.

2.3.6 Exportation de fichier cube

Le déroulement de l'exportation des résultats sous forme de fichiers *cube* est décrite sur le diagramme de cas d'utilisation 2.14. On remarque tout d'abord des tâches communes à la lecture des données d'entrée (cf. 2.3.4 page 62) (**open .cubeFile** et **close .cubeFile**). Il faut aussi noter que l'écriture de l'entête (**write header**) est désolidarisée de l'écriture des données de densité électronique (**write data**). L'entête à écrire, grâce à la tâche (**generate Header**) doit être généré à partir de l'entête du fichier source en tenant compte des transformations algébriques et géométriques induites par la quantification. De même pour les données, les valeurs de densité électronique qui sont agglomérées par le processus de quantification doivent être

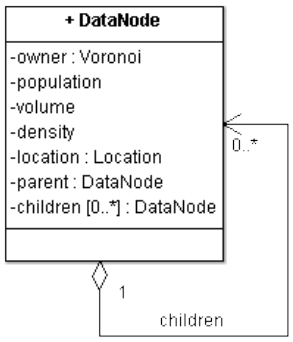


FIGURE 2.13 – Diagramme de classe de l'arbre de donnée.

reformatées sous la forme de texte ASCII (**format data**). Lors du parcours exhaustif de l'arbre de donnée (**browse dataTree**), la tâche (**align data on grid**) doit, si nécessaire en fonction de la résolution, les aligner sur la grille tridimensionnelle en accord avec résolution indiquée dans l'entête à écrire. Plus précisément, l'exportation à la résolution la plus fine correspondant, dans l'arbre de données, au niveau qu'on veut exporter.

Le diagramme de classe 2.15 décrit le structure de l'architecture d'écriture de fichier **cube** (ou export de données). Il est calqué sur le diagramme de classe de la lecture de fichier la lecture (2.10). L'écriture du fichier est aussi conçue de façon *générique* pour la même raison, à savoir, ne pas verrouiller le programme sur l'usage seul de fichier **cube**. Le processus se découpe en trois classes *abstraites* ; La classe **DataFileWriter** qui lit les données dans l'arbre (attribut **tree**), requiert le formatage et écrit les données dans le ficher cible. Avec toujours, la classe **DataFile** qui représente le fichier et la classe **FileHeader** qui symbolise l'entête du fichier à écrire.

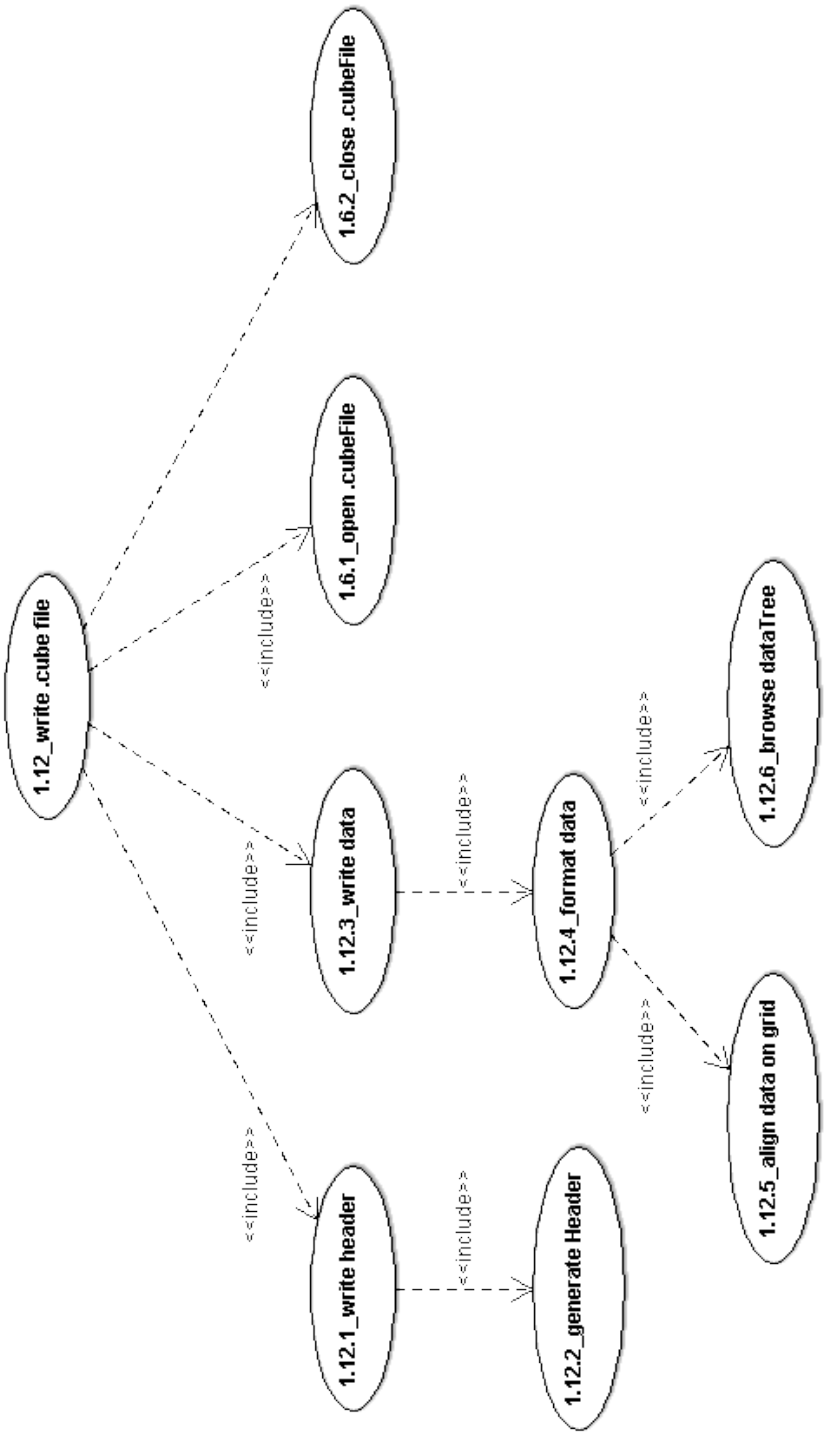


FIGURE 2.14 – Diagramme de cas d'utilisation de l'écriture d'un fichier cube.

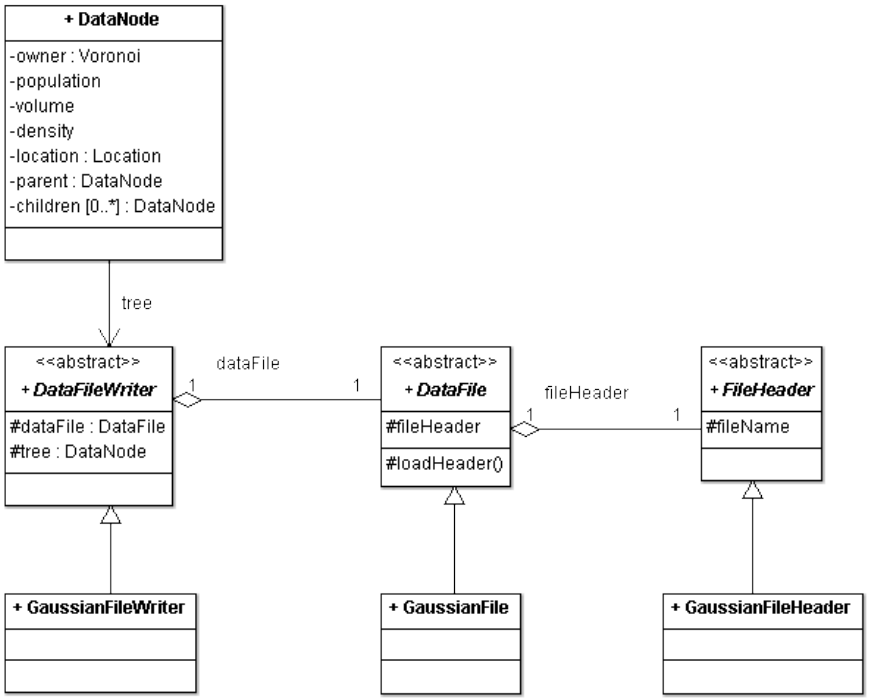


FIGURE 2.15 – Diagramme de classe de l'export de données (écriture de fichier **cube**).

2.3.7 Détection et analyse de points critiques

Pour caractériser géométriquement le volume, il est nécessaire de détecter et d'identifier la nature des points critiques (CP). Or les densités électroniques sont présentées sous forme d'une carte de densité électronique (EDM). Ces données, compte tenu de la résolution choisie au sein du logiciel *Gaussian*, représente plus ou moins finement une molécule. Dans le cas continu, la littérature base le traitement de ces opérations sur le calcul de gradients et de matrices hessiennes. Le cas discret, et donc notre traitement propre, nous impose de simuler cette démarche tout en conservant le principe.
Comme indiqué sur le diagramme de cas d'utilisation 2.16, la recherche de points critiques suit trois étapes ; Tout d'abord, la détection de points critiques (**search for CP**) grâce au calcul de gradients. Lorsque qu'un CP est trouvé, on détermine alors sa nature en évaluant la matrice Hessienne (**compute Hessian matrix**) et on termine par sauvegarder le point dans le fichier de description XML (**generate description file**) qui contient l'arbre de donnée sauvegardé (cf. 2.3.5 page 66). A l'issue de ce processus, on dispose d'un arbre très riche en information que le spécialiste en modélisation moléculaire est amené à analyser d'un point de vue strictement *métier*.
Le diagramme de classe 2.17 propose l'architecture de la méthode de détection et de caractérisation de points critiques. Chaque niveau de quantification (on rappelle que pour un niveau donné, on prend tous les noeuds de l'arbre qui sont au même niveau) est une liste de points de densité électronique (ou éléments de la classe **DataNode**). Sur cette dernière un ensemble de gradients puis un ensemble de matrices hessiennes (classe **HessianMatrix**) sont calculés et analysés (classe **CriticalPointAnalyzer**). Les points qualifiés en tant que *point critique* sont marqués. Lorsque le traitement est terminé, on obtient une arbre de points critiques grâce la liaison arborescente (membres

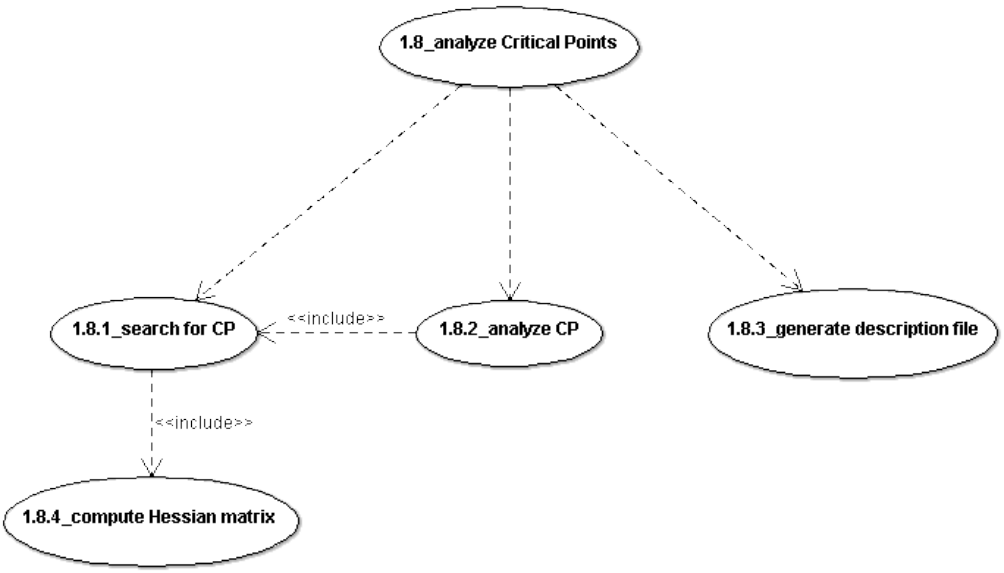


FIGURE 2.16 – Diagramme de cas d'utilisation de l'analyse de points critiques.

parent et **children**) de la classe **DataNode**.
Pour analyser la représentation géométrique de la molécule, le spécialiste en modélisation moléculaire n'a alors plus qu'à parcourir l'arbre (ou le fichier de description) au niveau de la résolution désirée.

2.4 Environnement technique

2.4.1 Matériel

Ce projet a été développé sur un PC *Intel Core duo E840* 3Ghz avec 4 Go de RAM, sous le système d'exploitation Windows Vista 32 bits. Les processus de quantification sur données réelles, particulièrement gourmands en ressource, ont été lancés sur un PC *Intel Xeon Octo-Core E5410* 2.33 Ghz avec 8 Go de RAM, sous système d'exploitation Windows XP Professionnal

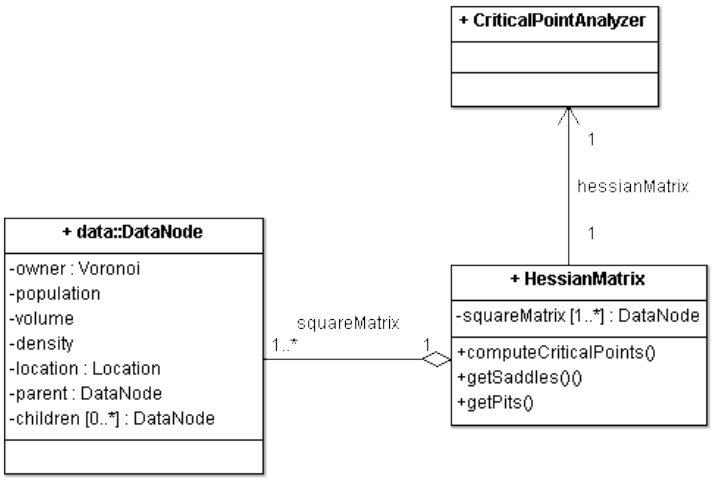


FIGURE 2.17 – Diagramme de classe de l'analyse de points critiques.

64 bits Version 2003 Service Pack 2.

2.4.2 MATLAB

Ce programme a été développé sous le logiciel *MATLAB*[1] version *R*2008*b*. *MATLAB* est un environnement de développement qui dispose de son propre langage de programmation. Il est développé et commercialisé par la société américaine MathWorks. Ce logiciel est utilisé comme outil de calcul numérique dans divers domaines tel que la recherche, l'éducation, ou l'industrie. On l'utilise aussi comme support de prototypage pendant les phases de développement de projets. La philosophie de l'outil est basé sur un jeu extensible de boîtes à outil (ou **toolbox**) qui couvre toutes sortes de domaines qui vont du traitement d'image au calcul statistique en passant par l'aéronautique et la bioinformatique.

Dans le cadre de ce projet, *MATLAB* nous apporte particulièrement des

1. `http://www.mathworks.com`

boîtes à outils graphiques notamment pour l'affichage tridimensionnel des données.

2.4.3 Jmol

Jmol[2] est un outil *open source* de visualisation de molécules conçu pour le monde de l'éducation et les chercheurs en chimie et en biochimie. Il est multi-plateformes et disponible sour trois formes différentes : L'application *Jmol*, une application Java que nous avons utilisée pour ce projet. *JmolApplet*, une applet pouvant être intégrée dans des pages web et *JmolViewer*, un kit de développement pour l'interfaçage à d'autres applications Java. *Jmol* est pourvue d'un langage de script simple et très utile basé sur *javascript*[3].
Le logiciel *Jmol* nous a servis à visualiser simplement les fichiers **.cube** avant et après quantification.

