UNIVERSITÉ PARIS XI - U.F.R. DE SCIENCES

THÈSE

pour obtenir le titre de

Docteur de l'Université Paris XI
Spécialité Informatique

Présentée par

Nicolas FÉREY

Exploration Immersive de Données Génomiques Textuelles et Factuelles

vers une approche par "Visual Mining"

---

le 14 décembre 2006, à 14h, devant le jury composé de

Mme	Monique	Bolotin-Fukuhara	Examinatrice
M.	Rachid	Gherbi	Directeur de Thèse
M.	Guy	Mélançon	Rapporteur
M.	Ahmed	Seffah	Rapporteur
Mme	Anne	Vilnat	Examinatrice

Thèse préparée au sein du
Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur
LIMSI - CNRS

Remerciements

Always two they are, no more, no less...
...a Master and an Apprentice

Maitre Yoda

Mes premiers remerciements vont à Rachid Gherbi, qui m'a motivé durant le stage de DEA, pour me dépasser afin d'obtenir une bourse de thèse pour continuer la route avec lui. Son encadrement m'a incité à publier très tôt, les travaux réalisés durant cette thèse, et les multiples voyages pour présenter nos articles ont construit dans l'équipe des relations qui ont vite dépassées le cadre professionnel, avec Joan Hérisson, Pierre-Emmanuel Gros, plus récemmement Claire Toffano-Nioche et Christine Martin.

Un grand merci à mes rapporteurs Guy Melançon et à Ahmed Seffah, qui malgré un laps de temps assez court pour évaluer et corriger ce travail, ont fourni dans leur rapport de nombreux commentaires très constructifs pour corriger et améliorer ce manuscrit. Merci aussi à Anne Vilnat et Monique Bolotin-Fukuhara pour avoir accepter d'examiner ce travail.

Un merci plus large à l'ex-groupe GI (Gestes et Images) et au membre de l'action transversale VENISE (Virtual ENvironment for Immersive Simulation and Experiments : www.limsi.fr/venise) et à ses doctorants, Guillaume Bouyer, Bruno Bossard, Thomas Convard, Nicolas Fauvet, Antoine Tarault, qui ont chacun contribué à une ambiance de travail qui a fait la légende du bâtiment 502 bis. Un petit clin d'oeil en passant aux collègues mécaniciens des fluides du 502bis, qui ont supporté nos conversations bruyantes et pas toujours scientifiques. Dans l'action transversale VENISE, une pensée plus particulière pour Patrick Bourdot et Jean-Marc Vézien, qui se battent quotidiennement pour mettre à disposition de l'ensemble du laboratoire et aux générations de futures, un environnement de réalité virtuelle, avec un savoir faire d'une grande qualité.

Je tiens d'autre part à remercier toutes les personnes qui ont construit et continuent à faire vivre le Plan Pluri Formation Bioinformatique de l'Université Paris Sud, en favorisant des échanges interdisciplinaires de qualité, propices à l'acquisition d'une compétence bi-disciplinaire, qui me fût nécessaire durant cette thèse.

Je remercie aussi tous les bioinformaticiens pour avoir consacré une partie de leur temps à tester et à améliorer *Genome3DExplorer*, avec un remerciement tout particulier à Claire Toffano-Nioche pour ses relectures attentives, et Oriane Matte-Taillez pour les données biologiques, toujours plus complexes les unes que les autres.

Pour finir, un grand merci à Amélie, ma femme, qui, telle une sirène pour un marin, la réussi quotidiennement à me détourner de cette longue traversée en mer que constitue un travail de la thèse, mais sans qui, paradoxalement, je n'aurais jamais pu entendre crier "terre à l'horizon".

A toute ma famille

Résumé

Ce travail de thèse porte sur l'exploration immersive de données génomiques textuelles et factuelles. Il s'agit d'étudier et de concevoir une nouvelle approche pour explorer dans un cadre immersif (i.e. de réalité virtuelle), des données d'une nature spécifique : les données génomiques. Ces données se présentent sous deux formes : la forme factuelle, c'est-à-dire l'ensemble des données structurées provenant des banques de données biologiques, et la forme textuelle, à savoir l'ensemble des données non structurées présentes dans les millions de publications concernant la génomique. Face à la complexité de ces données, l'enjeu est de proposer des paradigmes d'exploration et d'interaction, qui permettent de couvrir le plus largement possible ces données de nature variée. Ces paradigmes doivent être capables de gérer des données massives, et doivent être adaptés à la fois au cadre immersif et aux besoins des biologistes. Ainsi sur la base de l'étude des besoins des utilisateurs et de leurs usages, un paradigme de représentation a été défini en utilisant les caractéristiques très spécifiques des banques de données génomiques, et dans lequel la sémantique des données étudiées est traduite par des propriétés visuelles, géométriques, ou topologiques, choisies initialement par l'utilisateur. Un prototype a été implémenté pour tester et valider le paradigme précédemment défini, et plusieurs expériences d'analyse de données génomiques ont été menées en collaboration avec des biologistes ou bioinformaticiens, afin de mesurer la pertinence de cette solution d'exploration, face à plusieurs problématiques d'analyse de données génomiques.

Mots clefs : Données Génomiques, Exploration de Données Massives, Visualisation Dynamique de Graphes 3D, Réalité Virtuelle, Interaction Immersive.

Abstract

This thesis concerns the immersive exploration of textual and factual genomic data. The goal of this work is to design and study new approach for exploring genomic data within an immersive framework (i.e. of virtual reality). The knowledge about genome is constituted by factual data, coming from structured biological or genomic databanks, and by textual data, namely the unstructured data within the millions publications relating to the research about genome. These data are heterogeneous, huge in quantity, and complex. The stake of this work is to propose visualization and interaction paradigms, which are able to deals with these characteristics. These paradigms must also be adapted to the immersive framework, and must respect the needs of the biologists. We used common points of genomic databanks, to design an original visualization paradigm, where the user is able to choice a translation of the semantic of the genomic data to visual, geometric or topologic properties. We implemented a software prototype in order to test and validate the visualization paradigm within an immersive framework. In this context, we proposed and tested new interaction paradigms, in order to navigate, search and edit the genomic data during the immersive exploration. We used finally this software to lead several experiments of genomic data analysis with biologists, in order to measure the relevance of this visual mining approach on different kinds of genomic data.

Keywords : Genomical Data, Large Dataset Exploration, Dynamic Visualisation, Graph drawing, Virtual Reality, Immersive Interaction.

Table des matières

Remerciements iii

Résumé v

Table des matières vii

Introduction 1

1 Outils de visualisation des génomes 7
- 1.1 Les séquences . . . 8
  - 1.1.1 *FlagDB++* : visualisation d'une base de données de séquences d'ADN concernant *Arabidopsis thaliana* . . . 10
  - 1.1.2 *Sequence Logo* : visualisation de séquence consensus . . . 12
  - 1.1.3 *GeneWiz* : visualisation de paramètres structuraux d'ADN à partir de séquences de génomes microbiens . . . 13
- 1.2 Les structures moléculaires . . . 16
  - 1.2.1 *RasMol*, *YASARA*, et Cie : visualisation 3D de structures protéiques . . . 16
  - 1.2.2 *ADN-Viewer* : exploration immersive de structures nucléiques massives . . . 18
- 1.3 Les données d'expression . . . 22
- 1.4 Les données textuelles . . . 25
  - 1.4.1 *BioBibliometrics* : navigation dans la bibliographie des génomes 25
- 1.5 Les banques de données factuelles . . . 26
  - 1.5.1 *GenoMEDIA* et *Lysa* : intégration et visualisation collaboratives de banques de données factuelles . . . 27
  - 1.5.2 *Sequence World* : exploration tridimentionnelle de banques de données génomiques . . . 28
  - 1.5.3 *SARAGene* : exploration immersive de banques de données génomiques . . . 29
- 1.6 Conclusion . . . 30

2 Données génomiques : de leur caractérisation au contexte de leur exploration **31**
2.1 Caractérisation des connaissances génomiques . . . 31
2.1.1 Une croissance exponentielle de contenu . . . 32
2.1.2 Une hétérogénéité selon de multiples points de vue . . . 32
2.1.2.1 Une hétérogénéité de qualité . . . 32
2.1.2.2 Une hétérogénéité de contenu . . . 32
2.1.2.3 Une hétérogénéité technique . . . 36
2.1.3 Des points communs propres aux données génomiques . . . 36
2.1.3.1 Organisation autour d'un seul type d'objet biologique 36
2.1.3.2 Structuration en réseau de relations . . . 37
2.1.3.3 Conclusion . . . 38
2.2 Contexte de réalité virtuelle . . . 39
2.2.1 La réalité virtuelle : définition . . . 39
2.2.2 Les techniques de simulation de l'immersion . . . 40
2.2.2.1 Une surface d'affichage plus grande que le champs de vision . . . 40
2.2.2.2 La stéréoscopie . . . 41
2.2.3 Les périphériques d'interaction . . . 42
2.2.3.1 Le tracking de l'utilisateur . . . 43
2.2.3.2 La souris 3D . . . 43
2.2.3.3 Les gants de données . . . 43
2.2.3.4 Les interfaces haptiques . . . 45
2.2.4 Une gestion distribuée des périphériques . . . 45
2.3 Conclusion . . . 46

3 Représentation, exploration et interaction **47**
3.1 Modèle générique de représentation des données génomiques . . . 47
3.2 De la représentation informatique à la représentation graphique . . . 50
3.2.1 Représentation sous forme de graphes 3D . . . 50
3.2.1.1 Inventaire des attributs graphiques d'un graphe 3D . 50
3.2.2 Séparation entre données et représentations . . . 51
3.2.2.1 La couche *donnée* . . . 52
3.2.2.2 La couche *représentation* . . . 53
3.2.3 Normalisation de valeurs quantitatives . . . 55
3.2.4 Gestion et affichage des étiquettes . . . 56
3.3 Traduction d'une proximité biologique par une distance spatiale . . . 58
3.3.1 Approximation par simulation de forces . . . 58
3.3.1.1 Optimisation . . . 66
3.3.1.2 Complexité . . . 70
3.3.1.3 Discussion . . . 70

3.3.1.4 Dessin réactif à l'interaction . . . 71
3.4 La navigation . . . 71
3.4.1 Les déplacements locaux . . . 72
3.4.2 Les déplacements globaux . . . 72
3.4.3 La légende . . . 74
3.5 Un filtrage interactif des données . . . 74
3.5.1 Filtrage sélectif . . . 74
3.5.2 Un filtrage par seuillage interactif . . . 75
3.6 La modification des données et l'annotation . . . 76
3.6.1 La mise à jour, l'ajout, la supression . . . 76
3.6.2 Le problème du "Text Input" dans un cadre immersif . . . 76
3.7 Sonification de données . . . 77
3.8 Une architecture logicielle pour visualiser les données massives . . . 78
3.9 Interopérabilité et retour d'information . . . 79
3.10 Conclusion . . . 83

4 Applications et résultats 85
4.1 Analyse de l'architecture des génomes . . . 85
4.1.1 Analyse des duplications interchromosomiques chez la levure . 86
4.1.2 Caractérisation des introns/exons chez *Arabidopsis thaliana* . 89
4.1.2.1 Objectif . . . 89
4.1.2.2 Données . . . 90
4.1.2.3 Méthode . . . 90
4.1.2.4 Résultats et discussion . . . 93
4.1.3 Co-régulations intra-chromosomiques chez la levure . . . 96
4.1.3.1 Problématique . . . 96
4.1.3.2 Données *in silico* . . . 97
4.1.3.3 Résultats . . . 98
4.1.3.4 Discussion . . . 99
4.2 Exploration d'un réseau d'alignements massif de séquences protéiques 100
4.2.1 Données . . . 100
4.2.2 Problématique . . . 101
4.2.3 Méthode et résultats . . . 101
4.3 Exploration de données d'expression issues de puces à ADN . . . 103
4.3.1 Objectif . . . 103
4.3.2 Description des données de puces à ADN . . . 103
4.3.3 Exemple d'inter-opérabilité avec d'autres sources d'informations 104
4.3.4 Résultats, validation et conclusion . . . 105
4.4 Exploration de connaissances bibliographiques . . . 107
4.4.1 Données . . . 107
4.4.2 Résultats . . . 107

4.5 Conclusion 109

Conclusion **113**

Table des figures **123**

Liste des algorithmes **125**

Bibliographie **134**

A Le code génétique **135**

B Exemple de fichiers *PDB* (Cytochrome p450) **137**

C Duplications inter-chromosomiques chez la levure **139**

D Degré d'accessibilité des gènes **141**

E Les octrees, une méthode dichotomique de gestion de l'espace **143**

Introduction

Le commencement est beaucoup plus que la moitié de l'objectif

Aristote

Dans cette introduction, nous situerons dans un premier temps le contexte général dans lequel s'est déroulé ce travail, puis nous présenterons l'objectif et la démarche suivie durant cette thèse. Nous dresserons ensuite la problématique associée à l'exploration de données génomiques textuelles et factuelles, en expliquant dans quelle mesure la nature des données génomiques rend ce problème difficile. Nous terminerons enfin par une description de l'organisation de ce mémoire.