2.4.4 Editeur de texte

La lecture de fichier de données source et résultat requiert un éditeur de texte capable d'ouvrir des fichiers de très grands volumes. De nombreuses références peuplent le monde du développement et pour Microsoft Windows nous avons fait le choix de *Notepad++*[4] qui a montré de très grande qualité sur des fichiers XML de plusieurs millions de lignes.

2.4.5 LaTeX

Ce mémoire a été rédigé sous le logiciel de traitement de texte LaTeX. Il s'agit d'un langage de composition de texte très utilisé dans le monde libre

2. `http://jmol.sourceforge.net/index.fr.html`
3. `http://chemapps.stolaf.edu/jmol/docs/`
4. `http://notepad-plus-plus.org/`

et celui de la recherche pour l'écriture de documents scientifiques employant le *processeur de texte* TEX. Il est spécialement utilisé dans les mondes scientifiques et techniques pour la création de documents de taille plus ou moins importante tels des livres ou des thèses. Toutefois, on l'emploie aussi pour concevoir des documents moins conséquents (des lettres, des affiches, ou des transparents) [OPHS11].

2.5 Développement

2.5.1 Introduction

La réalisation de ce programme s'appuie sur la modélisation précédente, le découpage en *packages* a dirigé la succession des développements de chacune des parties. Néanmoins pour des contraintes d'adaptation au langage de *MATLAB*, la vision objet n'a parfois pas été calquée en l'état car la puissance des *toolboxes* de l'outil permet de s'en abstraire. En effet, par exemple avec *MATLAB* la lecture de données est proposée de base et de façon optimisée. D'où la réserve suivante qui consiste à rappeler que ce projet est fortement orienté *recherche fondamentale*. Son but est de produire un prototype qui puisse donner des résultats, au mieux, probants.

2.5.2 Architecture

Intéressons-nous pour commencer à l'architecture globale de programme ; Elle se dégage assez naturellement de la modélisation et notamment du diagramme de *package* (figure 2.4, page 58). Le programme se décompose en quatre groupements principaux :

dataLoading regroupe toutes les entités et de fonctions relatives à la lecture

et au chargement en mémoire des données sources.

simplification contient l'ensemble des entités et des fonctions relatives à la quantification vectorielle algébrique et arborescente (QVAA).

criticalPoint s'occupera de la détection et de la caractérisation des points critiques.

humanControl gère la mise en forme et l'export des données résultats vers les fichiers **cube** et **XML** dans le but de les rendre compréhensibles à un œil humain, en l'occurence, celui du spécialiste en modélisation moléculaire.

Ajoutée à ces groupes, une classe de lancement **runMe** sert de point d'entrée lors de l'exécution du programme.

2.5.3 Démarrage du traitement

Le traitement s'amorce par la classe **runMe** qui prend respectivement en paramètres d'entrée les éléments suivants :

cubeFile : Une chaîne de caractères qui indique le fichier **cube** des données source.

latticeModel : Une chaîne de caractères qui spécifie le modèle de réseau régulier de points à utiliser lors de la quantification vectorielle algébrique et arborescente (QVAA). À savoir $Z3$ pour le réseau cubique, $D3$ pour le réseau dont la cellule de *Voronoï* est un Dodécaèdre Rhombique, et $D3*$ pour le réseau dont la cellule de *Voronoï* est un Octaèdre Tronqué (figure 1.24, page 37).

lvlMax : Un nombre entier. La nature arborescente de notre méthode de quantification vectorielle a influencé la contruction récursive de l'arbre de donnée. Ainsi, cela impose de limiter le nombre de quantifications

successives afin d'éviter un temps de traitement infini. Toutefois, la définition de nos données donne un élément d'arrêt naturel de quantification lorsque la résolution atteinte est inférieure ou égale à la résolution imposée dans le fichier **cube** (c.f. chapitre 2.2.4, page 54).

b : Un nombre entier qui représente le facteur d'échelle (ou d'emboîtement) du processus algébrique de notre méthode de quantification (figure 1.26, page 39).

outFile : Une chaîne de caractères qui indique le préfixe des fichiers **cube** et **XML** des données de sortie.

La première tâche de la classe de lancement sera de charger le fichier **cube** source selon l'algorithme décrit chapitre 1.1.12, page 27. Il faut alors linéariser les données pour leur enlever en mémoire leur structure tridimensionelle. Ce qui pour autant ne supprime pas l'information spatiale, les coordonnées géométriques sont conservées, simplement plus sous la forme d'un cube tridimensionnel mais plutôt dans un conteneur de type *vecteur*. Cette nouvelle structure de donnée représente notre *signal source*, une liste de *noeud de donnée* (**dataNode**).

2.5.4 Lecture de données

La fonction **readcube**, qui appartient au groupement (ou *package*) **dataLoading**), est chargée d'extraire et de charger les données à partir du fichier **cube**.

Elle prend en paramètre d'entrée une chaîne de caractères pour le nom du fichier **cube** source (**fname**), et un booléen (**dataIsDensities**) qui spécifie si les données du fichier sont de type densité électronique (*dataIsDensities* = **true**) ou potentiel électrostatique (*dataIsDensities* = **false**).

Cette fonction retourne une structure (un vecteur de vecteurs) :

$$\begin{bmatrix} Density3D & X & Y & Z & NumAtoms & Header \end{bmatrix}$$

où **Density3D** est notre signal source sous la forme d'une matrice tridimensionnelle. **X, Y et Z** sont les vecteurs coordonnée calculés relativement à la position du centre du fichier **cube**, l'origine du repère. **NumAtoms** est le nombre d'atomes contenu dans le fichier, et **Header** une structure qui recense tous les éléments qui composent l'entête du fichier.

2.5.5 Simplification avec la QVAA

La quantification vectorielle algébrique et arborescente (QVAA) est effectuée sur des données vectorisées au sein de la classe de lancement. La fonction **TSLVQ** (figure 2.18), membre du package **simplification** est le point d'entrée de la quantification. Cette fonction prend respectivement en paramètres d'entrée les éléments suivants :

Header : Une structure qui contient l'entête du fichier **cube** des données sources. Ces informations indiquent comment lire et évaluer les données.

$$Header = \begin{bmatrix} commentaire \\ origine \\ voxels \\ atomes \end{bmatrix}$$

avec

$$commentaire = \begin{bmatrix} comment_1 & comment_2 \end{bmatrix}$$

$$origine = \begin{bmatrix} nbAtomes & origineX & origineY & origineZ \end{bmatrix}$$

$$voxels = \begin{bmatrix} nbVoxelX & longueur_{x1} & longueur_{y1} & longueur_{z1} \\ nbVoxelY & longueur_{x2} & longueur_{y2} & longueur_{z2} \\ nbVoxelZ & longueur_{x3} & longueur_{y3} & longueur_{z3} \end{bmatrix}$$

où $nbVoxel$ est le nombre de voxels selon la dimension X, Y ou Z.

$$atomes = \begin{bmatrix} na_1 & na_1 & posX_1 & posY_1 & posZ_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ na_{nbAtomes} & na_{nbAtomes} & posX_{nbAtomes} & posY_{nbAtomes} & posZ_{nbAtomes} \end{bmatrix}$$

où na est le numéro atomique de l'atome décrit.

data : Une matrice des données sources de dimensions $4 \times n$ avec le nombre de points $n = x_{max} \times y_{max} \times z_{max}$.

$$data = \begin{bmatrix} x_0 & \dots & x & \dots & x_{max} \\ y_0 & \dots & y & \dots & y_{max} \\ z_0 & \dots & z & \dots & z_{max} \\ density_0 & \dots & density & \dots & density_n \end{bmatrix}$$

lattice : Attribut équivalent à **latticeModel** du chapitre 2.5.3.

lvlRequested : Attribut équivalent à **lvlMax** du chapitre 2.5.3.

b : Attribut identique à celui du chapitre 2.5.3.

outFile : Attribut identique à celui du chapitre 2.5.3.

Après des contrôles de validité des paramètres d'entrée, on calcule le **facteur de normalisation** f [Ric96] grâce auquel on vérifie que le niveau maximum de quantification demandé (**lvlRequested**) ne conduit pas à travailler à une résolution plus fine que celle du fichier **cube**. Si une erreur est détectée, on calcule le nombre maximal de quantifications successives possibles que l'on

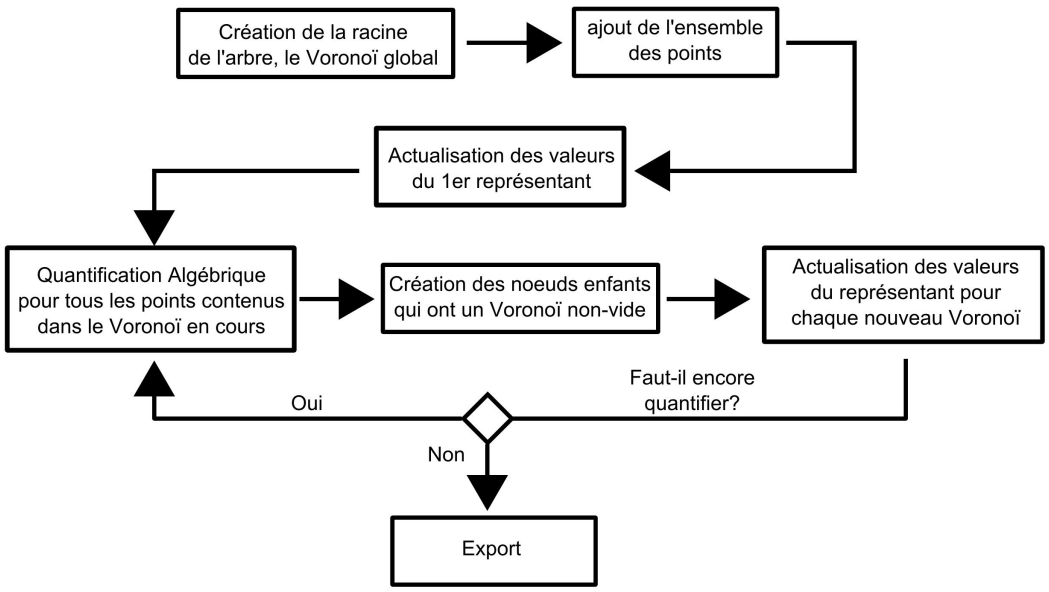


FIGURE 2.18 – La quantification vectorielle algébrique et arborescente (fonction **TSLVQ**).

substitue au niveau demandé. On applique le facteur de normalisation à nos données source, dès lors on peut commencer à quantifier.

On initialise la racine de l'arbre de donnée, ce dernier servira à stocker les résultats successifs de la quantification. Ce noeud racine est à ce niveau le représentant et le seul de l'ensemble des points source. Il s'agit du point le plus quantifié, celui qui est au centre de la cellule de *Voronoï* qui englobe tous les points de la source.

On peut alors lancer **recursiveQuantization**, la fonction récursive de quantification vectorielle algébrique (QVA) (figure 2.19). Elle prend en paramètres d'entrée :

tree : La cellule de *Voronoï* à quantifier qui est (ou sera) un noeud de l'arbre de donnée.

ptr_quantif : Le pointeur sur la fonction de quantification qui dépend du réseau régulier de points requis au démarrage du programme.

lvlQuantif : Le niveau courant de quantification. (le traitement est récursif !)

lvlQuantifMax : Le niveau de quantification à atteindre (et à ne pas dépasser)

boxingFactor : Le facteur d'emboitement de la QVAA.

Elle retourne **tree** qui, suite au traitement, est un sous-arbre de quantification.
En considérant l'exemple de la figure 2.19, de part la définition de la quantification vectorielle algébrique (QVA) avec le réseau régulier de points $Z3$, les représentants ont des coordonnées entières. La quantification dans ce cas consiste alors à **arrondir à l'entier le plus proche chacune des coordonnées** du point source considéré.

Chaque phase de quantification s'achève par une étape d'actualisation de attribut de chaque représentant :

- Le **volume** $V_r = \sum V_{fi}$ où V_{fi} est le volume des i cellules de *Voronoï* filles
- La **population** d'électrons $P_r = \sum P_{fi}$ où P_{fi} est la population d'électrons des i cellules de *Voronoï* filles
- La valeur de la densité électronique (**density**) $D_r = moyenne(D_{fi})$ où D_{fi} est la densité électronique des i cellules de *Voronoï* filles.

2.5.6 Exportation de fichier cube

Lorsque le processus de quantification est achevé, nous disposons d'un arbre de donnée complet en mémoire qui est alors désigné par la variable

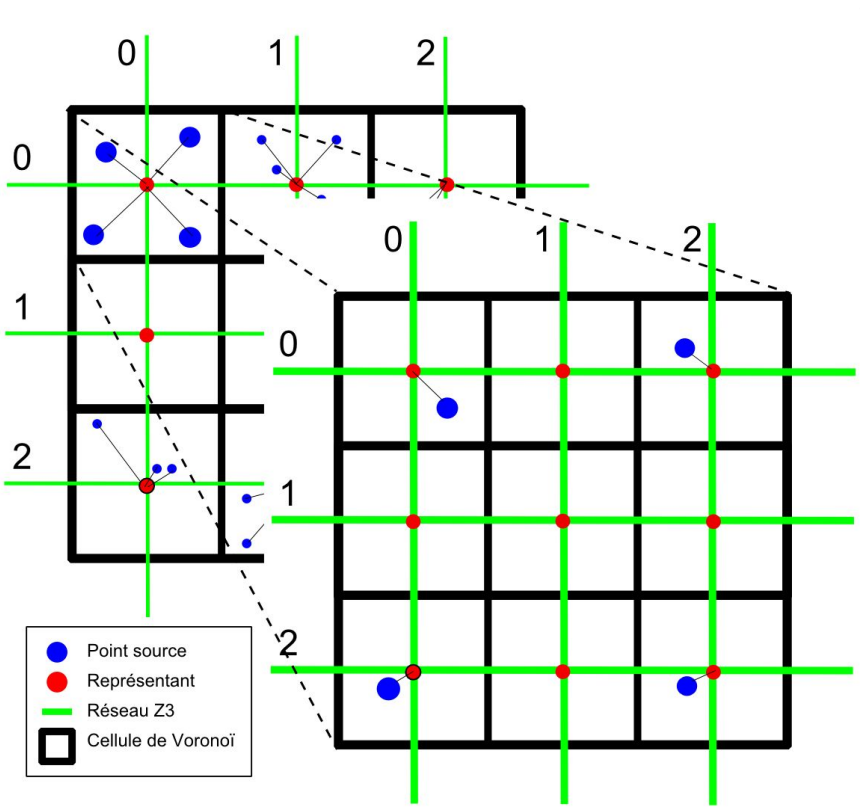


FIGURE 2.19 – La fonction **recursiveQuantization** effectue successivement des quantifications algébriques jusqu'à atteindre une description conforme à la résolution souhaitée (ou possible).

root. Cet arbre décrit la totalité des résolutions de quantification c'est pourquoi nous le sauvegardons en entier dans un fichier au format XML pour conserver de façon native la vision arborescente.
Par ailleurs, le travail du spécialiste en modélisation moléculaire consistera à explorer les volumes quantifiés. Pour cela nous exportons chaque niveau de quantification (ou résolution) sous la forme d'un fichier **cube** standard, c'est à dire un fichier où les données quantifiées ont été alignées sur la grille tridimensionnelle équivalente, dans une résolution donnée, à celle du fichier source

(figure 2.20). Il s'agit d'une contrainte à toujours respecter pour créer une fichier **cube** lisible. Un fihcier **cube** contient toujours des données *homogènes* au sens où ; les données sont toutes Le format **cube** paie sa simplicité par la nécessité de créer un fichier de sortie ayant exactement la même structure que le fichier source. Ainsi, avant d'enregistrer les données, il faut réaligner les données selon une matrice tridimensionnelle, propotionnellement aux dimensions indiquées dans l'entête du fichier source et dont toutes les positions contiendront désormais les valeurs des représentants calculés lors de la phase de quantification.

N.B : Il est possible de créer un fichier **cube** d'une résolution *physique* différente de celle du fichier source à condition de convertir tous les représentants à la résolution à laquelle l'export est effectué (figure 2.21).

Le *package* **humanControl** contient deux fonctions d'export que sont **exportTree2XML** et **exportTree2Cube**. La première reçoit l'arbre et le nom du fichier, et produit directement le fichier XML de l'arbre. La seconde fonction reçoit en plus, l'entête du fichier **cube** source et le niveau de quantification souhaité.

2.5.7 Détection et analyse de points critiques

La caractérisation des différents volumes quantifiés passe par l'implémentation d'une détection de points critiques. Ces points sont des voxels dispersés dans l'espace tridimensionnel qu'il faut comparer à leurs voisins directs (la relation de voisinage est définie en $2D$ par la figure 2.22) et généralisée en $3D$ à la figure 2.23. La comparaison est un calcul de variation sur les valeurs de densité électronique que l'on peut voir comme un calcul de gradient dans le cas discret.

La fonction prend en paramètre d'entrée l'arbre de description sachant que

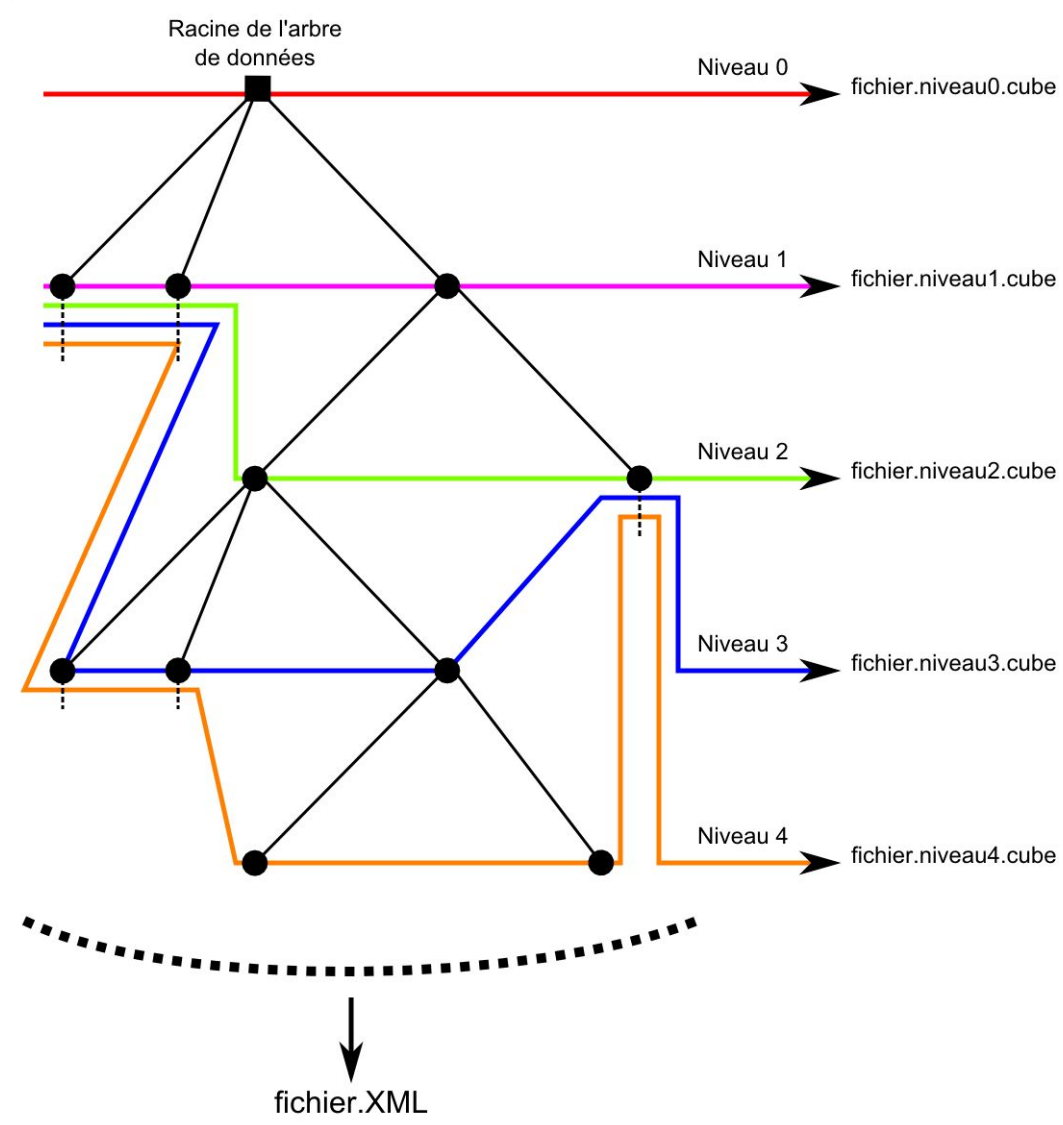


FIGURE 2.20 – A l'issue du processus de quantification l'arbre complet est sauvegardé au format XML et chaque niveau de l'arbre est exporté sous la forme d'un fichier cube dédié.

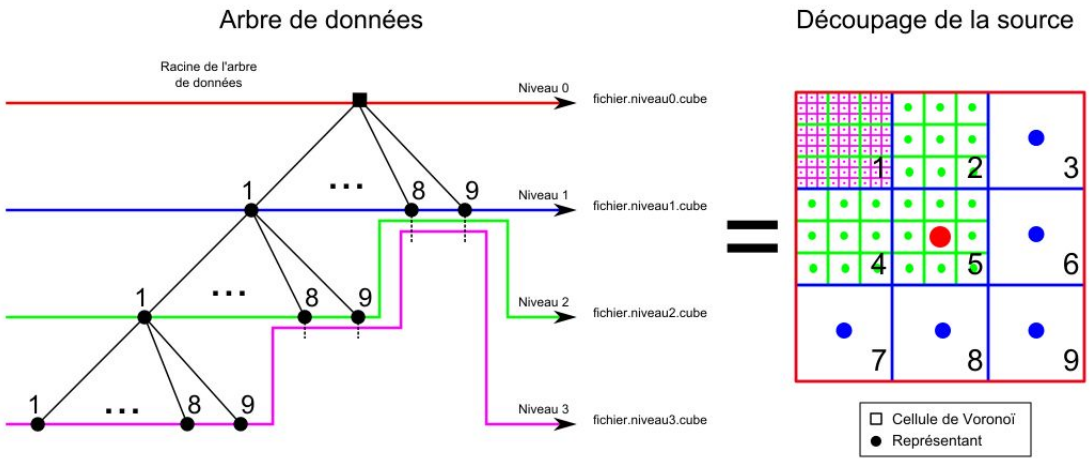
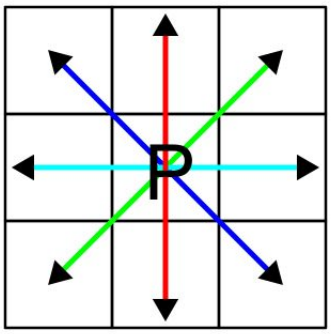


FIGURE 2.21 – d'un fichier cube dédié.



En 2D, 4 directions passent par le point central P

FIGURE 2.22 – En $2D$, on détecte les points critiques en calculant la variation de densité sur le carré central **P** par rapport à ses carrés voisins dans les 4 directions. Il s'agit d'un *8-voisinage*.

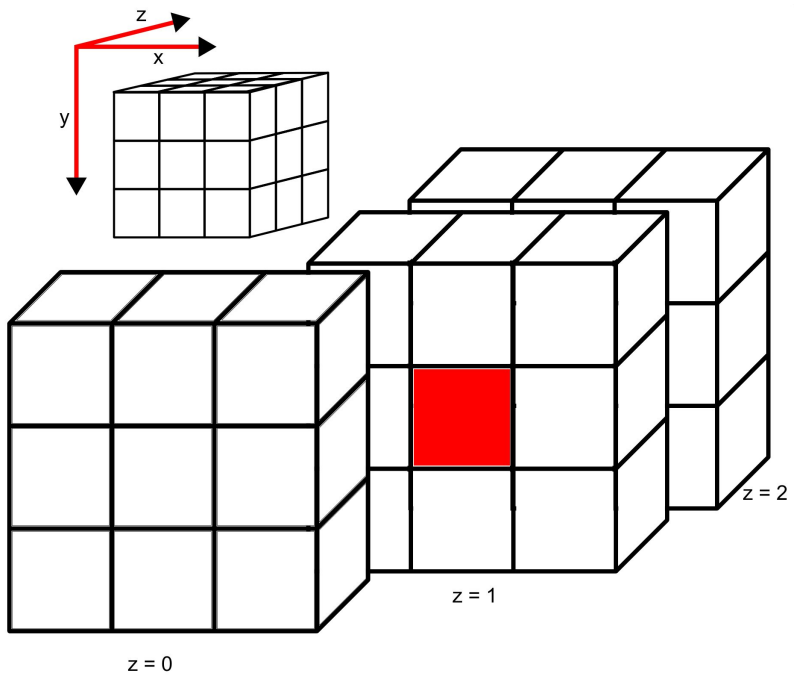


FIGURE 2.23 – En $3D$, la détection de points critiques est effectuée par calcul de variation de densité sur le voxel central (en rouge) par rapport à tous ses voxels voisins dans les 13 directions tridimensionnelles. Il s'agit d'un *26-voisinage*.

une seule résolution est explorée à la fois, ainsi le niveau de résolution doit être spécifié. Le programme cherche alors les points concernés et compare chaque point considéré selon chacune des 13 directions qui passe par ce point central. Une direction se définie donc comme un axe passant par le point central et deux voisins symétriques.

En fonction des valeurs rencontrées chez les voisins du point P considéré (figure 2.24), il peut ne pas être remarquable et donc ignoré, autrement on le qualifie de *pic*, *creux*, *selle*. Les figures 2.25 et 2.26 présentent des tests unitaires sur des cas simples en $2D$ de points critiques intuitifs. Ces points

ainsi que leurs coordonnées sont sauvegardés dans des listes par type de point critique.

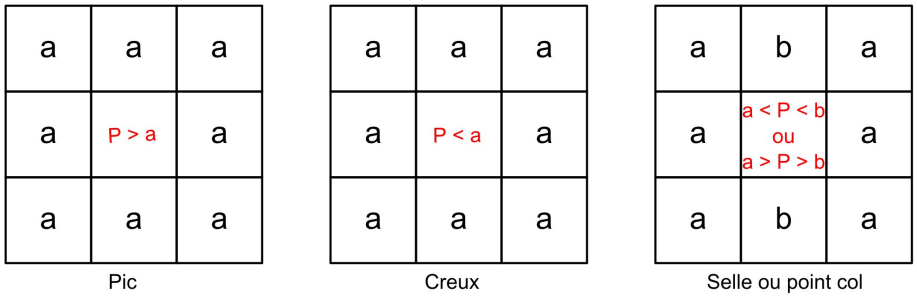


FIGURE 2.24 – Exemple en 2D (quatre directions) des trois types de points critiques P.

Lorsque tous les points critiques ont été détectés et caractérisés, afin de conserver cette précision de l'information spatiale nécessaire au spécialiste en modélisation moléculaire, il faut mettre à jour le fichier XML de description pour indiquer directement la nature de chaque point.

Pour finir, il important de garder à l'esprit qu'un point peut être critique à une résolution particulière de la quantification sans pour autant l'être dans les résolutions de quantification précédentes ou suivantes.

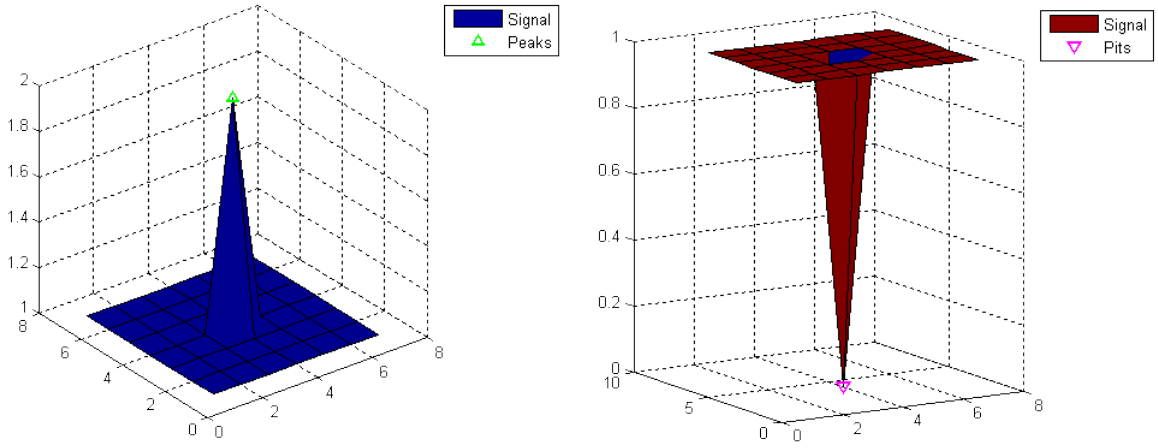


FIGURE 2.25 – Cas simple en 2D de points critiques de type **pic** (à gauche) et de type **creux** (à droite)

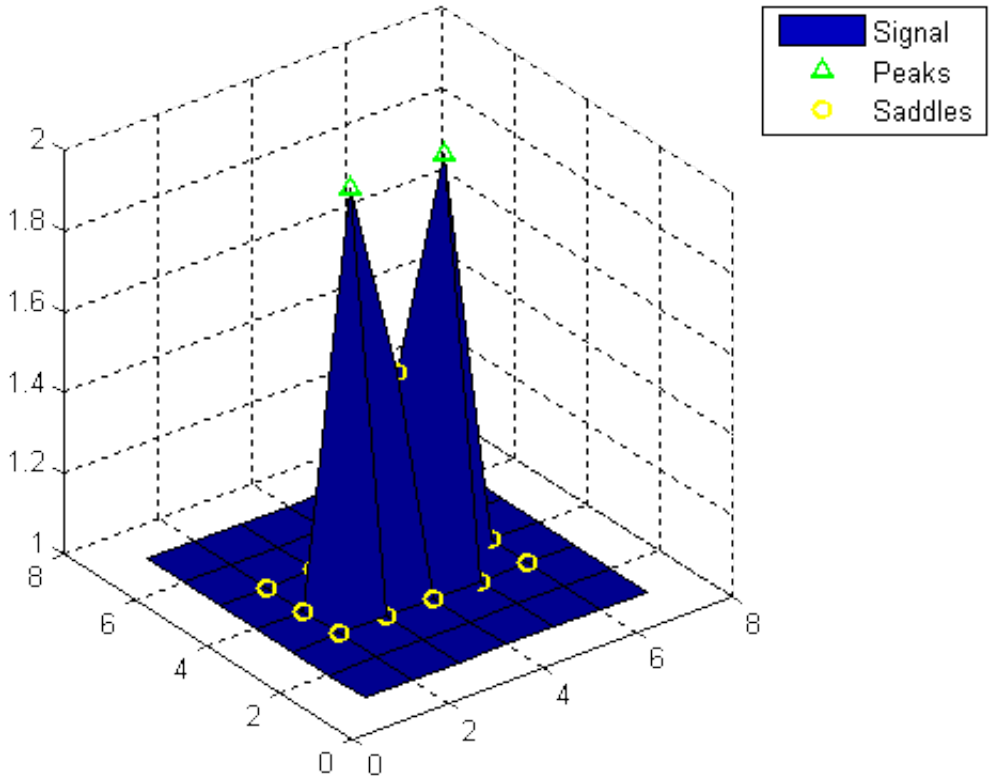


FIGURE 2.26 – Cas simple en 2D d'un point critique de type **selle** ou point col

Chapitre 3

Résultats et discussion

3.1 Introduction

La description complète de l'application de la quantification vectorielle algébrique et arborescente (QVAA) à la simplification de donnée moléculaire a été faite au chapitre 2. Il s'agit à présent de tester expérimentalement la démarche en l'intégrant au sein d'une structure logicielle globale pour faciliter la réutilisabilité basée sur d'autres configurations de quantification. Dans ce but, il est notamment intéressant de conserver une implémentation générique concernant le choix du réseau régulier de points, le facteur d'échelle et la gestion de l'arborescence.