Contexte général

Pendant ces dix dernières années, la communauté de la recherche sur le génome a été témoin d'une augmentation importante de la quantité de connaissances stockées dans les bases de données génomiques. La plus célèbre illustration de cette situation est l'achèvement du séquençage du génome humain dans le cadre du projet international *Human Genome Project* [6]. Si le résultat de ce travail a permis d'identifier et de situer une part importante des gènes de l'espèce humaine, il faut souligner le fait que ces gènes jouent un rôle plus limité que prévu dans la complexité du vivant, en pointant l'importance que jouent les phénomènes épigénomiques. La société a été alors le témoin d'un changement d'échelle par le passage de l'ère de la génétique, à l'ère de la génomique. En effet, si la génétique s'efforce d'étudier les gènes et leur séquence, support de l'hérédité, avec une vision très locale, la génomique est une approche plus globale des génomes dans leur environnement. Elle étudie plusieurs milliers de gènes à la fois, dans leur contexte, avec leurs interactions, dans l'espace et dans le temps. Le résultat de ce changement d'échelle eut pour conséquence à un apport exponentiel de nouvelles connaissances, rendues disponibles à la communauté grâce à leur accès public sur Internet.

Approche générale

Ces données, support informatique de cette nouvelle connaissance, constituent une importante source de données factuelles, et peuvent être traitées de façon manuelle ou automatique. Un processus clé pour les biologistes est l'extraction ou la découverte de connaissances, latentes dans les données brutes. Par exemple, le résultat du séquençage de l'ADN, une longue suite de lettres A,C,G T, reflète une réalité chimique, mais est vide de sens, si on ne connaît pas ses modalités de lecture. Par contre, à la lumière du code génétique, ces données permettent d'extraire les zones codantes, ainsi que les zones intergéniques, de fonctions différentes, mais non moins utiles. Ainsi les données brutes, associées à des connaissances contextuelles, permettent d'inférer de nouvelles hypothèses qui débouchent sur de nouvelles connaissances validées sur paillasse. Ces informations contextuelles trouvent leur source d'abord dans l'expertise du biologiste, ensuite dans les publications associées au génome, (données textuelles), mais aussi dans les banques de données structurées (données factuelles).

Pour explorer ces données génomiques par nature massives, les recherches menées durant cette thèse s'inscrivent dans une démarche que nous pourrions appeler *Visual Mining*, parallèle et complémentaire à l'approche du *Data Mining*. Dans les deux approches, le but est d'inférer de la connaissance, implicite dans les données brutes. Dans le *Data Mining*, cette extraction est effectuée de façon semi-automatique à partir d'une expertise formalisée des experts du domaine. Le *Visual Mining* est une approche plus exploratoire, qui exploite la perception visuelle pour analyser des données laissées le plus brut possible, en minimisant le nombre de traitements automatiques, afin de maximiser l'exploitation des compétences naturelles et de l'expertise de l'utilisateur, souvent difficilement formalisables.

Problématique et objectifs

L'objectif de cette thèse est de fournir des outils d'exploration visuelle de grande quantité de données génomiques factuelles et textuelles. En effet, les connaissances accumulées par les biologistes, vérifiées de façon expérimentale avec différents niveaux de certitude, ou prédites, sont principalement stockées sous deux formes. D'une part, elles peuvent être publiées, et constituent un ensemble des données bibliographique textuelles non structurées sur le génome, car exprimées en langage naturel. D'autre part, elles sont stockées dans des banques de données, de façon structurée. Par exemple, les résultats d'expériences biologiques permettant d'obtenir la structure spatiale de protéines, sont d'abord publiées, puis stockées dans une banque de données spécialisée dans ce type de données, *Protein Data Bank* [12]. De récents travaux se sont attachés à fournir des outils de visualisation et d'exploration de ces structures moléculaires tridimensionnelles massives comme *YASARA* [19] pour les structures protéiques, ou *ADN-Viewer* [64], pour les structures nucléiques, mais la visualisation et l'exploration dans les banques de données génomiques non moléculaires, qu'elles soient factuelles ou textuelles, est un problème moins étudié.

L'exploration visuelle de données génomiques a pour principal objectif, l'extraction de connaissances à partir la représentation visuelle de ces données, difficilement interprétables à partir des données brutes. On peut aujourd'hui extraire de la connaissance à partir de données brutes en utilisant un large éventail d'outils bioinformatiques de visualisation ou d'extraction de connaissances, mais ces outils d'analyse des connaissances génomiques sont souvent considérés ou utilisés comme des boites noires par leurs utilisateurs, et certains ne laissent que très peu de place à l'expertise des chercheurs travaillant sur les génomes. Ces outils sont par ailleurs construits par rapport à une tâche spécifique, ce qui laisse peu de place à la réutilisabilité, et rend très difficile le croisement de connaissances génomiques variées. Par ailleurs, ce processus d'analyse est rendu assez difficile par l'aspect massif de ces données acquises de manière automatique. Par exemple, la capacité des puces à ADN a été multipliée par 1000 en quelques années, et les outils bioinformatiques de visualisation de ces connaissances les plus utilisés, comme les dendogrammes de Eisen [45], peinent à gérer ce changement d'échelle, parfois pour des raisons algorithmiques, mais aussi à cause de problématiques liées à la visualisation de données massives. Sur un autre plan, cette acquisition de nouvelles connaissances est très rapide et a abouti à une très forte hétérogénéité, à la fois en terme de stockage que de qualité. Le but de mon travail est donc de faciliter le processus d'exploration de ces données massives, en proposant aux biologistes un cadre et des outils adaptés à ces nouveaux besoins.

D'un point de vue informatique, la difficulté est de développer des méthodes qui permettent une gestion efficace de données massives autant en terme de représen-

tation informatique, que de représentation visuelle. Donner la possibilité d'explorer ces données de manière simultanée, de façon à créer des ponts entre connaissances génomiques textuelles et factuelles d'une variété importante, nécessite de choisir des méthodes d'exploration et de visualisation qui soient assez génériques et réutilisables.

D'un point de vue biologique, il est important de trouver des méthodes d'exploration adaptées à la spécificité des connaissances génomiques, et plus généralement biologique, sans rompre avec les usages et les objectifs des biologistes lorsqu'ils utilisent leurs banques de données.

Le caractère pluridisciplinaire de notre approche est important, et nécessite un effort d'acquisition d'une double compétence, pour comprendre la nature et le contenu des données génomiques, et comprendre les problèmes de chercheurs qui ne font pas partie de la communauté des sciences de l'information. Ce document a donc été écrit afin d'être accessible aux deux communautés.

Organisation du mémoire

Le sujet de thèse s'intitule "Exploration Immersive de Données Génomiques Textuelles et Factuelles". Il s'agit, d'une part, de visualiser des données d'une nature spécifique, les données concernant les génomes. Ces données génomiques sont présentes sous deux formes : la forme factuelle, c'est-à-dire l'ensemble des données structurées et semi-structurées provenant des banques de données biologiques, et la forme textuelle, à savoir l'ensemble des données non structurées présentes dans les millions de publications concernant le génomique. D'autre part, cette exploration visuelle est qualifiée d'"immersive", car elle s'effectue dans le contexte spécifique que constitue la réalité virtuelle. Dans cette dernière, les concepteurs s'attachent à donner à l'utilisateur, l'impression d'être immergé dans son objet d'étude.

Le premier chapitre de cette thèse sera donc l'occasion de proposer un état de l'art sur les travaux consacrés à la visualisation et à l'exploration de données génomiques variées.

Le double cadre de ce travail, dont l'objet d'étude est l'exploration de données génomiques, et ceci, dans un contexte immersif, sera décrit dans le second chapitre, pour à la fois caractériser les spécificités inhérentes aux données génomiques, mais aussi pour décrire précisément les techniques et les dispositifs utilisés dans ce contexte afin d'offrir à l'utilisateur l'illusion d'immersion au coeur de ses données. Ce chapitre dégagera les besoins et les spécificités des domaines traités et sera nécessaire à la compréhension des solutions proposées par la suite.

Face aux données génomiques qui présentent de multiples complexités, l'enjeu majeur est de trouver des paradigmes pour leur exploration interactive, capables de

couvrir une grande variété de données génomiques et de gérer des données massives. Par ailleurs, ces paradigmes devront satisfaire les contraintes d'interactivité, exploiter les bénéfices du cadre immersif et enfin répondre aux besoins des biologistes. Nous décrirons dans le troisième chapitre, comment nous avons utilisé les spécificités des bases/banques de données génomiques, ainsi que leur points communs avec les données textuelles, pour construire un paradigme de visualisation dans lequel la sémantique des données étudiées est traduite par des propriétés visuelles, géométriques, ou topologiques, choisies initialement par l'utilisateur. Ce chapitre sera aussi l'occasion de décrire plusieurs patterns d'interaction, pour explorer la représentation des données, offerte par le paradigme de visualisation dans le cadre immersif.

Enfin, dès le début de ce travail, l'utilisateur a toujours été mis au centre de la boucle de conception, grâce à l'utilisation sur place d'un prototype, afin de tester et de valider les paradigmes de visualisation et d'exploration précédemment définis. Ceci a permis de préciser, au fur et à mesure, de ses besoins et de ses usages, et de confronter cette méthode d'exploration à des problématiques biologiques. Plusieurs expériences d'exploration de données génomiques variées ont donc été menées en collaboration avec des biologistes, décrites dans le quatrième et dernier chapitre de cette thèse.

Chapitre 1

Outils de visualisation des génomes

Les choses changent. Mais si vite... Est-ce que les habitudes des hommes pourront suivre ?

I. Asimov

Si l'objectif de ce travail de thèse est de proposer des solutions pour explorer, analyser et interagir avec des données génomiques factuelles et textuelles massives, dans le cadre spécifique de réalité virtuelle immersive, la liste de travaux présentée dans ce chapitre est loin de se vouloir exhaustif, mais reflète la variété des données génomiques à travers quelques outils de visualisation.

Ceci sera réalisé sans se limiter aux outils d'exploration dans un cadre exclusivement immersif. En effet, les outils d'analyse bioinformatique développés dans ce contexte sont pour la plupart à l'état de prototype, et ne sont pas encore rentrés dans les usages des chercheurs en bioinformatique. Nous trouvons plus pertinent d'étudier les habitudes actuelles des biologistes face à leurs données, et de présenter des exemples d'outils usuels d'analyse.

Certains outils qui seront présentés dans cette section, sont destinés à la visualisation de données moléculaires complexes, qui ne sont pas l'objet de ce travail. Ils seront présentées car ils se sont attachées à résoudre la problématique de gestion et de visualisation de données massives, et ont impacté de façon majeure les solutions proposées pour répondre à cette problématique associée à la visualisation d'un volume important de données.

Enfin, si ce chapitre se veut le reflet de l'usage des biologistes par rapport à l'activité "visualisation" de leurs données, il a aussi pour objectif de déterminer à la fois les limites et les frontières de ce domaine en terme de visualisation de données génomiques. En effet, certains outils présentés ici ne sont utilisés que ponctuellement, mais constituent un apport non négligeable en offrant des solutions innovantes qui

modifient profondément la façon avec laquelle les biologistes envisagent leurs objets d'étude. Dans ce sens, ces outils constituent donc une frontière en matière de recherche en visualisation de données génomiques, et certains d'entre eux rentreront peut-être dans un futur proche dans les usages des chercheurs sur le génome.

L'organisation de ce chapitre est le reflet de la variété des objets d'intérêt que les outils de visualisation du génome, de la simple séquence d'ADN, en passant par les protéines, vers la visualisation plus globale de banques de données, qu'elles soient textuelles ou factuelles.

1.1 Les séquences

Les objets qui sont au centre de l'étude des génomes sont les séquences. Il existe plusieurs types de séquence : les séquences nucléiques, ou les séquences protéiques. D'un point de vue informatique, ces objets sont particulièrement adaptés à des analyses automatiques, car elles constituent des mots avec un alphabet fini (4 symboles pour les séquences d'ADN, et 20 pour les séquences protéiques). Plusieurs approches permettent de traiter les séquences de façon automatique, dont les plus usitées sont les suivantes :

- L'alignement, qui permet de comparer les séquences entre elles, en donnant un score de similitude, et en extrayant les zones où ces séquences possèdent des sous-séquences communes. On peut comparer deux séquences ou un ensemble de séquences. Dans le dernier cas, les portions communes d'un ensemble de séquences alignées, sont appelées séquences consensus, les caractères non conservés étant remplacés par des caractères génériques.
- La recherche de motif, qui permet de trouver un patron prédéfini dans un ensemble de séquences.
- L'analyse statistique est elle aussi beaucoup utilisée pour extraire des caractères lexicographiques ou numériques présents dans la séquence, qui ont un impact biologique important, comme par exemple la richesse en nucléotides représentés par les lettres G et C dans les séquences d'ADN.
- Enfin, l'apprentissage, statistique ou symbolique, est une méthode particulièrement adaptée à la construction de modèles, lorsqu'ils ne peuvent être définis de façon déterministe. Les grandes classes d'algorithme d'apprentissage utilisés sont les arbres de décisions, les SVM, les réseaux de neurones, les modèles de Markov, et les algorithmes génétiques.

Cependant, les analyses automatiques ont leurs limites : elles permettent de répondre en général à une question précise, mais ne permettent pas une exploration sans *a priori* : elles ne constituent pas réellement une tâche exploratoire. D'autre part, ces outils automatiques sont souvent perçus ou utilisés comme des boites noires.

C'est la raison pour laquelle, en parallèle avec ces approches automatiques, de nombreux outils de visualisation on été conçus afin de compléter et de contrôler les analyses automatiques. Nous retrouverons donc dans cette section une sélection d'outils de visualisation et d'exploration de séquences que nous pensons les plus représentatifs, soit du fait de leur fréquence d'usage, soit de leur esthétisme, soit de leur facilité d'utilisation.

1.1.1 *FlagDB++* : visualisation d'une base de données de séquences d'ADN concernant *Arabidopsis thaliana*

Une fois le génome d'un organisme partiellement ou totalement séquencé, les biologistes disposent, selon les cas, de règles pour structurer cette séquence. Par exemple, l'identification de codons START (ATG) et STOP (TGA,TAA,TAG), quatre séquences de trois nucléotides retrouvées de façon quasi-universelle dans l'ensemble des organismes vivants, permettent de déterminer le début et la fin d'un gène. Ces connaissance sont utilisées pour annoter les séquences de façon automatique, ce qui permet d'organiser et de structurer ces séquences. Il est donc devenu nécessaire, à la lumière de ces différents modèles qui donnent du sens à la séquence, de développer des outils qui permettent d'explorer ces séquences à échelle du génome. A ce titre, *FLAGdb++*[90] est un exemple intéressant. Cet outil a été au départ conçu pour visualiser le résultat d' expériences biologiques de mutagenèse par insertion, qui consistent à insérer des portions d'ADN étrangers dans un génome, pour en étudier les conséquences, afin de mieux comprendre le fonctionnement de ce génome. Les séquences insérées sont encadrées par des séquences reconnaissables, appelées *FST* (Flanking Sequence Tag), pour de pouvoir ultérieurement localiser précisément dans le génome les séquences insérées. L'outil *FLAGdb/FST* a été développé afin de positionner ces *FST* dans la séquence. Cet outil s'est enrichi de nombreuses autres connaissances, et de liens vers d'autres bases, comme *GeneVestigator* [114], et le résultat est une base de donnée intégrative nommée *FLAGdb++*. Initialement asservi à l'étude du génome d'*Arabidopsis thaliana*, son domaine d'application a été étendu d'autres organismes comme le riz. La représentation permet de visualiser une séquence annotée, avec une fenêtre glissante sur la séquence qui permet de visualiser simultanément quelques milliers de paires de bases. Dans la figure 1.1, la ligne rouge représente la séquence, les numéros, la position dans le chromosome en nombre de bases, et chaque symbole correspond à une annotation. L'utilisateur navigue dans l'intégralité du génome en se positionnant à un niveau plus global sur l'un des cinq chromosomes de l'organisme, à droite sur la figure 1.1.

L'utilisateur peut évidemment choisir quelles annotations visualiser, et le logiciel lui offre une panoplie variée d'outils de traitement et d'analyse de séquences, comme l'alignement multiple, la recherche de motif, la localisation cellulaire.

Cette forme de visualisation de séquences annotées, analogue à celle offerte par l'outil de visualisation et d'exploration des génomes, *NCBI map Viewer* [11], est une des plus utilisées par les biologistes. Elle permet d'explorer les résultats de nombreuses annotations préalablement effectuées de façon manuelle par les biologistes, ou semi-automatique à l'aide d'outils de prédiction. Cette représentation est souvent utilisée pour ajouter manuellement de nouvelles annotations sur ces séquences. Par contre, si ce mode de visualisation de la séquence permet d'analyser la

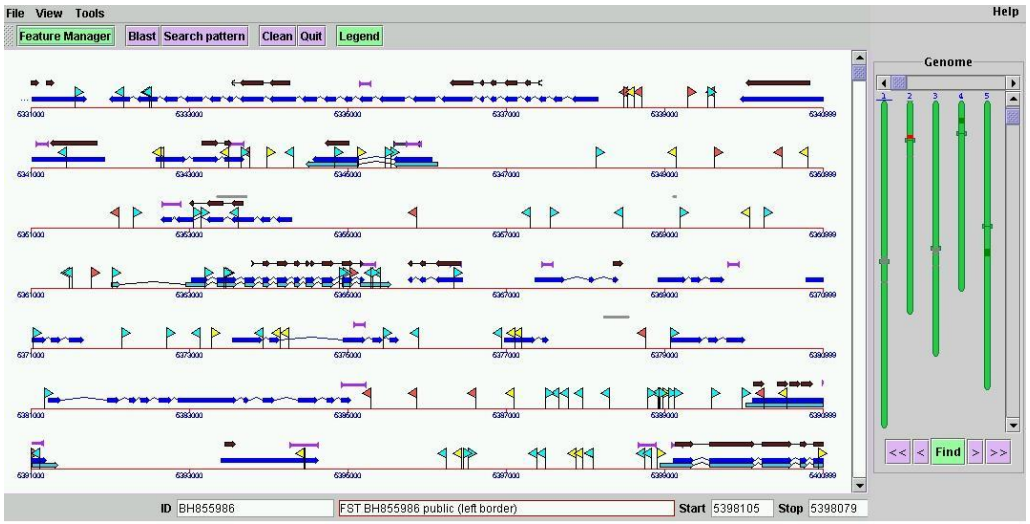


FIG. 1.1 – Interface de visualisation de *FLAGdb++*

structure locale des génomes, elle ne permet pas d'explorer facilement ces séquences d'un point de vue plus global, ce qui rend difficile l'étude de phénomènes non locaux dans les génomes, telles que les duplications de séquences, ou les phénomènes de co-régulations de gènes.

1.1.2 *Sequence Logo* : visualisation de séquence consensus

Sequence LOGO[91] est un paradigme de visualisation de petites séquences consensus (par exemple, les sites de liaison de la figure 1.2), basé sur le principe de la théorie de l'information de *Shannon*. Chaque position de la séquence est représentée par un empilement de tous les symboles de bases (A, C, G, T pour l'ADN). Dans cet empilement, la hauteur de l'empilement est proportionnelle à la conservation d'un symbole à cette position dans l'ensemble des séquences. D'autre part, une hauteur relative pour chaque symbole dans cette empilement reflète la fréquence d'occurrence de ce symbole dans les séquences.

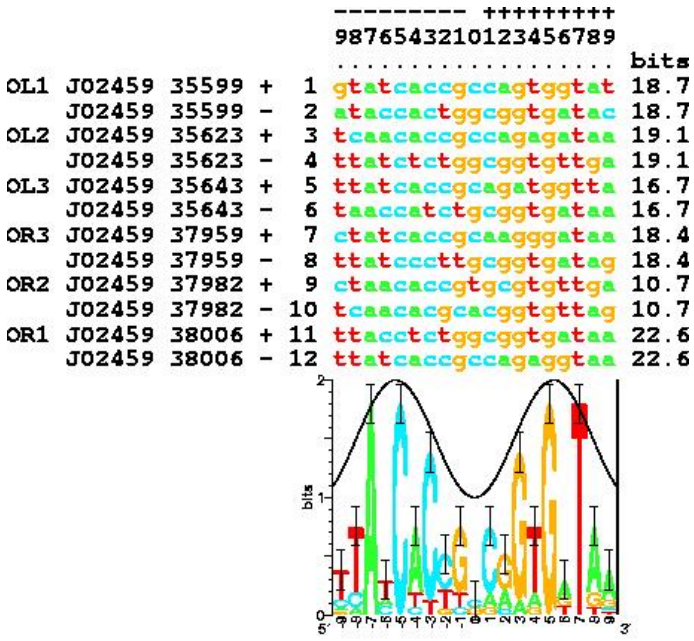


FIG. 1.2 – En haut, alignement de 12 sites de liaison *cI* et *cro*, en bas, représentation de cet alignement avec *Sequence LOGO*

Ainsi dans cette visualisation des séquences consensus, les zones très conservées émergent, mais on garde tout de même l'information sur les zones plus variables, contrairement à la visualisation purement textuelle d'une séquence consensus, utilisée aussi, mais dans laquelle les symboles non conservés sont remplacés par un symbole générique.

L'avantage de *Sequence LOGO* est de représenter de façon élégante les séquences consensus de petite taille (promoteur, zone de régulation,...) mais n'a pas été conçu pour de longues séquences. Il permet une compréhension directe et efficace de séquence d'ADN consensus.

Il est intéressant de constater que l'élégance et l'efficacité de ce paradigme sont

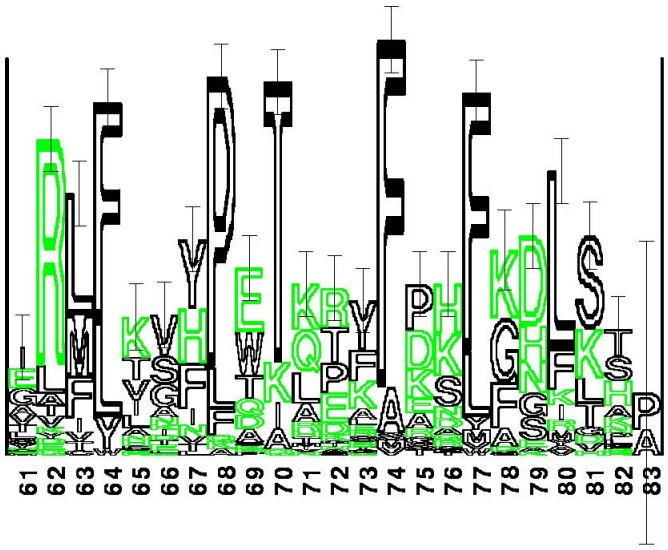


FIG. 1.3 – Consensus de séquence protéines de (Globine)

dues à la redondance de présentation de l'information dans la visualisation. L'information sur le contenu nucléotidique est présentée selon deux modalités. Chaque nucléotide est représenté par une lettre et un couleur spécifique. Le choix de cette double modalité dans cette représentation, permet une compréhension intuitive et efficace de comparaisons de séquences d'ADN constituées de 4 symboles, mais est moins efficace lorsqu'il s'agit de séquences protéiques, constituées de 20 symboles (cf. figure 1.4). Le nombre de couleur dépasse la limite pour laquelle l'utilisateur associe facilement chaque couleur à un symbole, et le nombre de lettres par étage, rend plus difficile la compréhension de ce type de données.