Pour tester notre démarche, nous avons utilisé les données source issues de fichier **cube** des trois anti-coagulants (des protéines simples) assez similaires bien que présentant des différences simples et remarquables [Leh01]. Au cours du projet, nous avons ajouté des fichiers **cube** de la molécule d'eau car il est apparu que dans un contexte de recherche en quantification, une molécule de trois atomes assez simple pouvait faciliter l'interprétation des résultats. Les fichiers sont fournis par Oana Cramariuc, notre *cliente*.

3.2 Description des données d'entrée

Les tests de pertinence et d'efficacité de notre méthode sont effectués à partir de trois protéines simples, les anticoagulants MQPA, NAPAP et 4-TAPAP décrits au chapitre 1.1.13 (page 28). En ce qui concerne la détection et l'analyse de points critiques, les tests ont été effectués avec la molécule d'eau car les molécules précédentes sont encore trop complexes pour nous permettre d'interpréter aisément les résultats.

3.2.1 Caractéristiques à mettre évidence sur les molécules

On rappelle que ces trois molécules adoptent toutes une structure en étoile à trois branches terminées par une fonction *Sulfonyle*, un anneau *Pipéridine* ou un groupe *Amidine* (figure 1.17, page 29).
Les figures 3.1, 3.2 et 3.3 sont visualisées grâce à un code couleur, du bleu (faible) au rouge (fort), et un coefficient de transparence en fonction du niveau de densité électronique de chaque voxel. Lorsque on analyse ces figures, il est important de garder à l'esprit que ces EDM obtenues à travers la quantification vectorielle algébrique et arborescente (QVAA) ont un niveau de détail proportionnel au nombre de quantification successives effectuées. Ainsi, sur nos exemples, après cinq quantifications successives, le résultat est très proche de nos données source. Dans chaque cas, l'atome d'oxygène de groupes *Sulfonyle*, *Carbonyle* ou *Carboxyle* induit des pics dans les EDM obtenues après cinq étapes de quantification successives. Ces groupes doivent facilement correspondre lorsqu'on les superpose.

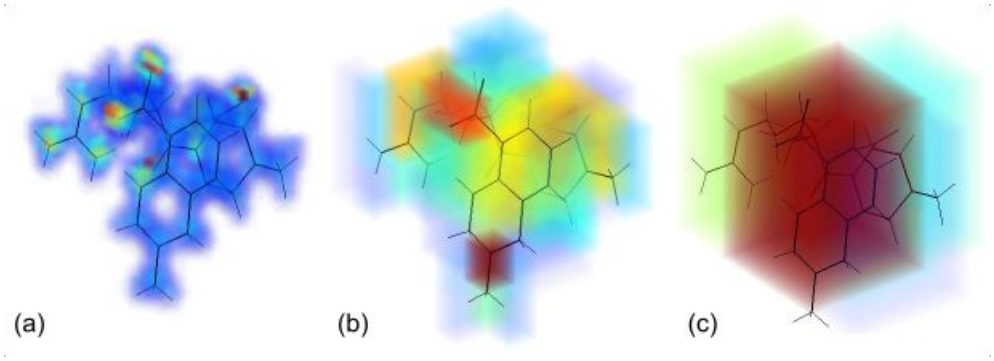


FIGURE 3.1 – EDM de la molécule MQPA (résolution $113 \times 96 \times 93$) avec les niveaux 5, 3 et 2 de QVAA (Réseau $Z3$, Facteur d'emboîtement $b2$).

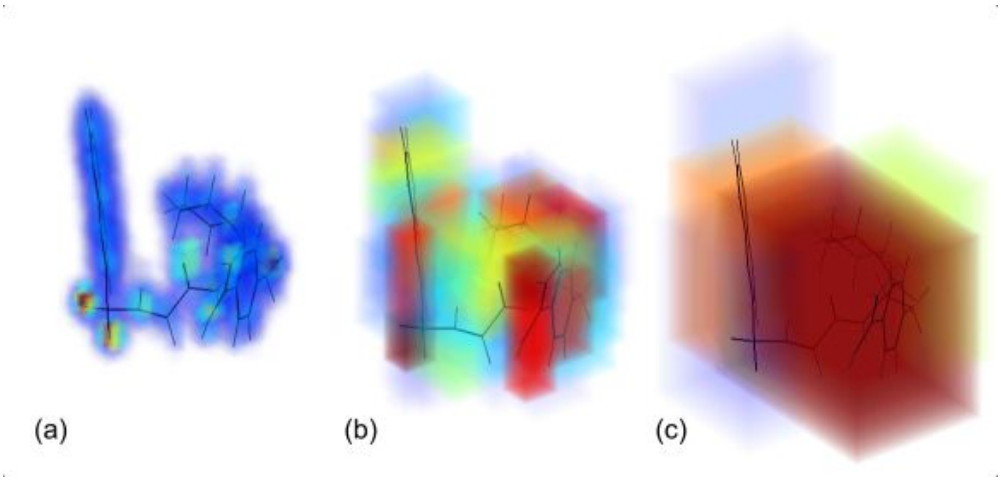


FIGURE 3.2 – EDM de la molécule NAPAP (résolution $120 \times 98 \times 85$) avec les niveaux 5, 3 et 2 de QVAA (Réseau $Z3$, Facteur d'emboîtement $b2$).

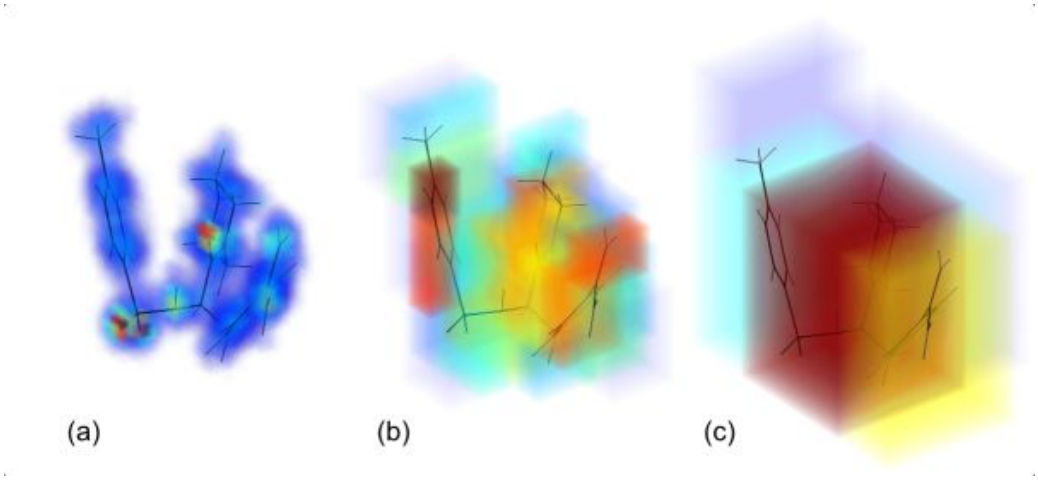


FIGURE 3.3 – EDM de la molécule 4-TAPAP (résolution $119 \times 94 \times 89$) avec les niveaux 5, 3 et 2 de QVAA (Réseau $Z3$, Facteur d'emboîtement $b2$).

3.3 L'analyse des résultats

3.3.1 Introduction

Les traitements de QVAA malgré la simplification apportée par la quantification vectorielle algébrique (QVA) restent assez long, de l'ordre de six à sept heures pour chacun de nos anticoagulants à des résolutions prochent de $120 \times 120 \times 120$. L'issue des traitements produit des fichiers **.cube** (un par niveau de quantification) et un fichier XML qui contient la totalité de l'arbre de donnée. Chaque fichier **.cube** se visualise avec l'application *Jmol* et, dans notre cas, le fichier XML avec un éditeur de texte avancé (*Notepad++*).
À partir des différents fichiers **.cube** quantifiés, nous pouvons alors procéder à la détection de points critiques. Cela offre la capacité de caractériser les nouveaux volumes engendrés.

3.3.2 Analyse visuelle - Jmol

Les figures 3.4, 3.5, 3.6 ont été générées à l'aide d'un script *Jmol*, qui charge toutes les molécules dans le même position, sous le même angle de vue et au même facteur de zoom. Ces visualisations nous montre l'effet de la quantification sur les données des fichiers **.cube**. À l'origine, les données sont intactes et complètes, on distingue les isocourbes dans les moindres de leurs détails. Après trois étapes de quantifications successives, les isocourbes se décomposent en ensemble de polyèdre cubique (lié au choix du réseau régulier de points $Z3$ de la QVA) car les valeurs de densité électronique des représentants ont remplacées celles des données source. De part l'alignement selon la structure du fichier **.cube** source, respectée pour des raisons de compatibilité, la forme de la cellule de *Voronoï*, dual du réseau régulier de points, transparaît, ici un cube. Cette effet ce précise d'autant plus qu'on ajoute des étapes de quantification (étapes 5 de la figure). Sachant que, pour rappel, lorsqu'on effectue trop d'étapes, la quantification n'a plus d'effet et donc, transparaissent les données source.
On distingue bien sur ces figures la *topologie* des molécules malgré la quantification ce qui, *visuellement*, montre qu'on ne perd pas l'information qui nous intéresse, c'est à dire **la forme de la molécule**.

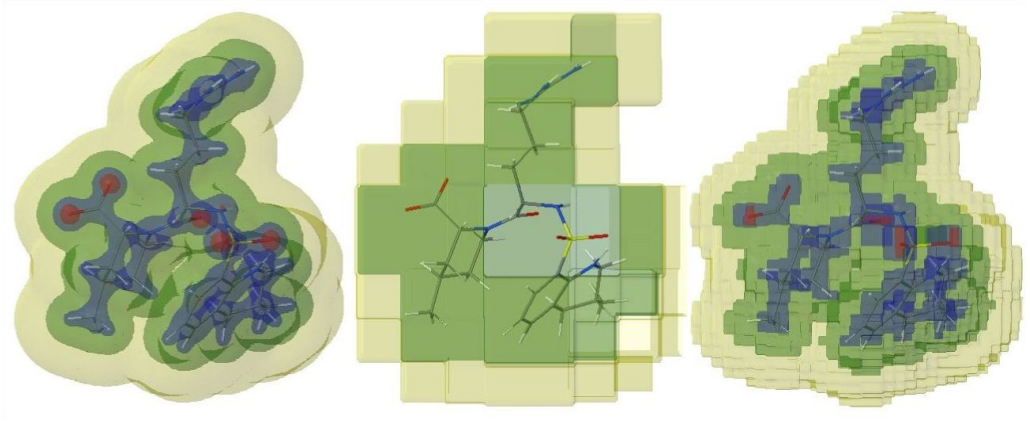


FIGURE 3.4 – Modèle en isosurface de la molécule MQPA (résolution $113 \times 96 \times 93$) d'abord sans quantification puis avec les niveaux 3 et 5 de QVAA (Réseau $Z3$, Facteur d'emboîtement $b2$).

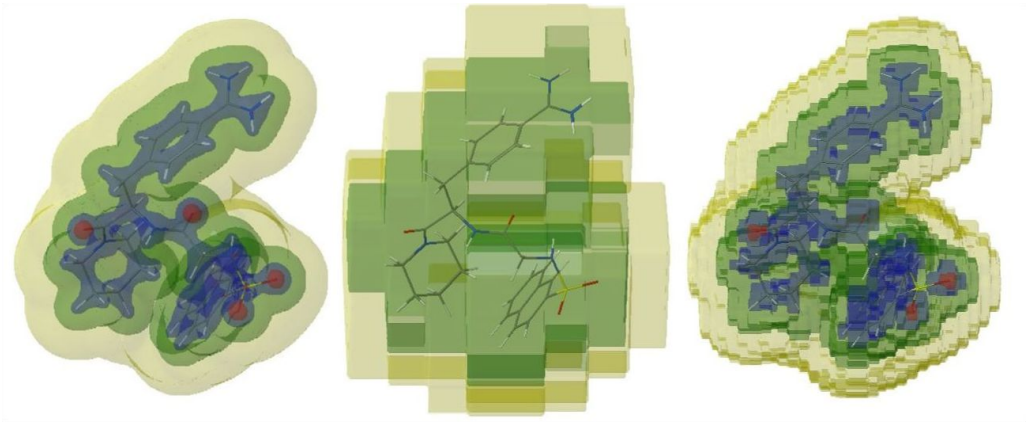


FIGURE 3.5 – Modèle en isosurface de la molécule NAPAP (résolution $120 \times 98 \times 85$) d'abord sans quantification puis avec les niveaux 3 et 5 de QVAA (Réseau $Z3$, Facteur d'emboîtement $b2$).