1.1.3 *GeneWiz* : visualisation de paramètres structuraux d'ADN à partir de séquences de génomes microbiens

Sequence LOGO répond au besoin d'étudier certaines caractéristiques de conservation d'une séquences à une échelle très locale. Un autre besoin est celui d'étudier une séquence avec à une échelle plus globale, voire dans sa globalité, pour permettre d'étudier l'architecture des génomes avec plus de recul. Pour y répondre, une équipe du CBS (Center for Biological Sequence Analysis), a développé une méthode de visualisation de paramètres structuraux de séquences d'ADN de génomes batériens [66] [85]. Cet outil a été initialement développé pour l'analyse de génome entier, mais peut aussi fournir une vision plus locale, jusqu'à une échelle de quelques milliers de bases. L'idée fondatrice est de représenter sur un cercle représentant un chromosome ou un génome entier, des propriétés structurales ou mécaniques intrinsèques dans la

séquence, dont voici quelques exemples :

- la courbure intrinsèque de l'ADN ;
- la rigidité ;
- la sensibilité à la déformation par des facteurs protéiques ;
- le taux de AT.

A l'échelle du génome entier, des propriétés architecturales globales peuvent être mises en évidence. A l'échelle de quelques gènes (entre 10000 et 20000 bases), on peut examiner par exemple l'architecture des régions intergéniques.

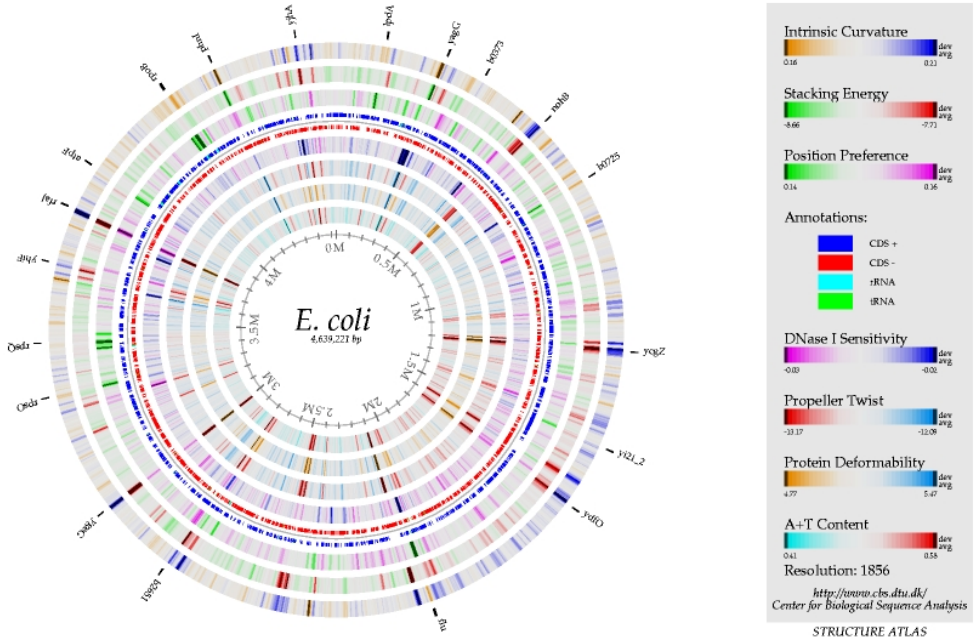


FIG. 1.4 – Visualisation du génome complet d'*Escherichia coli* avec *Genome Structure ATLAS*

Cette visualisation est particulièrement intéressante, car elle donne une vision globale et permet d'étudier ensemble plusieurs critères, comme certains descriptifs quantitatifs de l'architecture spatiale d'une séquence, comme la sensibilité à la déformation, ou la courbure intrinsèque de la séquence, ou d'autres plus classique, comme le taux de AT. Dans cette représentation, la séquence est assimilée à un cercle. Ces différents critères, indiqués dans la légende de haut en bas, correspondent aux cercles imbriqués du plus externe au plus interne. Les représentations radiales, de la plus simple dans *GeneWiz*, ou plus complexe, associée à une navigation multi-échelle comme dans le travail de Stasko et Zhang [101], sont particulièrement adaptées

pour étudier l'architecture des génomes bactériens, car leur séquence d'ADN est circulaire, ce qui n'est pas le cas des eucaryotes.

1.2 Les structures moléculaires

Si l'analyse des séquences nucléiques ou protéiques en tant que séquences textuelles a permis de faire largement avancer les connaissances sur les génomes, la réalité chimique derrière cette représentation est beaucoup plus complexe. En parallèle avec le séquençage systématique des génomes, plusieurs approches expérimentales permettent de déterminer l'agencement spatial des atomes qui composent les biomolécules. On peut ainsi représenter l'organisation spatiale d'ADN, d'ARN, ou de protéine, constituant, avec les molécules de la membranes cellulaires, les briques de base de l'organisation et du fonctionnement dans les organismes vivants.

Il existe principalement deux procédés d'observation de biomolécules, qui permettent d'établir leur cartographie tridimensionnelle complète (i.e. les coordonnées 3d de chaque atome qui constitue la molécule) : la méthode par cristallographie par rayons-X et la méthode par résonance magnétique nucléaire (RMN). Toutes les protéines ne sont pas aisément cristallisable, et quand c'est le cas, la cristallographie par rayon X ne permet pas de déterminer la position de certains atomes comme les atomes d'hydrogène. La méthode RMN permet d'obtenir la position des atomes d'hydrogène, mais se limite à l'étude de protéine de poids inférieur à 30kiloDalton. La méthode RMN est une méthode de choix pour de petites protéines qui ne sont pas aisément cristallisable, car elle permet l'étude des molécules en solution, moins dénaturées que lors de la cristallisation. Elle permet d'obtenir des résultats de structure que l'on pense plus proches des conditions *in vivo*.

L'organisation spatiale des protéines est très importante, car cette structure détermine en partie leur fonction dans la cellule, mais il faut insister sur le fait que les résultats obtenus par les deux méthodes décrites précédemment, permettent d'obtenir la structure tridimensionnelle de la molécule dans l'état dans lequel celle-ci se trouvait pendant les mesures, et ne tient donc pas compte de la plasticité très variables des protéines.

Quelle que soit la méthode d'observation utilisée, le résultat obtenu est stocké dans un fichier standardisé : le format PDB pour *Protein Data Bank* [12], dont un extrait est fourni en annexe B. Ce format standardisé permet l'échange des résultats et constitue les données en entrée des logiciels de visualisation de structures moléculaires, dont quelques exemples seront présentés dans la section suivante.

1.2.1 *RasMol*, *YASARA*, et Cie : visualisation 3D de structures protéiques

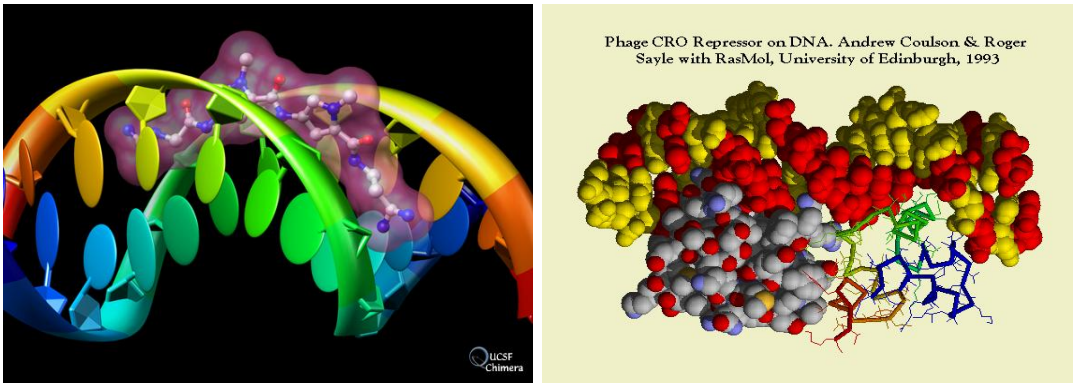
Pour visualiser les objets moléculaires tridimensionnel, comme les structures protéiques, de nombreux outils ont été développés. Dans l'état de l'art de Carlos Luiz

N. dos Santos, réalisé en 2005 [41], pas moins de 80 logiciels dédiés à la visualisation, à l'exploration, et à l'interaction avec ce type de molécules sont répertoriés.

Parmi les plus utilisés, *PyMOL* est un logiciel libre, auquel les utilisateurs peuvent ajouter des scripts en *Python* pour modifier assez profondément son interface, développer facilement de nouveaux plugins, afin de répondre à des besoins très spécifiques, sans changer ou redévelopper d'interface.

Développé plus récemment, *YASARA* est sans doute un des outils les plus performants d'un point de vue graphique, en partie grâce à des optimisations pour chaque type d'architecture matérielle, mais aussi grâce à un travail important de gestion de scène, orienté en particulier vers la visualisation immersive. Plus qu'un simple outil de visualisation, *YASARA* intègre aussi des outils de modélisation, de simulation de la dynamique moléculaire, ou de prédiction de structure 3D.

D'autres outils existent, tels que *Chimera* ou *Rasmol*, qui permettent d'obtenir de visualiser la structure tridimensionnelle de protéines, de complexes protéiques, ou de complexes ADN/protéine, à partir de fichiers *PDB*, comme illustré dans la figure 1.5.



(a) Fixation d'une mini-protéine sur l'ADN avec *Chimera*

(b) Interaction entre une protéine de répression et l'ADN Avec *RasMol*

FIG. 1.5 – Visualisation de biomolécules

1.2.2 *ADN-Viewer* : exploration immersive de structures nucléiques massives

Concernant les séquences d'ADN, les deux méthodes expérimentales, RMN, ou cristallographie par rayon X, révélant la structure tridimensionnelle de complexe moléculaire, permettent de ne traiter que les petites séquences de quelques centaines de paires de bases, car les longues séquences sont difficilement cristallisables. Parce qu'il est très difficile d'observer *in vitro* et encore plus *in vivo* de l'ADN, des biologistes et biophysiciens ont établi, par des méthodes statistiques et sur de petits fragments d'ADN (quelques centaines de nucléotides), des modèles de conformation spatiale d'ADN nu. Cela signifie qu'à partir de n'importe quelle séquence textuelle d'ADN, il est possible, grâce à ces modèles, de construire une trajectoire tridimensionnelle qu'adopterait la séquence si elle se trouvait hors de son environnement du fait de contraintes biophysiques.

Deux paires successives de bases complémentaires (appelé aussi plateau) de la double hélice que constitue l'ADN, s'orientent dans l'espace l'une par rapport l'autre selon 3 angles de rotations : *tilt*, *roll* et *twist*. En 1981, des mesures expérimentales (électrophorèse, cristallographie...) des angles *twist* entre les bases A-T en poly(dA) · poly(dT) [84, 103, 86] et les bases G-C en poly(dG) · poly(dC) [84] ont été effectuées. Ces mesures indiquent fortement que les angles *twist* ne sont pas identiques pour différentes combinaisons de paires de bases au sein de la molécule d'ADN. C'est en 1982 que Kabsch et al. fournissent le premier modèle générique des angles *twist* de l'ADN-B [70]. Plus tard, en 1991, les valeurs de *tilt* et *roll* de Bolshoy et al. [31] viennent compléter le modèle existant. Le modèle fourni en 1993 par Bolshoy et al. [96] est une version épurée et complète des précédents modèles. C'est un modèle di-nucléotidique, c'est-à-dire que la position et l'orientation d'un plateau ne sont influencées que par le plateau précédent : il s'agit donc d'un modèle local, illustré par la figure 1.6. Ces déviations successives entre deux plateaux permet de construire la trajectoire tridimensionnelle de la séquence, d'où découle la notion la courbure intrinsèque pour cette séquence.

La construction de la trajectoire tridimensionnelle de l'ADN, nécessite donc deux entrées : d'une part, la séquence textuelle d'ADN, et d'autre part, le modèle de conformation utilisé contenant les valeurs d'angles et de rotation pour chaque dinucléotides. A partir de ces deux entrées, un algorithme construit la trajectoire tridimensionnelle par empilement de chaque plateau en respectant les valeurs d'angles et de translation du modèle dinucléotidiques.

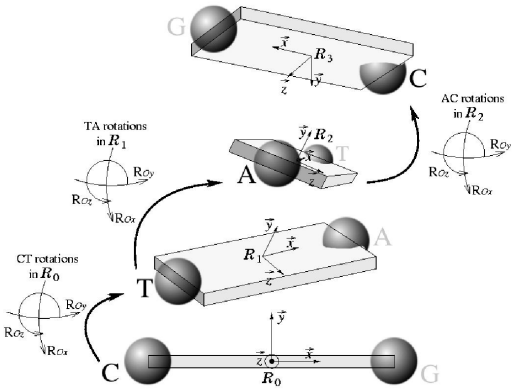
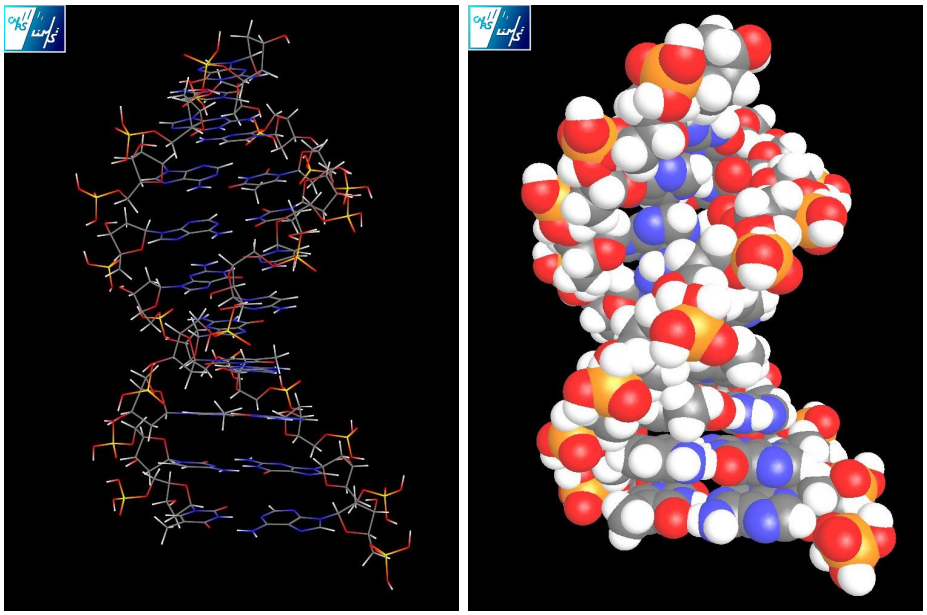


FIG. 1.6 – Empilement des plateaux selon le modèle dinucléotidique de Bolshoy et al.

ADN-viewer [64] est un logiciel développé par Joan Hérisson durant sa thèse, et permet à partir d'une séquence textuelle l'ADN, et d'un modèle dinucléotidique précédemment décrit, de visualiser la trajectoire de l'ADN dans l'espace. La quantité de données traitées, jusqu'à plusieurs millions de bases, a évidemment fait l'objet d'un travail de gestion de scène important[64], pour pouvoir visualiser ces objets complexes en temps interactif, à ces trois échelles, autant dans un cadre usuel de bureau, que dans un cadre immersif. L'optimisation qui permet la visualisation de structure volumineuse en respectant les contraintes précédemment énoncé, est un filtre dynamique, qui diminue ou augmente le niveau de détail de représentation graphique de la séquence, en fonction de la position de l'observateur. La visualisation de cette séquence peut s'envisager à plusieurs échelles : l'échelle atomique, la plus fine, permet d'observer les atomes de la structure en double hélice formée par quelques nucléotides, l'échelle génique, qui permet de visualiser la trajectoire de quelques gènes, et enfin l'échelle chromosomique, la plus macroscopique, permet de visualiser la structure d'un chromosome entier.

L'échelle atomique (voir figure 1.7) permet de visualiser localement et en détail de petite portion de l'ADN, dans le contexte global que constitue la trajectoire tridimensionnelle de la séquence entière.

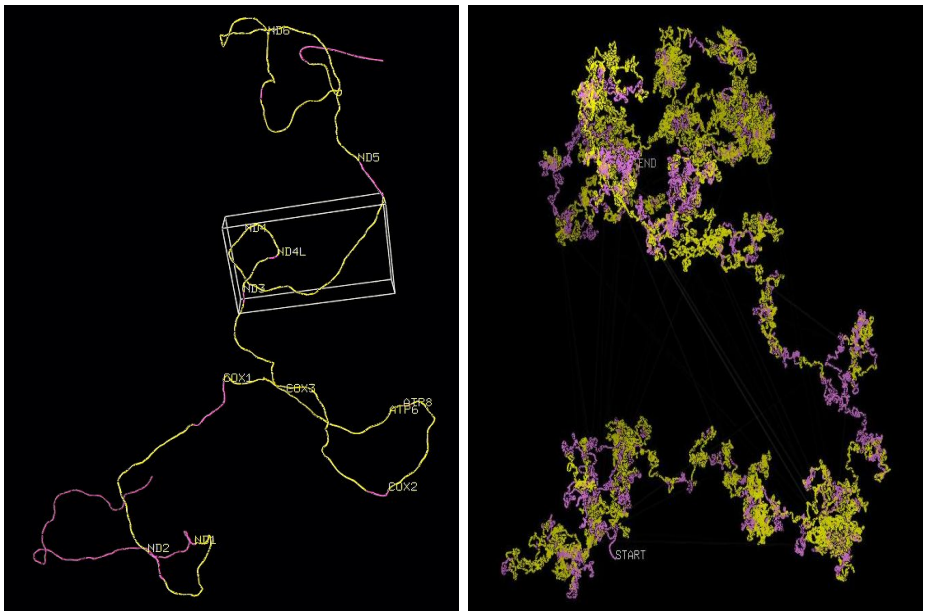


(a) Représentation filaire (b) Représentation atomique

FIG. 1.7 – Représentations d'une trajectoire d'ADN de 10 bp à l'échelle nucléique avec *ADN-viewer*[64]

L'échelle génique illustrée par la figure 1.8(a) permet d'observer la trajectoire à l'échelle du gène. En effet, si deux séquences d'ADN identiques, aboutissent évidemment à une trajectoire identique, deux séquences d'ADN différentes peuvent donner des trajectoires similaires. Il est donc particulièrement intéressant de connaître l'impact de l'architecture spatiale d'un gène sur un phénomène biologique donné. D'autre part, la visualisation de la trajectoire d'autres séquences, qui ne correspondent pas forcément des portions codantes, comme certaine zones intergéniques ou des transposons, aboutit parfois à des hypothèses biologiques intéressantes, qu'il convient de valider en paillasse, mais qui n'aurait pas émergée sans cette représentation.

Enfin, la trajectoire de l'ADN à l'échelle chromosomique présentée dans la figure 1.8(b) est importante car certains phénomènes biologiques sont liées à la fois aux propriétés structurales et à la conformation spatiale des chromosomes. Les niveaux d'organisation supérieurs comme les boucles et axes de la chromatine, les connections inter-chromosomiques pourraient être importants pour la morphogénèse du chromosome et joueraient également un rôle dans l'expression des gènes et la recombinaison [58]. Les activités comme la transcription et la synchronisation de la réplication sont liées à la disposition spatiale globale de différentes régions du noyau ainsi que leur relation avec l'enveloppe de ce dernier [39, 53, 37]. A chacun de ces niveaux, l'organisation du chromosome est très dynamique, variant à la fois au cours du cycle de la cellule et selon les différents types de cellule.



(a) Echelle génique (b) Echelle chromosomique

FIG. 1.8 – Représentation d'une trajectoire d'ADN avec ADN-Viewver

ADN-viewer a été développé dans un cadre immersif. Ce cadre sera décrit précisément dans le chapitre 2, car les modalités de visualisation et d'interactions avec les données lui sont très spécifiques. Ce travail a montré que le cadre immersif est particulièrement adapté à l'exploration et l'interaction avec des données génomiques massives.

1.3 Les données d'expression

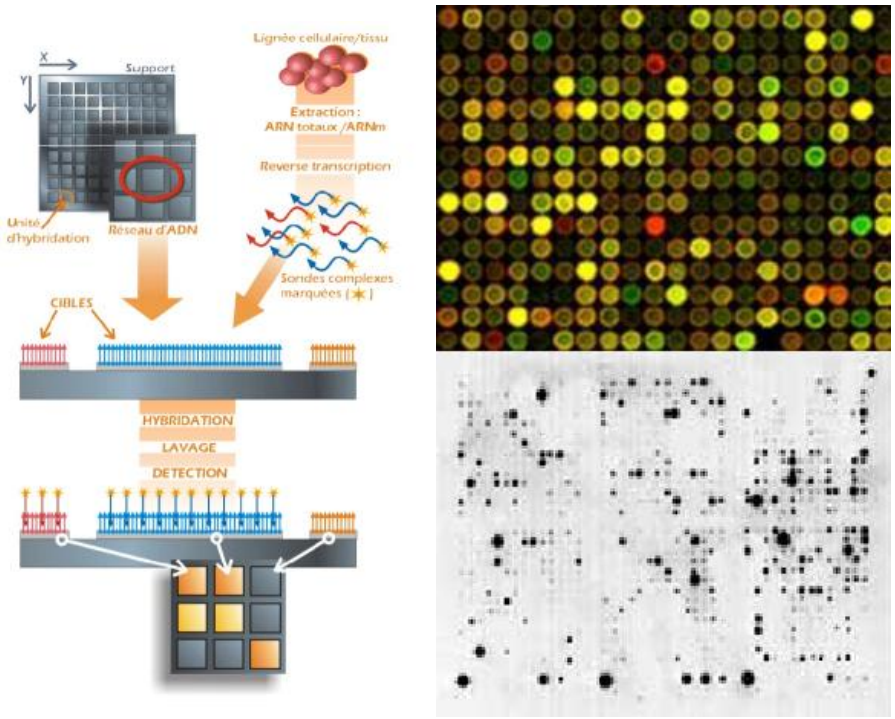
Les outils présentés dans les sections précédentes, se focalisent sur l'analyse des génomes en tant que séquences, et permettent de visualiser la structure, linéaire, ou tridimentionnelle des gènes dans leur environnement, et de leurs produits, les protéines.

Une autre grande classe de données concernant les génomes, issue de la technologie des puces à ADN ou bio-puces, sont les données d'expression. Cette technologie, permet de mesurer en le niveau d'expression *i.e*, l'activité, de grand ensembles de gènes, voire d'un génome entier, dans des dizaines de conditions expérimentales différentes, physiologiques ou pathologiques.

La technologie des puces à ADN ou bio-puces, a connu un essor très important et suscite un fort intérêt dans la communauté scientifique. Son utilité provient du fait que la connaissance du niveau d'expression d'un gène dans différentes situations constitue une avancée vers sa fonction, et permet donc d'aller vers le criblage de nouvelles molécules, aboutissant à la conception de nouveaux médicaments et de nouveaux outils de diagnostic. Par ailleurs, cette méthode permet d'étudier l'activité de grands ensembles de gènes simultanément, et constitue donc une avancée vers l'étude du génome à un niveau global, en mettant en lumière les groupes des gènes dont le niveau d'expression varie de façon identique selon les conditions expérimentales qui leurs sont imposées.