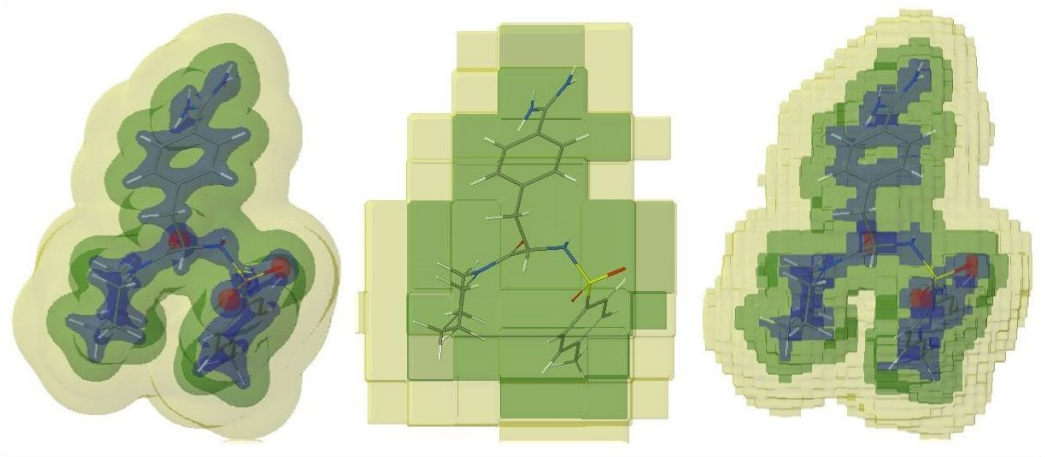


FIGURE 3.6 – Modèle en isosurface de la molécule 4-TAPAP (résolution $119 \times 94 \times 89$) d'abord sans quantification puis avec les niveaux 3 et 5 de QVAA (Réseau $Z3$, Facteur d'emboîtement $b2$).

3.3.3 Analyse informatique - points critiques

Lorsqu'on dispose de fichier **.cube** quantifié, il est nécessaire de disposer d'un outil qui pourra caractériser de façon plus mathématique (que la visualisation *Jmol*) la forme des molécules. La détection de points critiques assurent cette tâche. Pour simplifier l'interprétation des résultats, ce calcul a été effectué sur la **molécule d'eau** (seulement trois atomes) à une résolution néanmoins significative ($120 \times 120 \times 120$). Pour la même raison, les résultats sont proposés en $2D$ sur une *tranche* d'EDM selon le plan décrit figure 3.7. La figure 3.8 présente le résultat de la détection de points critiques sur l'EDM source (donc non quantifié) de ce plan. Attention, en $2D$ il y a bien, 3 dimensions (x, y , densité) !

On remarque trois pics évidents qui sont : le centre de l'atome d'oxygène et les centres des atomes d'hydrogènes. Mais d'autres pics sont détectés et

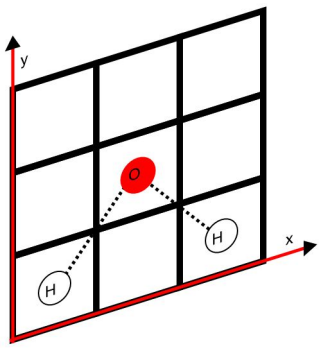


FIGURE 3.7 – Extraction d'une tranche de l'EDM de l'eau (H_2O). On s'intéresse à toutes les valeurs de densité électronique contenues dans le plan (x,y) qui passe par les centres des trois atomes.

ce sont typiquement ces *détails* topologiques qui intéresse la modélisation moléculaire. Ils sont très caractéristiques, ce qui constitue une avancée vers l'objectif visé de déterminer une *signature*.

Après trois étapes de quantification (figure 3.9), les données sont très dégradées, elles sont alignées sur les faces de la cellule de *Voronoï* inhérente au réseau régulier de points utilisé. Des pics se distinguent dans les formes, une localisation générale des maximas se devine mais de prime abord, l'information est plutôt noyée par un effet de quantification nettement trop fort.

Après quatre étapes de quantification (figure 3.10), bien que le signal soit toujours assez dégradé, une comparaison avec la figure 3.8 permet de remarquer la forme globale de la molécule. L'effet de lissage de la quantification fait nettement ressortir les centres des atomes où les pics de densité électronique sont les plus importants.

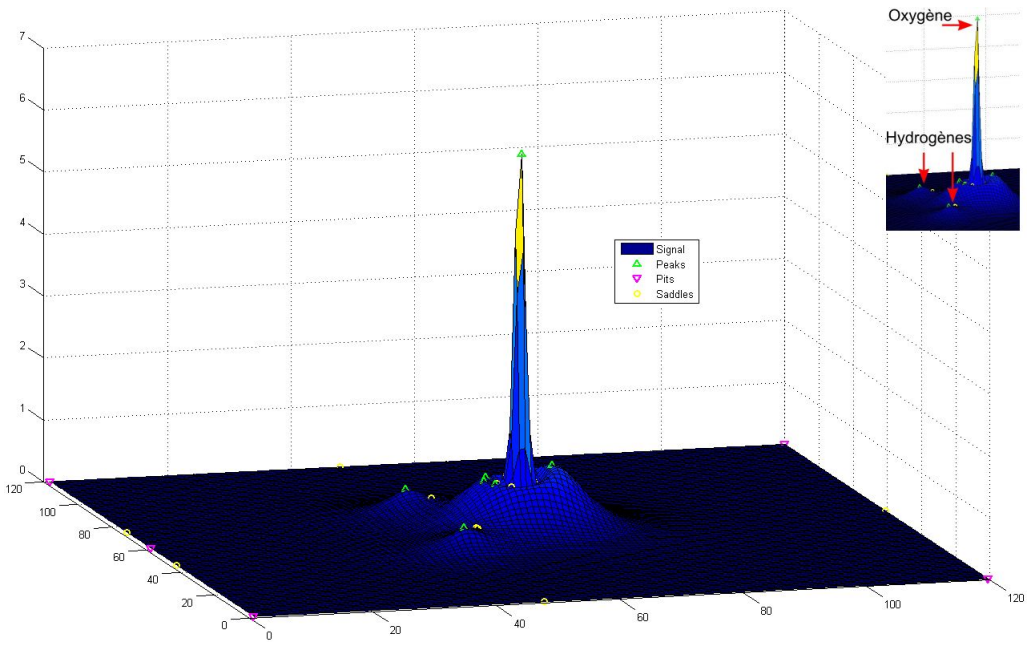


FIGURE 3.8 – Calcul de points critiques pour la molécule d'eau (résolution $120 \times 120 \times 120$) **sans quantification**.

Le principe de l'utilisation de la quantification vectorielle algébrique et arborescente (QVAA) appliquée à la modélisation moléculaire (MM) montre des qualités indéniables de caractérisation de la topologie électronique. L'idée de *signature* prend tout son sens à l'étude des résultats obtenus.

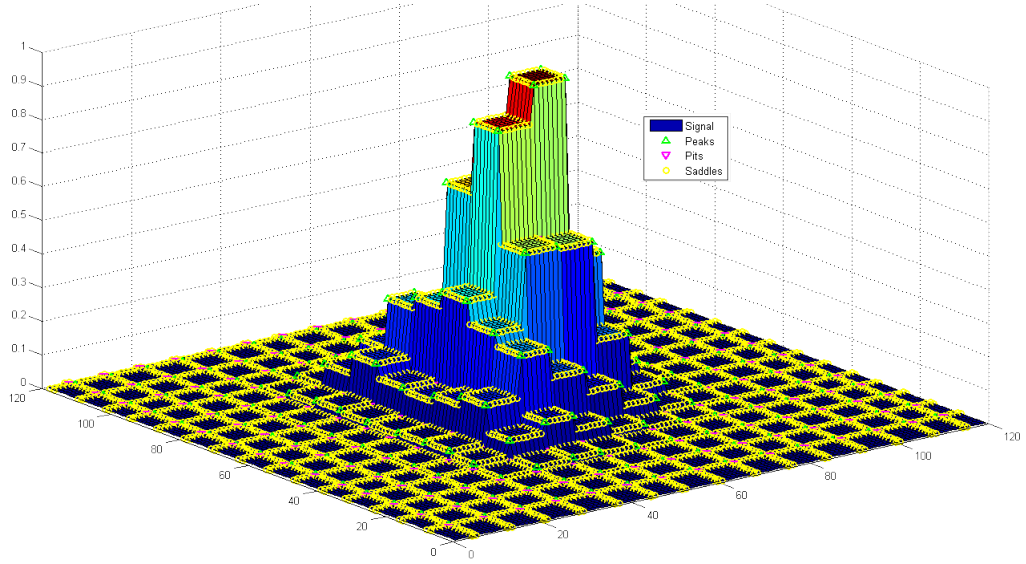


FIGURE 3.9 – Calcul de points critiques pour la molécule d'eau (résolution $120 \times 120 \times 120$) après **trois** niveaux de QVAA (Réseau $Z3$, Facteur d'emboîtement $b3$).

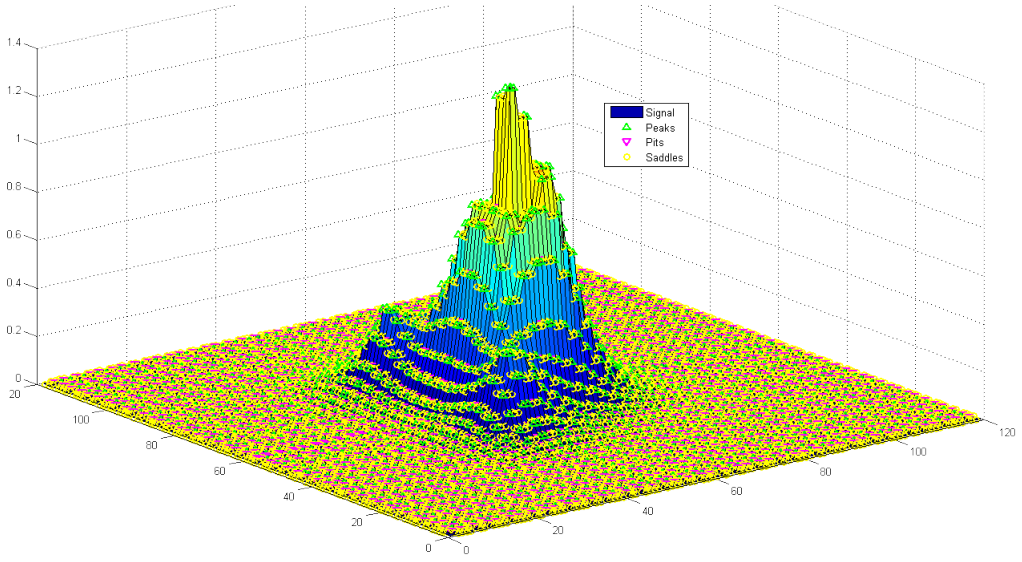


FIGURE 3.10 – Calcul de points critiques pour la molécule d'eau (résolution $120 \times 120 \times 120$) après **quatre** niveaux de QVAA (Réseau $Z3$, Facteur d'emboîtement $b3$).

Chapitre 4

Gestion de projet

4.1 Méthode de gestion de projet

4.1.1 Introduction

Les méthodes de gestion de projet traditionnelles prônent une succession des différentes activités qui suivent une planning établi, depuis les spécifications qui répondent au cahier des charges jusqu'à la validation du système [B01]. Elles visent à anticiper au mieux la façon dont les choses *devraient* se dérouler. Malheureusement, cette approche n'est pas toujours réaliste quand on considère les projets. Les tâches de l'ingénierie ne sont pas un enchaînement successif dépourvu du moindre bouleversement de planning souvent éphémère. La conséquence est que plus de 80% des projets exécutés selon ces méthodologies connaissent des retards, des dépassements budgétaires, quand ils ne finissent pas en échec total, pour n'avoir pas su satisfaire les attentes des clients [Bec99] [TN86].
Ces problèmes sont liés à plusieurs caractéristiques fondamentales de ces anciennes méthodologies :

Le rôle du client : le client n'est présent qu'au lancement du projet, à quelques jalons majeurs plus ou moins espacés et surtout à la fin de projet pour la réception et la recette du système réalisé. Cet *effet tunnel* conduit à une solution souvent inadaptée et dont la qualité est inacceptable.

Le contrat au forfait qui durcit les relations entre client rend le passage de témoin long et douloureux à la fin du projet.

La standardisation des activités d'ingénierie, dont l'enchaînement se révèle souvent inefficace. Dans les faits, dans les premières étapes, les contrôles d'avancement et de qualité ne peuvent être menés qu'à partir de documents. Or bien des organisations sont devenues des usines à produire de la documentation au lieu de produire des fonctions logicielles pertinentes pour les clients et les utilisateurs.

Le passage de relais entre les phases successives dans lesquelles oeuvrent des équipes différentes, généralise une relation de type client-fournisseur et n'encourage ni l'empathie ni l'esprit d'équipe, bien au contraire. Chaque transition se traduit par une perte de temps, de savoir, d'informations ou de responsabilité.

Assez curieusement, même si le prise de conscience fut progressive, aucune de ces difficultés inhérentes aux anciennes méthodologies de gestion de projet ne sont intervenues pendant mon mémoire. Le client était représenté par mes encadrants, qui se sont montrés *force de proposition* constructive et omniprésente. Le contrat se résumait à la rédaction du mémoire et des articles. Le monde de la recherche s'abstrait, par définition, de toute standardisation et le passage de relais n'interviendra, hypothétiquement, que lors de l'industialisation.

Néanmoins, de part la nouveauté de l'activité pour moi et pour organiser la

période de façon souple, le développement de ce projet s'est inscrit dans un processus de gestion de projet basé sur les **méthodes agiles**.

4.1.2 Méthodes agiles

Les méthodes Agiles consistent en un ensemble de pratiques conçues pour éviter les difficultés rencontrées lors de projet aux cycles standards de développement en **cascade** ou en **V** (figure 4.1).

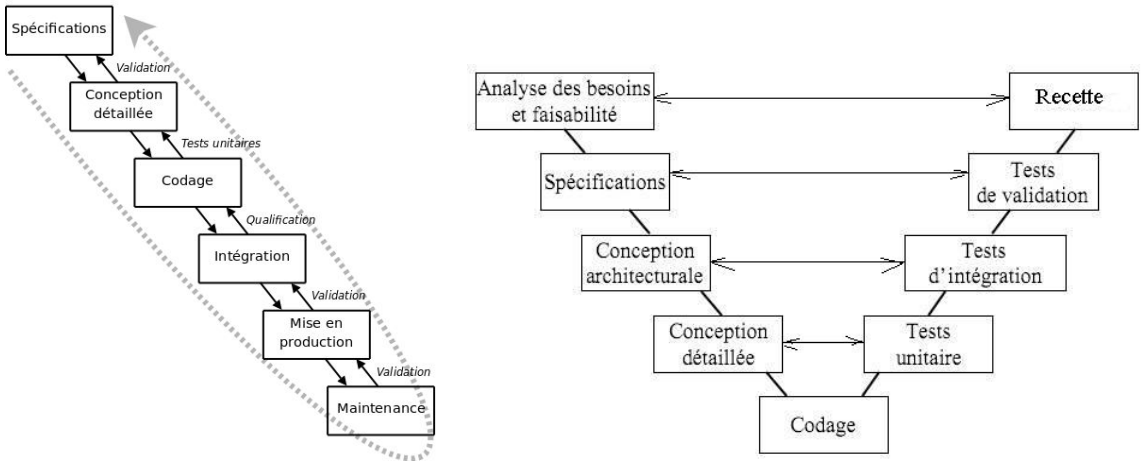


FIGURE 4.1 – Cycles standards de développement en **cascade** (à gauche) et en **V** (à droite) (source Wikipédia).

Ces pratiques se schématisent figure 4.2 et s'énoncent ainsi :

L'adoption d'un cycle itératif et incrémental permettant à une équipe de s'adapter au contexte ainsi qu'aux changements qui ne manquent pas de survenir au cours d'un projet. Dans notre cas, ce cycle se composait de couple discussion/développement.