D'autre part, cette méthode d'analyse du vivant fait appel à des expertises très variées. Les puces à ADN sont constituées par un support solide (petite lame de verre comme celles utilisées en microscopie traditionnelle ou membrane de nylon) sur lequel des milliers de fragments d'ADN simple brin sont déposés de façon géométrique à l'aide d'une micropipette robotisée. Grâce à cette technique, chacun des fragments d'ADN est représenté par un point sur le support (ou puce). Ils servent de sondes pour fixer de façon très spécifique les fragments d'ADN complémentaires simple brin (cibles), présents dans les échantillons biologiques à tester : leur mise en contact permet de reconstituer la double hélice d'ADN. Ce phénomène (hybridation) peut être mis en évidence par des techniques optiques sous éclairage fluorescent ou par détection de radioactivité ; un système de « marquage » de l'échantillon au moyen de traceurs fluorescents ou radioactifs ayant été réalisé au préalable (voir figure 1.9).

La quantification des signaux obtenus et l'identification des fragments de gènes reconnus sont ensuite rendues possibles au moyen d'un système d'acquisition d'images dont un exemple de résultat est présenté dans la figure 1.9.



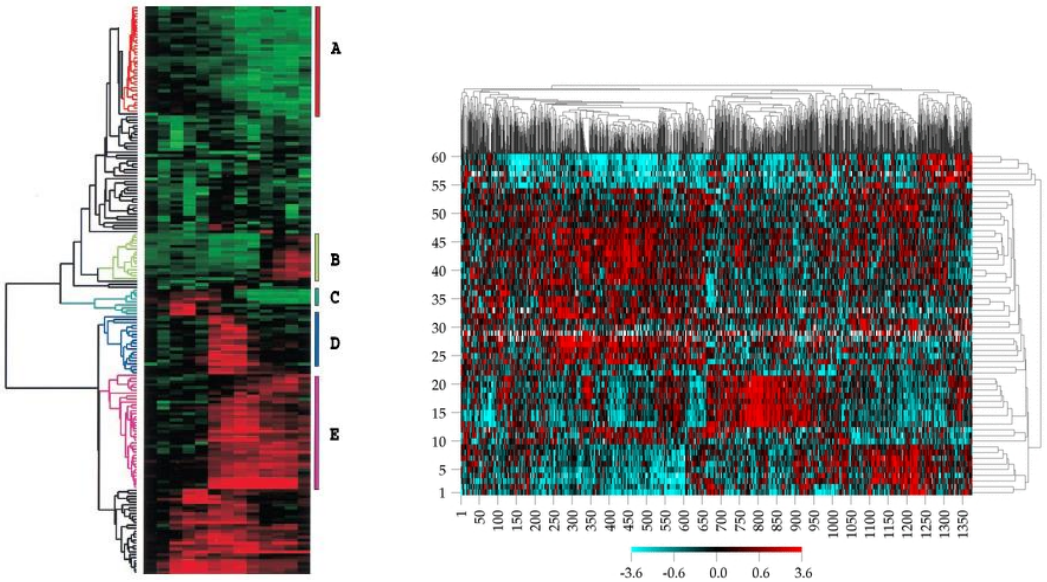
(a) Principe de fonctionnement

(b) Résultat d'un marquage florescent (en haut) ou radioactif (en bas)

FIG. 1.9 – Les puces à ADN

L'analyse des données est ensuite effectuée à l'aide de logiciels informatiques spécialement conçus à cet effet. Les résultats obtenus sont ensuite validés sur le plan statistique et interprétés dans un contexte biologique spécifique. La principale méthode d'analyse utilisée consiste à regrouper les gènes par clusterisation hiérarchique et à visualiser ces résultats par des dendogrammes [45], comme montré dans la figure 1.10.

Cette méthode pose cependant des problèmes de lisibilité pour des données toujours plus volumineuses. En effet, la taille des puces à ADN a considérablement augmenté depuis quelques années, et atteint désormais plusieurs dizaines de milliers de gènes par puces, et la visualisation par clusterisation hiérarchique rend l'analyse difficile, comme illustré dans la figure 1.10. Cependant cette problématique a été adressée par Munzner et al. qui ont mis en oeuvre dans *TreeJuxtaposer* [82] une solution originale de navigation Focus+Context pour garantir la lisibilité lors de la visualisation d'arbre de grande taille.



(a) Moins d'une centaine de gènes

(b) Quelques milliers de gènes

FIG. 1.10 – Analyse de données d'expression par clusterisation hiérarchique

D'autre part, la méthode d'analyse par clusterisation hiérarchique ne permet d'affecter un gène qu'à un seul groupe, ce qui exclu toute possibilité d'analyse d'un éventuel rôle multiple de son produit dans la cellule. C'est la raison pour laquelle certains travaux ont proposé d'autres méthodes de clusterisation plus souples [52], qui permettent de refléter et de prendre en compte cette réalité biologique. D'autres méthodes plus récentes, comme l'ensemble d'outils proposés par *GeneVestigator* [114], permettent de faire des méta-analyses de données d'expression en explorant ces données selon plusieurs points de vue. Par exemple, *Response Viewer* permet de visualiser la sensibilité d'un gène donné à une condition, ou bien *GeneChronologer* offre de suivre l'expression d'un gène pendant différentes étapes de la morphogénèse.

1.4 Les données textuelles

Une grande quantité de connaissances concernant les génomes se trouvent aussi dans les publications (données textuelles), desquelles ont d'ailleurs été extraites manuellement nombre de données factuelles qui ont permis de créer les premières banques de données génomiques factuelles. Ces données textuelles sont stockées sous une forme qualifiée de non structurée d'un point de vue informatique, car elle sont exprimées en langage naturel. Faire le lien entre ces données bibliographiques textuelles et les banques de données factuelles, constitue un enjeu important. D'une part, cela permet d'enrichir ces bases de données factuelles de connaissances provenant de données textuelles, et d'autre part, cela facilite le processus de recoupement entre les bases/banques de données génomiques, et les publications, qui constitue le coeur de l'activité du chercheur en biologie dans la phase de recherche bibliographique.

1.4.1 *BioBibliometrics* : navigation dans la bibliographie des génomes

Dans ce cadre, B.J.Staplet et G.Benoit [100] proposent un prototype de visualisation d'informations biologiques textuelles et factuelles (voir figure 1.11). Les gènes sont extraits des documents *PubMed* [13] à partir de termes appartenant au thesaurus *MeSH* [17]. L'article propose d'utiliser le coefficient de co-occurrence de *Dice* comme mesure de proximité bibliographique entre deux termes ou alias désignant les gènes. Ces informations bibliographiques sur ces gènes sont représentées par un graphe. Chaque gène est représenté par un noeud, la relation bibliographique minimale (seuil de co-occurrence défini par l'utilisateur) entre deux gènes est représentée par un arête. Enfin, le degré de proximité entre chaque paire de gènes est représentée par la longueur de ces arcs. Les co-occurrences de ces gènes trouvées automatiquement par le système dans la base textuelles servent à les renforcer et à les affaiblir.

Cette exploration simple et élégante, ne saurait s'appliquer à toutes les publications biologiques, qui ne serait de toute façon pas pertinente. Elle doit être associée à un domaine spécifique, choisi par l'utilisateur en amont, par un terme désignant une thématique précise de recherche (exemple : DNA repair). Cette spécialisation de l'exploration, associée à la possibilité d'effectuer un seuillage statique préalable sur la co-occurrence, permets d'extraire de petits graphes représentant une synthèse bibliographique organisée par les co-occurrences des gènes dans la bibliographie. Enfin, une fois cette synthèse réalisée, *BioBibliometrics* permet la récupération des articles en renvoyant sur la base *PubMed* [13].

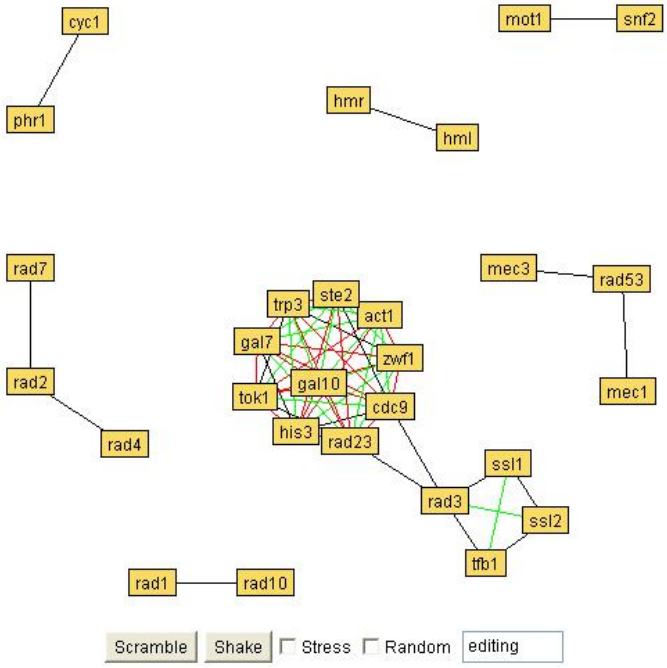


FIG. 1.11 – *BioBiblioMetrics* : Réponse à la requête "DNA repair and DNA Transcription"

1.5 Les banques de données factuelles

Les bases de données commencent à être considérées comme des objet d'étude en tant que tel, dans la mesure où les croisements de plusieurs bases donnent accès à des informations de type différent. Donner la possibilité de faire des recoupements entre les bases, permet de faire des découvertes ou de poser des hypothèses de nature biologique. Par exemple, le recoupement des données concernant les activités enzymatiques, avec les données de séquence, ont permis d'évaluer le pourcentage d'activités enzymatique répertorié, mais sans séquences associées (que les auteurs ont appelé "enzyme orpheline"), à environ 40 % [78]. Ce type de recherche systématique est cependant long et difficile, compte-tenu de la quantité de données à traiter, et de la difficulté d'intégration de format et de nomenclature assez hétérogène. Représenter ces données de façon visuelle, permettrait, je pense, de faciliter l'analyse ce ces bases, et serait une aide précieuse à ce travail souvent collaboratif. Quelques outils de visualisation de bases de données sont donc présentés dans cette section pour répondre à cette problématique.

1.5.1 *GenoMEDIA* et *Lysa* : intégration et visualisation collaboratives de banques de données factuelles

Le problème de l'intégration des bases/banques de données génomiques, est un problème en soi. Je donne ici un exemple d'outil d'intégration transparente et de visualisation de données génomiques proposé par Pierre-Emmanuel GROS, durant son travail de thèse, en présentant le middleware *GenoMEDIA* [57] pour l'aide à l'intégration de données génomiques et de services bioinformatiques. Ce middleware permet d'effectuer des requêtes en *SQL*, sur plusieurs bases, de façon transparente pour l'utilisateur. Ces requêtes peuvent intégrer des services génomiques, permettant par exemple d'utiliser l'outil *BLAST* [21] au sein d'une requête *SQL*, ou bien n'importe quel autre outil choisi ou codé par l'utilisateur.

Outre cette capacité d'intégration des données et des services, grâce à une approche "entrepôt de données" ou "médiateur" selon le contexte, ce middleware propose aussi un module de visualisation et d'exploration collaboratif du résultat d'une requête, *Lysa*. Chaque tuple résultat d'une requête est projeté sur une sphère transparente, et le positionnement relatif de chaque tuple (DataBall), est contraint par une distance (figure 1.12), reflet d'une métriques entre deux tuples. Cette distance peut aussi être établie de façon arbitraire par l'utilisateur afin d'organiser et de trier ses données manuellement. Un des points forts de cet outil est son aspect collaboratif. En effet, plusieurs utilisateurs peuvent à distance et simultanément effectuer cette tâche d'organisation. D'autre part, cet outil propose à l'utilisateur, de façon automatique, une liste d'outils externes appropriés (*ADN-Viewer*, *RasMol*, ...) aux données contenues dans un tuple (séquence nucléique, séquence protéique, ...) afin de visualiser plus en détail son contenu.

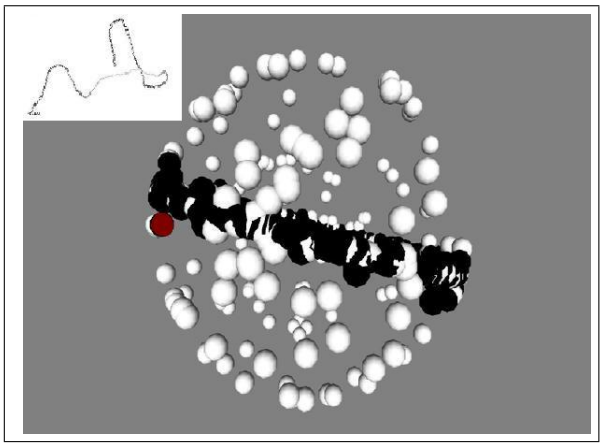


FIG. 1.12 – Visualisation du résultat d'une requête avec *Lysa* et de la séquence sélectionnée avec *ADN-viewer* en haut à gauche

1.5.2 *Sequence World* : exploration tridimentionnelle de banques de données génomiques

Une autre approche a été empruntée par le logiciel *Sequence World* [87], un des premiers travaux qui traite de l'exploration de bases de données génomiques dans un contexte tridimentionnelle, est de représenter les gènes d'une base de données dans un paysage, que l'acteur explore en restant sur le sol, pour utiliser les capacités d'orientation de l'être humain. Comme illustrée dans la figure *sequenceworld*, les gènes peuvent être regroupés géographiquement selon plusieurs critères biologiques, comme la similarité de séquence, en utilisant par exemple l'algorithme *BLAST* [21] pour calculer un score d'alignement traduit en distance entre les objets du monde virtuel.

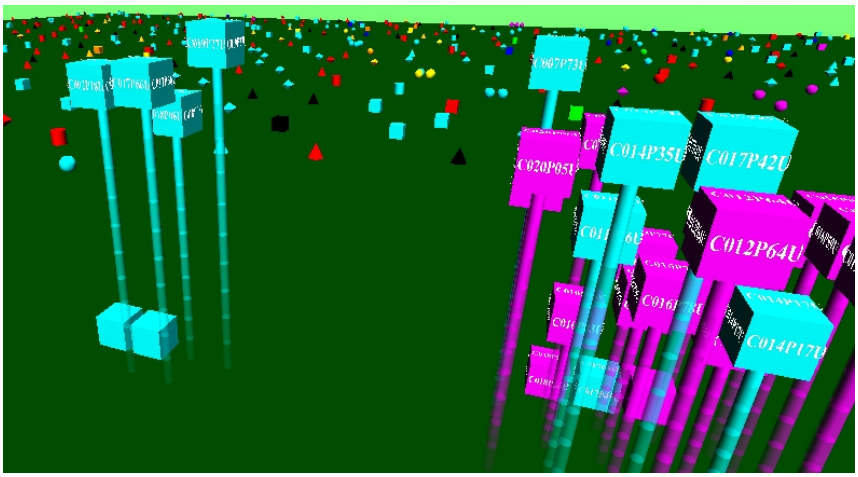
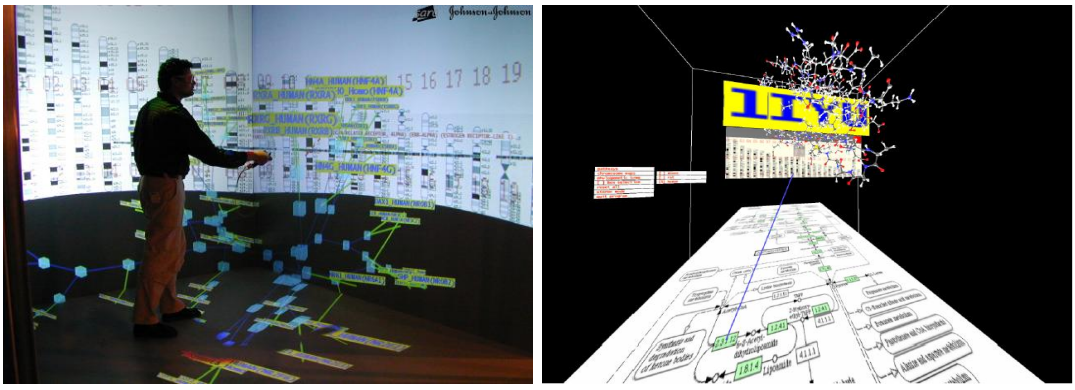


FIG. 1.13 – *Sequence World* : Exploration tridimentionnelle d'une banques de données génomiques en Réalité Virtuelle

1.5.3 *SARAGene* : exploration immersive de banques de données génomiques

Un travail plus récent et plus abouti a été réalisé par l'équipe de Bram Stolk en 2002. Le but de ce travail est d'explorer des bases de données génomiques dans un cadre immersif. L'application nommé *SARAGene* [102], permet d'explorer de manière simultanée, des données d'alignement, de phylogénie, de métabolisme (*KEGG*) [8], de réseau de régulation, des données moléculaires (*PDB* [12]) en restant dans le même environnement. Cette approche intégrative de différents paradigmes de visualisation (figure 1.14) pour l'exploration de données génomiques, permet de créer rapidement des ponts entre des données génomiques de nature variées, ce qui facilite la compréhension du fonctionnement d'un processus biologique.



(a) Exploration d'un réseau d'alignement de séquences d'ADN chez l'homme

(b) Structure moléculaire de l'enzyme EC 2.3.1.12 (voie métabolique de la glycolyse, entrée PDB : 1IYU). La ligne bleue renvoie à sa localisation dans le réseau métabolique

FIG. 1.14 – *SARAGene* : un système de visualisation de données génomiques dans un contexte immersif

Il y a plusieurs leçons à tirer de la conception de solution d'exploration de banques de données dans un cadre immersif. La visualisation tridimentionnelle dans un cadre de réalité virtuelle, peut exhiber de nouvelles connaissances, latentes dans les données et difficiles à extraire sans représentation tridimentionnelle de ces données qui ne sont pas, par nature, tridimensionnelles. Il est primordiale pour les biologiste de représenter visuellement le degré de crédibilité des faits biologiques, en particulier si plusieurs sources de données sont utilisées. En effet, un degré de crédibilité des données factuelles est très variable, et dépend du type d'expérience qui ont fourni ces résultats. Par ailleurs, de nombreux résultats proviennent d'outils bioinformatiques de prédiction, et ne sont pas vérifiés expérimentalement.

Utiliser des représentations avec lesquelles les biologistes ont l'habitude de travailler constitue un point positif, cependant, transposer directement des représentation visuelle bidimentionnelle développés pour un cadre de bureau, à un contexte immersif, fait perdre les avantages de ce contexte. En particulier, certains retours d'information, comme le montre la figure 1.14, sont le résultat d'outils externes de visualisation 2D, et sont importés dans la scène par texturage. Le passage d'une représentation bidimensionnelle à une tridimensionnelle diminue l'impression d'immersion perçue par l'utilisateur.

1.6 Conclusion

Nous avons vu dans ce chapitre une présentation d'outils d'exploration de données génomiques variées. Ces outils ont individuellement répondu certaines problématiques, associées à l'exploration immersive de données génomiques textuelles ou factuelles. En particulier, la problématique associée à la visualisation de données massives est particulièrement bien traitée dans *ADN-viewer*, ou plusieurs millions de paires de bases, un chromosome entier, peuvent être visualisées en temps interactif. Par ailleurs, le problème de l'hétérogénéité des bases/banques de données est assez difficile à résoudre, et *GenoMEDIA* est un bon exemple d'intégration de données hétérogènes au sein d'une plate-forme transparente pour l'utilisateur. Enfin, la conception de solution comme *ADN-viewer* ou *SaraGENE*, a permis de démontrer que le cadre immersif permet de visualiser des données génomiques massives, mais a fait aussi émerger des problématiques nouvelles associées au contexte immersif, comme celle de l'utilisation de la multimodalité qui consiste à utiliser d'autres canaux que le canal visuel pour interagir avec son objet d'étude.

Chapitre 2

Données génomiques : de leur caractérisation au contexte de leur exploration

Le double cadre de ce travail, dont l'objet d'étude est l'exploration de données génomiques, ceci dans un contexte immersif, sera décrit dans ce second chapitre, pour, d'une part, caractériser les spécificités inhérentes aux données génomiques, et d'autre part, décrire plus précisément les techniques et les dispositifs utilisés dans le contexte immersif afin d'offrir à l'utilisateur l'illusion d'immersion.

2.1 Caractérisation des connaissances génomiques

Trop de connaissance ne facilite pas les plus simples décisions

F. Herbert

Avant les années 1980, la plupart des connaissances concernant le génome, étaient stockées et échangées grâce aux publications. Cependant, la quantité de données produite par l'essor du séquençage, importante pour l'époque qui a suivi, a rendu ce média peu pratique pour la maintenance, la diffusion, et le partage de ce nouveau type de connaissances.

C'est au début des années 80, qu'une des premières bases de données est utilisée pour faciliter la maintenance et la vérification de données de séquences [72]. C'est en 1985 qu'apparaît *GenBank* [2], ou 6000 séquences, au total 5 millions de paires de bases, sont extraites de 4500 publications, pour être distribuées sur disquette.

2.1.1 Une croissance exponentielle de contenu

Après la création de *GenBank*, l'émergence de nouvelles bases fut ensuite extrêmement rapide, associée au développement d'Internet, et à la progression surprenante de la technologie du séquençage. Ceci a aboutit à une croissance exponentielle de la quantité d'information à stocker, à organiser, puis à analyser. Ce volume grandissant nécessite un travail de recherche important pour rendre plus efficaces les algorithmes automatiques d'analyse des séquences à grande échelle, et pour automatiser l'annotation manuelle de plus en plus longue et fastidieuse.

2.1.2 Une hétérogénéité selon de multiples points de vue

Comme nous l'avons vu dans la sélection d'outils présentée dans le chapitre précédent, les données génomiques sont loin de se limiter au résultat du séquençage systématique réalisé durant la dernière décennie. La structure et le contenu de ces bases de données répondent à des besoins d'organisation en perpétuelle mutation, ce qui a aboutit à une grande hétérogénéité en terme de contenu, d'organisation, de modalités de représentation et de stockage de l'information, ou de qualité.

2.1.2.1 Une hétérogénéité de qualité

Face à cet afflux massif de données à traiter, des politiques de stockage contradictoires de ces connaissances eurent pour conséquence une très importante disparité de qualité et de format de ces banques de données. Les utilisateurs accordent ainsi à certaines bases de données génomiques une très forte crédibilité, comme *Swissprot* [15], car toutes les données proposées au stockage sont systématiquement contrôlées par des experts du domaine. D'autres bases, comme *GenBank* [2], sont plus laxistes dans leur processus de sélection, et sont donc considérées comme moins crédibles.

Paradoxalement, ces bases connaissent un succès identique, dans la mesure où les premières contiennent peu d'informations mais avec une très grande crédibilité, alors que les secondes contiennent des données d'une qualité très variable, mais en quantité très importante. Comme le montre la figure 2.1, la croissance de toutes ces bases est très variable selon la politique de gestion de ces connaissances.