L'implication du client dans le développement, permettant au client et à l'utilisateur de donner leur feedback quant au devenir de l'application

en cours de développement, annulant ainsi tout *effet tunnel*. L'implication de mes encradrants a permis l'omniprésence du *client*.

La définition d'objectifs à court terme qui permet de maintenir une pression constante mais supportable sur l'équipe, alors qu'au début d'un cycle en V chacun a l'impression d'avoir suffisamment de temps devant lui et subit finalement une pression énorme à l'approche de la livraison. Pour ce mémoire, Le principe même de l'activité de *recherche* impliquait des petits objectifs sous la forme d'idée à valider.

La collaboration entre les personnes et les équipes qui combat les passages de relais en rassemblant dans un même espace toutes les énergies et la compétence de personnes centrées sur l'application à réaliser. L'équipe définit des tâches ponctuellement, *quand c'est le moment*, plutôt qu'au début du projet.

La livraison d'un produit opérationnel de bonne qualité parce que souvent testé, doté de la seule documentation strictement nécessaire, et répondant à coup sûr aux vrais besoins des utilisateurs puisqu'il est régulièrement soumis à leur feedback. Nos résultats sont basés sur une application simple mais fonctionnelle.

En génie logiciel, les méthodes agiles se basent sur la méthode ***SCRUM*** qui elle-même est implémentée à l'aide de l'***eXtreme Programming (XP)***.

La méthode ***SCRUM*** [TN86] est un processus de développement de projet appliqué au génie logiciel qui s'intéresse plutôt à l'organisation du projet qu'aux aspects techniques. Son objectif étant d'améliorer la productivité des équipes auparavant ralenties par des méthodologies plus lourdes, cette méthode pourrait être appliquée à d'autres domaines. Elle est très flexible grâce à l'approche incrémentale et basée sur la priorisation des besoins du client.

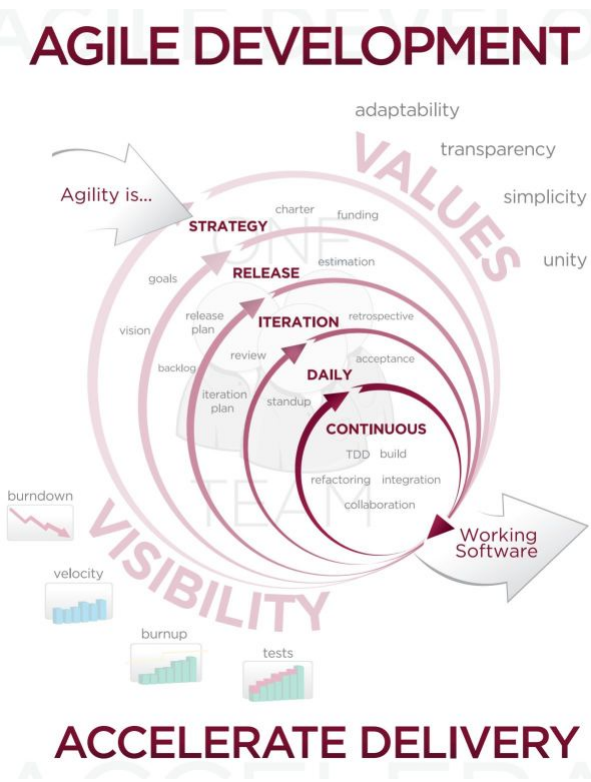


FIGURE 4.2 – Méthodologie AGILE de développement logiciel (source Wikipedia).

Les pratiques de ***SCRUM*** sont avant tout orientées vers la maîtrise de livraison d'incréments (ou *sprint*) d'une façon assez stricte au point de ne pouvoir modifier les fonctionnalités au cours de l'implémentation. Cette limite interdit la mise en oeuvre d'une conception itérative comme celle d'XP basé sur de possibles raffinements par modification permanente. Par ailleurs ***SCRUM***, ne dispose pas de métrique de gestion du changement à ce niveau, elle nécessite donc une importante spécification préalable à la mise en production (ou *backlog produit*) et, du fait de cette prédictibilité imposée, cette méthode ne peut pas être considérée comme réellement itérative.

Tout en s'appuyant sur des *bonnes* pratiques de programmation [Bec99], ***eXtreme Programming (XP)*** propose un développement par itérations courtes et gérées collectivement. Le client est impliqué à tous les niveaux du projet. Cette relation au client implique qu'il soit très disponible.
eXtreme Programming (XP) repose sur cinq valeurs fondamentales :

La communication C'est le nerf de la **guerre** qui permettra d'anticiper tous les problèmes qui ne manqueront pas de survenir. XP imposent une communication de tous les instants. Les tests conçus en amont, la programmation (de préférence à deux) et les évolutions de planning obligent les développeurs et les clients à communiquer. Lorsqu'un problème apparait néanmoins, un chef de projet doit l'identifier et ensuite provoquer une rencontre entre les personnes concernées.

La simplicité que l'on peut résumer comme suit : *« Si c'est compliqué, c'est faux ! »*. Chercher à anticiper les futures extensions est une perte de temps car de toute façon une application simple est forcément plus facile à faire évoluer.

Le *feedback* Le retour d'information est primordial pour le programmeur et le client. Les tests unitaires indiquent si le code fonctionne. Les tests fonctionnels donnent l'avancement du projet. Les livraisons fréquentes permettent de tester les fonctionnalités rapidement.

Le courage Certains changements demandent beaucoup de courage. Il faut parfois changer l'architecture d'un projet, jeter du code pour en produire un meilleur ou essayer une nouvelle technique. Le courage permet de sortir d'une situation inadaptée. C'est difficile, mais la simplicité, le *feedback* et la communication rendent ces tâches accessibles.

Le respect Cette valeur bien que rajoutée à la deuxième édition de [Bec99] est une des plus emblématique de l'XP car d'elle découle la qualité de

la communication.

Dans le cadre de ce mémoire, la forte implication des encadrants et la nature de l'activité de recherche ont permis une relation proche au *client* d'où l'utilisation d'un cycle de développement basé sur l'XP. En particulier, une longue phase d'exploration de la littérature ponctués de discussion régulière puis des cycles de développement hebdomadaires jalonnés de réunions de validations informelles.

4.2 Diagrammes de Gantt

4.2.1 Introduction

Le diagramme de Gantt est un outil utilisé en gestion de projet qui schématise dans le temps toutes les tâches d'un projet. Il s'agit d'une représentation d'un graphe connexe, valué et orienté qui permet de représenter graphiquement l'avancement du projet.
Cet outil a deux objectifs : optimiser la planification et proposer un support de communication sur le planning. A l'aide de ce diagramme on peut facilement :

- déterminer les dates qui jalonnent le projet,
- identifier les marges d'évolution du planning,
- visualiser le retard ou l'avancement des tâches.

Le diagramme de Gantt ne permet pas de solutionner le problème de concurrence de ressources. On peut, toutefois, s'en affranchir avec des règles de priorité. Il s'agit d'ordonner selon la priorité les tâches à effectuer dans le diagramme de Gantt en tenant compte de la disponibilité des ressources.

Lors de la plannification du projet, un diagramme de GANTT prévisionnel a été construit de façon à parallèliser les phases de conception/réalisation et la rédaction du mémoire.

4.2.2 GANTT prévisionnel

Le diagramme de GANTT prévisionnel du projet (figure 4.3)

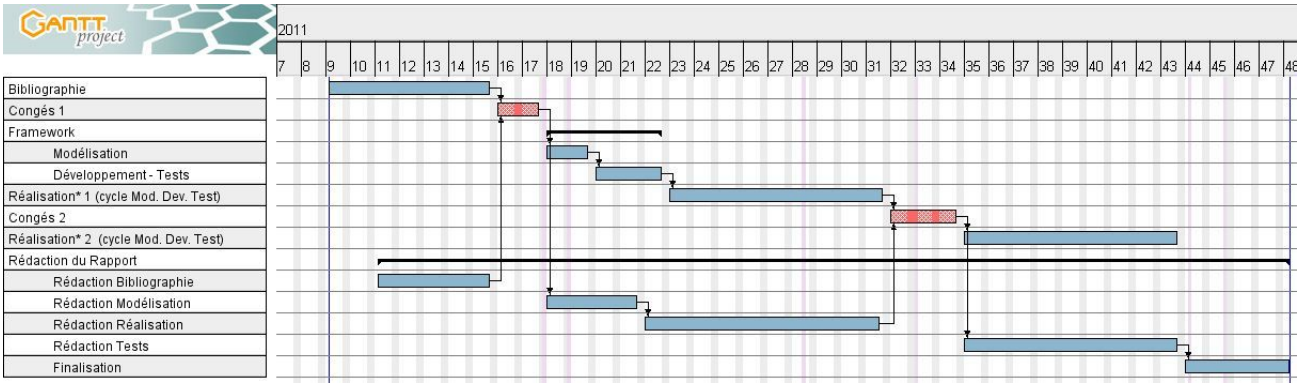


FIGURE 4.3 – Diagramme de GANTT prévisionnel.

4.2.3 GANTT effectif

Le diagramme de GANTT effectif du projet (figure 4.4). On remarque tout d'abord que la rédaction du mémoire a commencé près d'un mois après la date prévue car la phase de bibliographie s'est montrée particulièrement ardue de part la somme conséquente de nouveaux concepts issus autant du traitement du signal (notamment le formalisme mathématique) que de la modélisation moléculaire. Par ailleurs, la rédaction de l'article pour la conférence SPAMEC 2011 n'a pu être anticipée et a impliqué une ventilation d'environ un mois des tâches prévues. Ce retard a pu être compensé sur la deuxième phase de développement qui s'est révelée plus courte que prévu. Ainsi dans l'ensemble, à part donc la rédaction, le planning a été respecté.

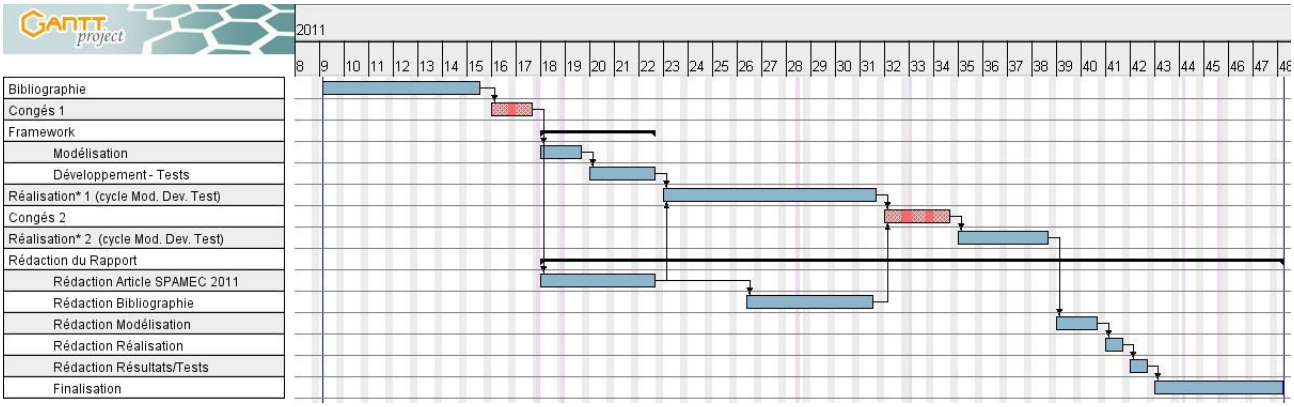


FIGURE 4.4 – Diagramme de GANTT effectif.

4.3 Expérience personnelle

4.3.1 Introduction

J'ai rédigé ce mémoire au sein de l'équipe **Images et VidéoCommunications (IVC)** du laboratoire de l'*Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN)* **(c.f. Annexes présentations du labo...)**.

Le projet est encadré par Vincent Ricordel, maître de conférences et directeur du département informatique de l'école d'ingénieur de l'université de Nantes, *Polytech'Nantes*. Deux chercheurs externes à l'équipe IVC ont été aussi moteurs du projet, Oana Cramariuc du *Département de physique de l'université de technologie de Tampere (TUT)* en Finlande et Bogdan Cramariuc du *Centre informatique pour la Science et la Technologie (CIST)* de Bucarest en Roumanie.

4.3.2 Intégration à l'équipe IVC

Pendant les neuf mois du projet, j'ai été intégré à l'IVC comme membre de l'équipe au même titre qu'un doctorant. Cela m'a permis de découvrir et prendre goût au monde de la recherche. Les rencontres internes au laboratoire, parfois internationales, de chercheurs, doctorants, post-docs, ingénieurs, sont autant d'occasion d'élargir les connaissances et l'ouverture d'esprit. En effet, le laboratoire de recherche est un lieu très stimulant intellectuellement où l'*idée* et sa réalisation sont des richesses partagées et partageables, à condition de respecter les codes et les règles souvent non-écrites. En effet, la liberté de *pensée* est assujettie à une rigueur structurante, autant dans la réflexion que dans la communication. Ainsi, j'ai eu l'opportunité de pouvoir publier un article lors d'une conférence (SPAMEC 2011) en Roumanie [RRCC11] (en Annexe, page 114) et l'expérience que je retire de cet exercice est aussi formatrice que gratifiante.

J'ai eu la chance d'être fortement impliqué dans cette vie de l'équipe qui est rythmée par les nombreuses arrivées, départs et séminaires de chercheurs du monde entier. J'ai participé à la promotion du laboratoire lors des *portes ouvertes* et géré des formalités administratives inhérente à la vie de l'équipe. Des amitiées se sont créés qui dépassent déjà largement le cadre de cette période de mémoire. Et au-delà de cette expérience humaine, j'ai découvert un métier : *chercheur*, qui, assez abstrait vu du monde de l'entreprise d'où je viens, est une révélation qui a suscité le renouveau d'une passion pour mon activité professionnelle et peut-être une opportunité de changement de carrière.

Conclusion

Cette étude d'une méthode de quantification vectorielle algébrique et arborescente (QVAA) appliquée à la modélisation moléculaire (MM) s'inscrit parmi les travaux de recherche engagés pour le développement de nouveaux outils de synthèse de molécules. Les enjeux suscités, notamment par le volume et la complexité calculatoire, sont en attente d'instruments de simplification et d'analyse.

L'utilisation de techniques de traitement du signal est une idée révolutionnaire et récente dans le monde de MM et par extension de la chimie. Cependant, notre but n'a consisté qu'à s'intéresser à l'angle analytique de la quantification. En général, cette démarche s'inscrit dans une finalité de compression de donnée en limitant les pertes d'information. Mais la compression n'est pas notre but, nous voulons obtenir une description exhaustive des volumes unitaires composant la molécule, afin d'en déterminer son architecture tridimensionnelle.

La technique de QVAA, de part sa nature algébrique basée sur des réseaux réguliers de points, permet de réduire de façon drastique la complexité calculatoire ; De plus, les données sont organisées en niveaux de résolution grâce à la structure arborescente des données produites. Cette idée simplifie la description sans perdre l'information primordiale, dans notre cas la disposition tridimensionnelle des atomes d'une molécule. La vision simplifiée de

la molécule est alors caractérisée à l'aide d'un outil de détection de points critiques.

Notre méthode est fonction de plusieurs paramètres (Méthode de quantification, Réseau, facteur d'emboîtement, résolution) et chacune de leur combinaison (ou configuration) mérite une étude approfondie pour déterminer la plus performante.

Le but de ce mémoire était de disposer d'une structure logicielle (ou *framework*) où il serait possible d'implémenter ces configurations afin de les étudier en détail. L'objectif est rempli mais de nombreuses tâches restent à achever car les résultats ne montrent qu'une idée de la *signature* représentant la molécule et ses atomes. Il faut maintenant démontrer la détection des groupes fonctionnels (des groupes d'atomes) qui composent la molécule. Les liaisons des atomes avec certains autres d'entre-eux doivent être qualifiées. [Leh01] suggère le calcul de diagrammes de connectivité, et nous pensons que cette idée est pertinente pour notre modèle.

Annexe A

L'IRCCyN - Équipe IVC

L'Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN) est une unité mixte de recherche du Centre National de la Recherche Scientifique (UMR CNRS 6597), rattachée principalement à l'Institut des Sciences de l'Ingénierie et des Systèmes (INSIS), et secondairement à l'Institut des Sciences Informatiques et de leurs Interactions (INS2I) ainsi qu'à l'Institut des Sciences Biologiques (INSB), et dont les tutelles locales sont l'Ecole Centrale de Nantes, l'Université de Nantes et l'Ecole des Mines de Nantes.

La recherche à l'IRCCyN vise à la fois à produire de nouvelles connaissances et à développer des méthodes et des outils destinés à apporter des solutions à des problèmes concrets issus d'acteurs du tissu économique local. Cette démarche permet de déterminer de nouveaux axes de recherche pour les chercheurs de l'Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN).

L'institut est constitué de onze laboratoires appelés **équipes** qui couvrent un large spectre scientifique, qui va de l'automatique des systèmes complexes à la psychologie cognitive en passant par la robotique ou la vidéo communi-

cation et le traitement de l'écriture manuscrite.

L'équipe Images et VidéoCommunications (IVC) s'intéresse plus particulièrement au traitement et à l'analyse de la plupart des types de signaux multimédias : l'acquisition, la transmission, le codage, le stockage et la visualisation. Á ces domaines s'ajoute le traitement de l'écriture manuscrite. Plusieurs secteurs de recherche sont explorés autour de trois pôles principaux qui sont : la perception, la communication et la représentaton.
Les projets en cours à l'IVC sont :

HWR2 : Ressources pour la reconnaissance de l'écriture manuscrite ;

Image and video quality assessment : Environnements normalisés de test, ressources matérielles et logicielles, bases de données pour la mesure le la qualité visuelle des images et des vidéos ;

3D TV and human factors : Ressources pour l'étude des facteurs humains en visualisation en relief ;

Visual attention : Plateforme matérielle et base de données pour l'étude de l'attention visuelle ;

Network simulation and emulation : Emulateur matériels et simulateurs de réseaux grande échelle hétérogènes ;

Watermarking and Security : Atelier logiciel et bases de données ;

Discrete geometry and Mojette : Application de la géométrie discrète au réseau et à l'imagerie médicale.

PERSEE : *PERceptual Scheme for 2D ans 3D vidE(E)0 coding.* Schémas perceptuels et codage vidéo 2D et 3D.

Annexe B

Article SPAMEC 2011

L'article suivant a été soumis et accepté à la conférence **Signal Processing and Applied Mathematics for Electronics and Communications (SPAMEC** 2011) qui s'est déroulée du 26 au 28 août 2011 dans la ville de Cluj-Napoca en Roumanie.
Ce document a été défendu sur place par Bogdan Cramariuc.

LATTICE VECTOR QUANTIZATION FOR THE ANALYSIS OF MOLECULAR DATA

Cédric RAMASSAMY[1,*], Vincent RICORDEL[1], Oana CRAMARIUC[2], Bogdan CRAMARIUC[3]

[1]LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597 (Institut de Recherche en Communications et Cybernétique de Nantes), Polytech Nantes, rue Christian Pauc BP 50609 44306 Nantes Cedex 3; [2]Department of Physics, Tampere University of Technology, P.O. Box 692, FI-33101 Tampere, Finland; [3]IT Center for Science and Technology, Av. Radu Beller 25, Bucharest, Romania.

cedric.ramassamy@univ-nantes.fr, vincent.ricordel@univ-nantes.fr, oana.cramariuc@tut.fi, bogdan.cramariuc@citst.ro

ABSTRACT

We introduce a novel simplification scheme of molecular data using Tree-Structured Lattice Vector Quantization (TSLVQ). The method, based on the embedding of truncated lattices, permits also hierarchical description of the 3-D volume through a tree-structure. We apply TSLVQ to simplify the electron densities maps of three thrombin inhibitors (MQPA, NAPAP, 4-TAPAP).

Keywords: *computational chemistry, electron density, simplification, tree-structure, lattice, vector quantization, multi-resolution, molecular similarity, MQPA, NAPAP, 4-TAPAP.*

1. INTRODUCTION

During the last decades signal processing methods have been employed beyond their traditional application domains and have started to contribute significantly to advancements in biology, biochemistry and biomedicine. What was earlier viewed as digital signal processing represents nowadays only a small part of the new concept of signal processing which can be described as the collection of methods for analyzing, manipulating and presenting natural information. One underlying cause of this expansion is the exponentially growing volume of numerical data obtained through modeling and simulation techniques or by employing modern high-throughput experimental investigation tools such as DNA-, protein-, cellular- and antibody-microarrays. One other cause is the substantial effort employed in developing systemic models of processes taking place in living organisms, models which combine individual processes into a larger coherent picture.

The computing power has now increased to a level high enough to process large amounts of molecular data. Nowadays large systems of 10.000 – 50.000 atoms are approached computationally. Beyond the mere resources limits of processing such a huge amount of data, the major task is to detect and extract semantic knowledge in the molecular system. Thus, more than data compression, a simplification of the molecular topology can help in this task. Additionally, Critical points (CP) and singularities, computed from the simplified molecular data, become representatives of groups of atoms rather than atoms themselves [1].

With specific molecular properties conferred by functional groups (e.g. hydroxyl –OH and carboxyl -COOH) efficient ways to identify these groups are sought [1]. This process can lead to simplified representations of the molecules, for e.g. amino acid residues in proteins can be depicted using a lower level representation, *i.e.* two or three pseudo-atoms representing the whole residue. Such representations are of importance for drug design applications and to overcome structural inaccuracies issued from experimental data such as X-ray analysis. In protein studies, for instance, researches focus on the detailed characterization of macromolecular surfaces and topologies [2]. Synthesis of drug-like molecules is driven by the 3D structural characteristics of molecules [3].

Previous works on simplification focus on a crystallography-based formalism, whereas a recent trend aims at adapting methods taken from signal processing domain. Exactly, the electron densities are decomposed into successive resolution of wavelets. CP analysis on low wavelet resolution showed relevant results [1]. In addition, from the image processing domain, Vector Quantization (VQ) [4] was adapted to electron density simplification, in particular a neural network method based on a cost function [2]. The technique of VQ represents the data into a set of reproduction vectors or codewords while keeping the sensible shape of the source density map.

In the present work we employ Vector Quantization, a method typically used in transmission (source coding) and data classification, as a first step in processing of molecular electron densities for further molecular similarity analysis, topological exploration and visualization. The considered electron densities are obtained from quantum mechanical calculations which, as opposed to previous attempts, allow for a full control of the input data quality and a good understanding of the structure-density relationship. The paper is organized as follows: In the first part the Tree-Structured Lattice VQ (TSLVQ) approach is detailed, the second part deals with the simplification of the Electron Density Maps (EDM) using TSLVQ, results are described in the third part and a conclusion is given at the end.

2. QUANTIZATION

2.1. Vector Quantization

A concise description defines quantization as the computation of an approximation for a given signal [4]. VQ consists in representing, an ordered set of numbers (or vectors) by a more reduced set called the codebook [4,5]. VQ is mainly

applied to source encoding in order to shrink the volume of data needed to represent the information [5]. Our method deals with decreasing the granularity of the source representation in order to characterize hierarchically the information.

2.2. Lattice Vector Quantization

2.2.1. General

The goal of Lattice VQ (LVQ) is to strongly structure the codebook in order to reduce the computing complexity involved in the codebook design [5]. LVQ does not require a training step nor an exhaustive search in order to build the codebook. The codewords are points of a truncated lattice regularly scattered into space [6]. The codewords are also the centres of the *Voronoi* cells, the duals of lattices. The space embedding of the lattices will determine their properties. Unlike in a LBG-based (Linde, Buzo, Gray algorithm) method [4,7], in LVQ the encoding is related to the coordinates of the vector. Thus, it is only based on rounding and scaling operations. Therefore, the coding is very simple [5,6].

2.2.2. Lattices

In [6] it is shown that all lattices have not necessarily an optimal space embedding. Moreover, the efficiency of the quantization is directly linked to the geometry of the chosen polytope which describes the *Voronoï* cell. The choice of the best lattice in the molecular situation leads to an orientation independence feature. The EDMs are three-dimensional matrices, thus we need 3-D lattices.

There are only three lattices for dimension 3 for which fast quantization algorithms are known: **Z3** Cubic lattice, **D3** the Rhombic dodecahedron and **D3*** the Truncated Octahedron. Figure 1 shows a 2-D lattice embedding example with the hexagon.

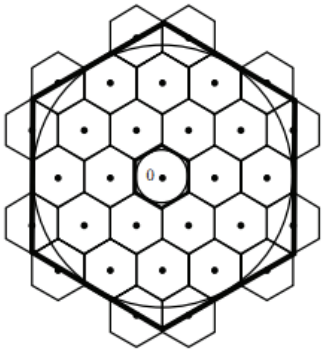


Figure 1: *A sub optimal embedding - the hexagonal Lattice* [8].

2.3. Tree-Structured Vector Quantization

Tree Structured VQ is a gathering of many quantization approaches where the quantization is processed through a decision tree [4]. Its advantages are reduced computation with the use of simpler sub-codebooks, and a structure adapted to progressive representation.

2.4. Tree-Structured Lattice Vector Quantization

Tree Structured LVQ (TSLVQ) [9] aims at using a hierarchical set of embedded lattices which is achieved such as it is possible to embed a lower scale truncated lattice into a cell of the next higher scale truncated lattice. So a scaling factor between consecutive lattices of the hierarchy has to be set (Figure 2). The principles of the TSLVQ are [8]:

1. A source vector is projected into a first truncated lattice;
2. To get a finer quantization, another lower scale truncated lattice is embedded into the *Voronoï* cell where lies the input vector;
3. The previous operation can be repeated.

It is more convenient to deal with the input vector scale than to use several lattices with different scales. The principles of the encoding are shown at Figure 3.

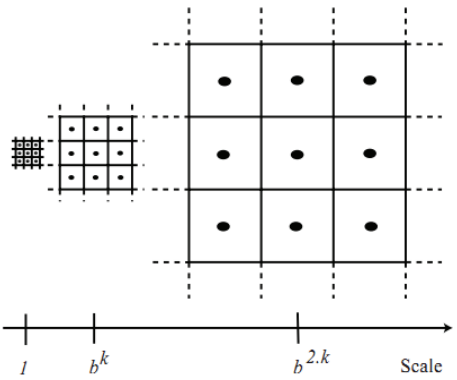


Figure 2: *Hierarchical set corresponding to the cubic lattice. Here, the scaling factor b=3* [8]**.**

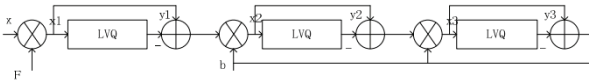


Figure 3: *The principle of TSLVQ* [8]*: x is the source vector, y_n the successive reproduction vector, F the first scaling factor and b the next scaling factors.*

Where we have: the scaling factor F used to project the input vector *x* into the first truncated lattice:

$$F = \frac{b \times \rho}{L_{2\max}}$$

where ρ is the corresponding packing radius, and $L_{2\max}$ the maximal L_2 norm of *x*. So, all the inputs are projected into a hyper-sphere whose radius equals $b \times \rho$.

In the normalized space, the scaling factor used to project each translated vector into the next truncated lattice of the hierarchy is *b*. The reproduction vector of the truncated lattice for the j^{th} level is y_j. The final value of the reproduction vector associated with x will be then:

$$y = \frac{1}{F} \times \sum_j \frac{y_j}{b^{j-1}}$$

with *j* the level of the quantization. At each step, the same fast quantization algorithm is used.

We introduce a novel simplification scheme of molecular data based on TSLVQ. This kind of VQ is very fast thanks to the LVQ and its tree structured codebook. It is, therefore, well adapted to the analysis of the distribution of the source.

3. METHODS

The data source is a *Gaussian cube* file [10], the output data are a set of *cube* files, from the TSLVQ step. Output *cube* files are snapshots of the different levels of quantization (Figure 4).

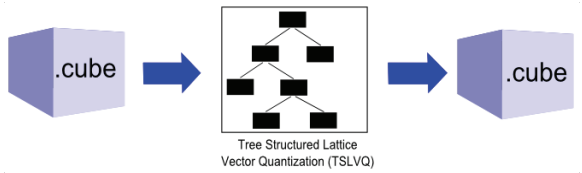


Figure 4: *General scheme of the method.*

3.1. Electron densities

The electron density of a chemical entity plays a critical role in explaining and understanding its properties. In fact, it was proven in 1964 by Hohenberg and Kohn that the electron density determines all properties of the ground state of a chemical system, including the ground state energy. Accordingly, the work of Popelier, based on Bader's Atoms in Molecules theory, proves that it is relevant to exploit 3D electron density data by taking advantage of its topology [11,12]. It has also been shown that the topological analysis of electron density allows simplification of the 3D distribution in reduced representations without losing significant information. In this work, we employ quantum mechanical calculations at the density functional theory level to calculate the EDMs of the compounds described in section 4. The Per-dew-Burke-Ernzerhof exchange-correlation functional to-gether with a TZVP basis set were used as implemented in the Gaussian 09 computational package *www.gaussian.com*.

3.2. Vector Quantization of the electron densities.

3.2.1. Data loading

A Gaussian *cube* file describes the 3-D maps of EDMs or of electrostatic potentials but the present work focuses on EDMs only. The latter stands for a grid that slices the space of the studied molecule. Three data sets of *Gaussian cube* [10] files were used with the new TSLVQ method. We focused our study on three anticoagulants, thrombin inhibitors (MQPA, NAPAP, 4-TAPAP), which were previously used by Leherte as test systems for the wavelet based simplification [1]. Each of the molecules is described in three resolutions of electron densities through their dedicated *cube* file. Given the amount of data, the EDMs need to be batch processed. Flaws of the data slicing should introduce sets of non-trivial errors. Indeed, the size of the batch directly impacts the precision of the quantization.

3.2.2. Quantization with TSLVQ

TSLVQ with *cube* files source needs two parameters: scaling (or boxing) factor and quantization level. There are two ways of constructing the tree: a) "Merging" way consists in building the whole tree in a first step. Then the tree has to be fully browsed in order to merge leaves successively. b) "Splitting" way constructs the tree progressively at each iteration of the quantization. At the first level all points are combined in one single node, this is the most degraded representation of the source, and at the next levels, if the processing rule requires it; a leaf is splitted in a range of children leaves and so on. Using a higher number of quantization levels, we get a more detailed (and voluminous) description of the molecule.

For obvious reasons of resources saving, we have implemented the splitting way. The coordinates of the density are rounded to be aligned on the current lattice reproduction vectors. Once all data are processed a mean is calculated between all the densities of the current codeword. The scaling factor controls the number of possible children for the current node (e.g.: for the cubic lattice, if b=3 in 3-D → 3^3= 27 children possible for each node but only those whose dual *Voronoï* cell contains densities are created). The existing lattice is rescaled for the current child node, and all the densities that belong to the parent node are quantized. The process stops when it reaches the quantization level given in parameter. The quantization tree resulting is the input of the next step.

3.3. Visualization and analysis of the results

The method produces as result a set of *cube* files (one file per level). This allows the visualization and analysis with common computational chemistry tools (e.g.: *Gaussian* [10]).

4. RESULTS

The compounds have a star shape with three branches completed by a cyclic substructure bound to sulfonyl function, piperidine ring, or amidino group (Figure 5).

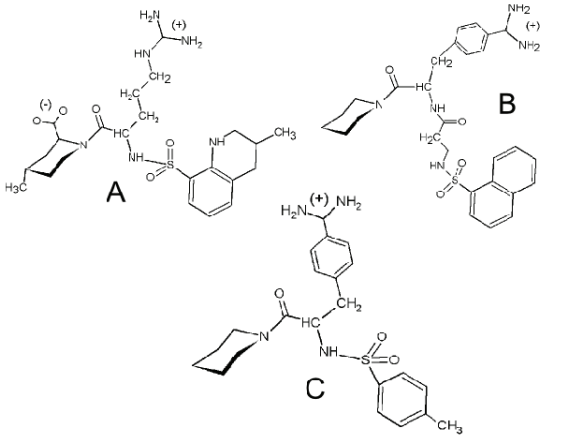


Figure 5: *2-D structures of compounds MQPA (A), NAPAP (B), and 4-TAPAP (C)*

In Figures 6-8 the EDMs are visualized using a blue (low)-to-red (high) colour map and a transparency coefficient proportional to the voxel value. When analyzing Figures 6-8, it is important to keep in mind that the EDMs obtained by TSLVQ have a level of detail proportional to the number of quantization steps. Subsequently, with 5 levels of quantization, the obtained EDMs are closest to the source data. In each case the oxygen atoms of the sulfony, carbonyl and/or carboxy groups lead to peaks in the EDMs obtained after 5 levels of quantization. Such groups should thus easily match together in a superposition algorithm using a density based

similarity measure. Other atoms in the functional groups like for e.g. nitrogen and sulfur (see Figure 5) are not detected at this resolution.

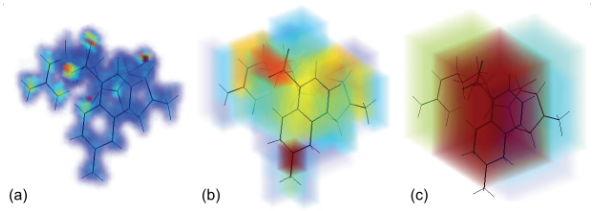


Figure 6: *3-D MQPA EDM (resolution 113×96×93) with 5, 3 and 2 levels of TSLVQ (Z3, b2), respectively.*

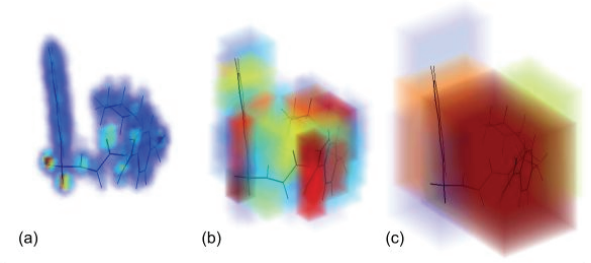


Figure 7: *3-D NAPAP EDM (resolution 120×98×85) with 5, 3 and 2 levels of TSLVQ (Z3, b2), respectively.*

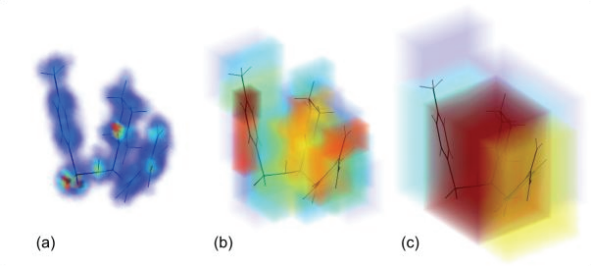


Figure 8: *3-D 4-TAPAP EDM (resolution 119×94×89) with 5, 3 and 2 levels of TSLVQ (Z3, b2), respectively.*

The terminal amidino groups become visible in the third quantization level in which they bear clear EDM peaks. At the same time also the other nitrogen atoms like the ones in the hetero-cycles become visible. Independent on the quantization level no peaks are identifiable with the alkyl part of MQPA implying that chemical groups composed of only C atoms can remain undetected. We do expect however that aromatic rings connected to functional groups to produce clear peaks in the EMS due to the disruption in π-conjugation. Further studies are needed to justify this assumption. The order of the peak density values can be approximately summarized as: sulfonyl ≥ (CO(O) > amidino/N > alkyl chain.

The above analysis, although performed at a qualitative level, agrees well with the results obtained from employing wavelet quantization [1]. Some differences exist though like for e.g. the identification of the sulfonyl groups which is done in our case based on the peaks created by the oxygen atoms rather than by sulfur. This is why in our case sulfonyl groups become less distinguishable from other oxygen bearing functional groups like carboxyl. Nevertheless, we can conclude that the same atoms or functional groups can be identified at certain quantization levels of all molecules considered in this study which could in principle make the theoretical assignment of a chemical group to a peak possible. However, this assertion needs to be validated through further work implementing for e.g. critical points analysis and graph representation of the electron density topology.

5. CONCLUSION

In this paper we showed that TSLVQ is well adapted to the simplification of EDMs. Using a cubic lattice we obtain a set of descriptions hierarchically organized in levels (with one cube file per level): the deeper the level the finer the resolution. Result are presented for three molecules (MQPA, NAPAP, 4-TAPAP) and compared qualitatively with the ones obtained by Leherte [1]. The next step will be to include the computation of critical points directly into the 3-D hierarchical description (the tree-structure).

REFERENCES

[1] L. Leherte, “Application of multiresolution analyses to electron density maps of small molecules: Critical point representations for molecular superposition,” *Journal of Mathematical Chemistry*, vol. 29, 2001, pp. 47-83.

[2] J. Burton, N. Meurice, L. Leherte, and D.P. Vercauteren, “Can Descriptors of the Electron Density Distribution Help To Distinguish Functional Groups?,” *Journal of Chemical Information and Modeling*, vol. 48, 2008, pp. 1974-1983.

[3] P.A. De-Alarcón, A. Pascual-Montano, A. Gupta, and J.M. Carazo, “Modeling shape and topology of low-resolution density maps of biological macromolecules.,” *Biophys J*, vol. 83, 2002, pp. 619-632.

[4] A. Gersho and R.M. Gray, *Vector quantization and signal compression*, Norwell, MA, USA: Kluwer Academic Publishers, 1992.

[5] V. Ricordel, “Etude d un schéma de quantification vectorielle algébrique et arborescente. Application à la compression de séquences d'images numériques, PhD thesis,” Université Rennes 1, 1996.

[6] J.H. Conway, N.J.A. Sloane, and E. Bannai, *Sphere-packings, lattices, and groups*, New York, NY, USA: Springer-Verlag New York, Inc., 1993.

[7] Y. Linde, A. Buzo, and R. Gray, “An Algorithm for Vector Quantizer Design,” *IEEE Transactions on Communications*, vol. 28, 1980, pp. 84-95.

[8] V. Ricordel and C. Labit, “Vector Quantization by packing of embedded truncated lattices,” *IEEE International Conference on Image Processing (ICIP)*, Washington DC: 1995.

[9] V. Ricordel and C. Labit, “Tree-Structured Lattice Vector Quantization,” *European Signal Processing Conference (EUSIPCO)*, Trieste, Italy: 1996.

[10] P. Bourke, “Gaussian Cube Files specifications - http://paulbourke.net/dataformats/cube/,” 2003.

[11] P.L.A. Popelier, “Molecular similarity and complementarity based on the theory of atoms in molecules,” *Molecular similarity in drug design, Dean, P. M.*, 1995, pp. 215-240.

[12] P.L.A. Popelier, “Integration of atoms in molecules: A critical examination.,” *Mol. Phys*, vol. 87, 1996, p. 1169–1187.

Bibliographie

[AM00] Nathan Argaman et Guy Makov : Density functional theory : An introduction. *American Journal of Physics*, 68(1):69, juin 2000.

[BÓ1] Jean-louis Bénard : Livre Blanc : Méthodes Agiles - Etat des lieux. *Business Interactif*, 2001.

[BCCSX05] Chandrajit Bajaj, Julio Castrillon-Candas, Vinay Siddavanahalli et Zaiqing Xu : Compressed Representations of Macromolecular Structures and Properties. *Structure*, 13(3):463–471, 2005.

[Bec99] K Beck : *Extreme Programming Explained*. Addison-Wesley, 1999.

[BG01] Leonid A Bendersky et Frank W Gayle : Electron Diffraction Using Transmission Electron Microscopy. *Journal Of Research Of The National Institute Of Standards And Technology*, 106(6):997–1012, 2001.

[BMLV08] Julien Burton, Nathalie Meurice, Laurence Leherte et Daniel P Vercauteren : Can Descriptors of the Electron Density Distribution Help To Distinguish Functional Groups ?

Journal of Chemical Information and Modeling, 48(10):1974–1983, 2008.

[Bou03] Paul Bourke : Gaussian Cube Files specifications - http ://-paulbourke.net/dataformats/cube/, 2003.

[Bro25] Louis De Broglie : *Recherches sur la théorie des quanta.* Thèse de doctorat, Sorbonne, 1925.

[Cha11] Hind Charaf : Rapport de recherche et développement Visualisation 3D de molécules par quantification vectorielle. Rapport technique, Polytech'Nantes, 2011.

[CSB93] J H Conway, N J A Sloane et E Bannai : *Sphere-packings, lattices, and groups.* Springer-Verlag New York, Inc., New York, NY, USA, 1993.

[DAPMGC02] P A De-Alarcón, A Pascual-Montano, A Gupta et J M Carazo : Modeling shape and topology of low-resolution density maps of biological macromolecules. *Biophys J*, 83(2): 619–632, 2002.

[DM10] Jérémy Duval et Teddy Martin : Rapport de recherche et développement Visualisation 3D de molécules par quantification vectorielle. Rapport technique, Polytech'Nantes, 2010.

[Esc11] Helmut Eschrig : *Topology and Geometry for Physics.* Springer, 2011.

[Gab01] Pete Gabor : Morse Theory. *Mol. Phys*, 2001.

[GG92] Allen GERSHO et Robert M GRAY : *Vector quantization and signal compression.* Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[Jen99] Frank JENSEN : *Introduction to Computational Chemistry*, volume 2. Wiley, 1999.

[JV91] Yves JEAN et François VOLATRON : *Les orbitales moléculaires en chimie : introduction et applications.* McGRAW-HILL, 1991.

[KA54] Harold KLUG et Leroy ALEXANDER : *X-ray diffraction procedures.* John Wiley and Sons, 1954.

[Kav07] Lydia E. KAVRAKI : Molecular Shapes and Surfaces, 2007.

[LBG80] Y LINDE, A BUZO et Robert M GRAY : An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications*, 28(1):84–95, 1980.

[Leh01] Laurence LEHERTE : Application of multiresolution analyses to electron density maps of small molecules : Critical point representations for molecular superposition. *Journal of Mathematical Chemistry*, 29(1):47–83, 2001.

[Lev91] Ira N LEVINE : *Quantum Chemistry*, volume 171. Prentice Hall, 1991.

[Lew03] Errol G. LEWARS : *Computational Chemistry.* Kluwer Academic Publishers, Norwell, MA, USA, 2003.

[Luo10] Siya LUO : 3D Visualization of molecules by lattice vector quantization. Mémoire de D.E.A., Polytech'Nantes, 2010.

[MG01] Jean Pierre Mercier et Pierre Godard : *Chimie Organique Une Initiation.* Presses Polytechnique et Universitaires Romandes, 2001.

[Mil63] J W Milnor : *Morse Theory*, volume 71 de *Annals of Mathematics Studies no. 51.* Princeton University Press, 1963.

[Nak03] M Nakahara : *Geometry, Topology and Physics*, volume 822 de *Lecture Notes in Physics.* Institute of Physics Publishing, 2003.

[OPHS11] Tobias Oetiker, Hubert Partl, Irene Hyna et Elisabeth Schlegl : The Not So Short Introduction to LaTeX 2. *CTAN*, page 171, 2011.

[Pau08a] Renée Paugam : *Initiation à la modélisation moléculaire.* Université Paris Sud, 2008.

[Pau08b] Loïc Paulevé : Euclidean lattices for high dimensional indexing and searching. Research report, INRIA, 2008.

[Ric96] Vincent Ricordel : *Etude d'un schéma de quantification vectorielle algébrique et arborescente. Application à la compression de séquences d'images numériques.* Phd thesis, Université Rennes 1, 1996.

[RL95] Vincent Ricordel et Claude Labit : Vector Quantization by packing of embedded truncated lattices. *In IEEE International Conference on Image Processing (ICIP)*, Washington DC, 1995.

[RRCC11] Cédric Ramassamy, Vincent Ricordel, Oana Cramariuc et Bogdan Cramariuc : Lattice Vector Quantization for the Analysis of Molecular Data. *In SPAMEC 2011 proceeding*, volume 6597, pages 1–4, Cluj-Napoca, Romania, 2011.

[Sch26] E Schrödinger : An Undulatory Theory of the Mechanics of Atoms and Molecules. *Phys. Rev.*, 28(6):1049–1070, 1926.

[TN86] H Takeuchi et Ikujiro Nonaka : The new new product development game. *Harvard Business Review*, 64(1):137–146, 1986.

Table des figures

Résumé

Nous présentons un nouveau schéma de simplification de données moléculaires utilisant la quantification vectorielle algébrique et arborescente (QVAA). Cette méthode, basée sur l'emboîtement de réseaux réguliers tronqués, propose aussi une description hiérarchique d'un volume trimensionnel à travers un arbre de données. Nous appliquons la QVAA pour simplifier des carte de densité électronique (EDM) de trois anticoagulants (MQPA, NAPAP, 4-TAPAP).

Mots-clés : modélisation moléculaire, densité électronique, simplification, arbre de données, réseau régulier de points, quantification vectorielle, multirésolution, similarité moléculaire, MQPA, NAPAP, 4-TAPAP.

Abstract

We introduce a novel simplification scheme of molecular data using Tree-Structured Lattice Vector Quantization (TSLVQ). The method, based on the embedding of truncated lattices, permits also hierarchical description of the $3D$ volume through a tree-structure. We apply TSLVQ to simplify the electron densities maps of three thrombin inhibitors (MQPA, NAPAP, 4-TAPAP).

Keywords : computational chemistry, electron density, simplification, tree-structure, lattice, vector quantization, multiresolution, molecular similarity, MQPA, NAPAP, 4-TAPAP.