2.1.2.2 Une hétérogénéité de contenu

Le séquençage ayant été ensuite réalisé organisme par organisme, les données génomiques ont logiquement été regroupées par organisme, comme pour la base *SGD* (Saccharomyces Genome Database [14]) ou *MGI* (Mouse Genome Informatics [14]). Parallèlement, d'autres bases, dites intégratives, comme *GenBank* [2] ont répondu à un besoin de regrouper plusieurs organismes, pour faciliter les analyses comparatives sur génomes pour l'étude de la phylogénie des espèces par exemple. Répondant à

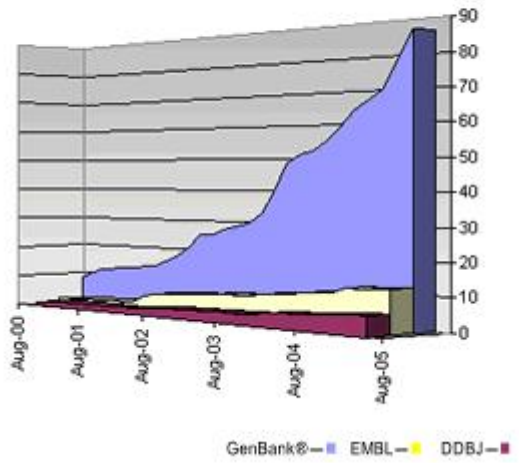


FIG. 2.1 – Croissance en millions de paires de bases de 3 banques de données génomiques

d'autres logiques d'organisation, certaines bases regroupent des objets biologiques de plus en plus spécifiques. Ainsi, *SwissProt* [15] est spécialisé dans le regroupement des connaissances concernant les séquences protéiques et leurs caractéristiques (fonction, localisation cellulaire, organisme), et la base *RCSB PDB* [12] donne accès aux données de structure tridimentionnelle des protéines. Plus spécialisée encore, on peut citer *European Ribosomal RNA Database* [1] qui rend publiques des données associées à la structure des ARN ribosomiaux, ou *IMGT* (ImMunoGeneTics) [7], dont le rôle est de regrouper les données génomiques associées aux systèmes immunitaires d'organismes variés.

Malgré cette disparité en terme de contenu et d'organisation, l'objectif commun de toutes ces bases est de faciliter la compréhension de mécanismes biologiques, à toutes les échelles de l'organisme. Je propose dans la figure 2.2, une organisation des connaissances génomiques factuelles en quatre classes, les données concernant le génome, le transcriptome, le protéome, le métabolome.

L'objectif de l'étude des données concernant le génome est d'analyser les données issues du séquençage systématique de l'ADN d'organismes variés. Le programme génétique que porte l'ADN, polymère constitué d'une chaîne de quatre monomères de bases appelés nucléotides, l'adénine, la cytosine, la guanine, la thymine (A, C, G et T), est propre à chaque espèce, et identique dans toutes les cellules constitutives d'un organisme. Le génome contient plusieurs niveaux d'organisation. Il peut être étudié d'un point de vue linéaire, par l'analyse de la séquence sous sa forme textuelle. Il peut être étudié de manière plus globale, avec ses différents niveaux d'organisation, en

particulier son architecture spatiale, de la simple molécule d'ADN au chromosome.

Un processus biologique important dans la réalisation du plan orchestré par le programme génétique, est le phénomène de transcription (*cf.* annexe A). Ce phénomène permet diffuser dans la cellule, les ordres issus du programme génétique. Cette diffusion, s'effectue par la traduction de la portion d'ADN, en molécule d'ARN. D'une part, l'objectif de l'étude des données du transcriptome, est de caractériser les molécules d'ARN, plus complexes que les molécules d'ADN, à travers de l'analyse de leur structure primaire (séquence textuelle), secondaire (repliement dans le plan), ou tertiaire (repliement dans l'espace). D'autre part, et à une échelle plus globale, les données d'expression permettent de mieux comprendre comment les gènes s'expriment et agissent de concert dans un organisme en fonction de l'environnement.

Un autre phénomène important est la traduction. Une partie des ARN transcrits, sont traduits en protéines, selon un code génétique, universel à tous les organismes, à quelques exceptions près (*cf.* annexe A). Les protéines sont à la fois les briques et les ouvriers (appelés enzymes) qui oeuvrent à la construction cellulaire. De même que pour l'ARN, les protéines possèdent une structure primaire, une séquence d'acides aminés, une structure secondaire, et une structure tertiaire. L'étude des données associées au protéome, en particulier de la structure tertiaire des protéines (repliement tridimensionnel) est très importante, puisqu'elle constitue un indice précieux pour comprendre la fonction de la protéine dans un organisme.

Enfin, l'étude des données associées au métabolome se consacre aux réactions biochimiques qui permettent, par l'intermédiaire, entre autre, des protéines enzymatiques (enzymes), de synthétiser et de transformer les briques de bases qui permettent à un être vivant de se construire et d'assurer les fonctions de bases d'un organisme, de l'échelle cellulaire jusqu'à l'organisme dans son ensemble.

Associées à ces données factuelles structurées dans des banques de données génomiques, dont les plus institutionnelles ont été indiquées en italique dans la figure 2.2, les données bibliographiques, de nature textuelles, ont aussi été regroupées en banques de données, comme PubMed [13]. De nombreux travaux sur ces données textuelles, pour faciliter le traitements automatiques de ces documents, a abouti à la production, de nombreuses terminologies et ontologies spécifiques à l'étude des génome, comme *GeneOntology* [3]. Ces données sont le résultat de consensus longuement débattu, en perpétuel mutation, entre les experts du domaine.

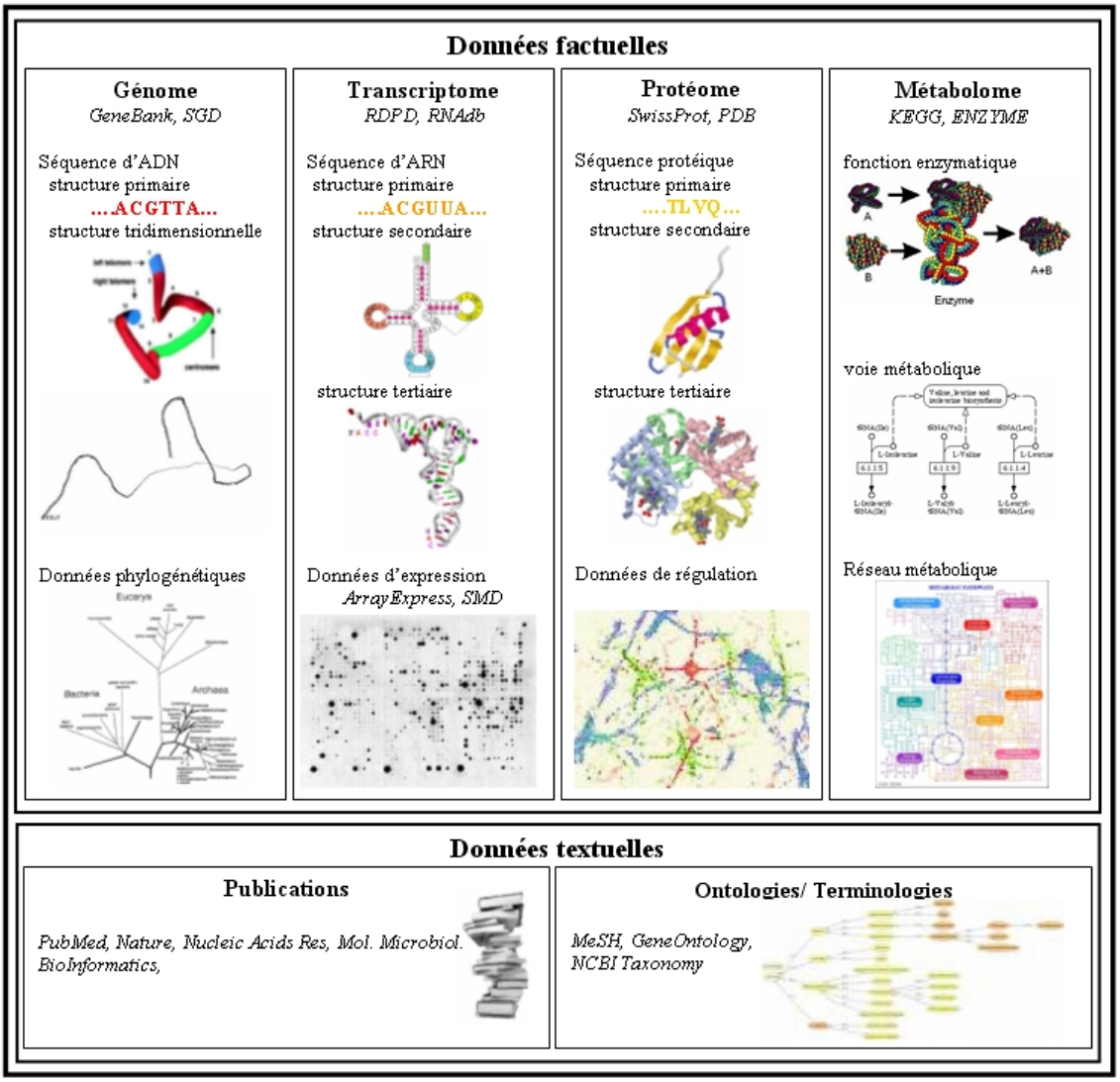


FIG. 2.2 – Vue d'ensemble de l'organisation des banques de données génomiques

2.1.2.3 Une hétérogénéité technique

Outre cette hétérogénéité de contenu, la quantité toujours plus importante de données à stocker, a aboutit à la fois à une incohérence et à une redondance dans la nomenclature et l'indexation des objets biologique d'intérêt. Des séquences identiques sont référencées de manière différente selon les bases, aboutissant à des redondances difficiles à traiter, comme le problème de l'identification des références croisées. Ces besoins de structuration et d'organisation de ces bases de données, émis par la communauté des chercheurs en biologie, aboutirent à une problématique biologique et informatique difficile : comment inter-connecter ou intégrer ces bases de données, de contenus variés, ainsi que de nomenclature et de qualité différente. Enfin, les bases de données génomiques font preuve d'une disparité importante dans le formatage de leur données, avec des formats et des technologies de gestion de bases de données différentes.

2.1.3 Des points communs propres aux données génomiques

2.1.3.1 Organisation autour d'un seul type d'objet biologique

Cependant, la structure interne des bases de données génomiques est longtemps restée d'un point de vue informatique très simple, puisqu'il s'agit en général d'une liste d'objets, certes volumineuse, mais d'un seul type. C'est la raison pour laquelle les instances de chaque gène ont longtemps été stockées sous forme de fiche signalétique, dans des fichiers plats (flat file). Cependant, la nécessité d'intégrer des données de nature et de source différentes, de visualiser ces données pour faciliter l'analyse, fut à la source de profonds changements dans l'architecture de ces bases, en particulier dans les bases génériques, dont le rôle est finalement de créer des interfaces entre les données de natures et de sources différentes.

Les bases de données génomiques ont cependant conservé ce caractère "ancestrale", et restent fréquemment centrées sur un type d'objet biologique unique. Voici quelques exemples :

- la protéine, en tant que séquence pour *SwissPROT* [15], en tant que structure moléculaire, pour la *Protein Data Bank* [12].
- le gène, en tant que séquence. Ces bases peuvent être spécifiques à un organisme ou une famille d'organisme, comme *FLAGdb* [90], Saccharomyces Génome Database [14], Mouse Genome Informatics [10], ou sont génériques, comme *GenBank* [2]. Les bases de données générique ne sont en général que l'intégration du contenu des bases de données spécifiques.
- les anticorps, dans la base *IMGT* [7].
- les termes spécifiques au domaine, hiérarchisés dans *GeneOntology* [3].
- ...

2.1.3.2 Structuration en réseau de relations

En prenant du recul sur ces bases de données, on peut constater que les données génomiques, au niveau *"overview"* cher à Shneiderman [95], s'organisent toutes sous forme de réseaux. Au niveau du génome, les réseaux philo-génétiques, au niveau du transcriptome, les réseaux de régulation et de transcription, au niveau du métabolome, les réseaux métaboliques (cf. figure 2.2).

A une échelle plus locale, les instances d'objet qui constituent ces bases de données, sont très souvent comparées deux à deux, comparaisons qui constituent le coeur des analyses bioinformatiques : similarité de séquence, similarité fonctionnelle, similarité d'expression, similarité de localisation cellulaire, appartenance à une même voie métabolique. Les relations entre ces objets, peuvent être caractérisées par des valeurs quantitatives, comme le score de similarité entre deux séquences, ou bien qualitatives, comme la séquence consensus.

Ces données génomiques stockées de façon structurée, ne constituent cependant pas l'unique source de connaissances génomiques, et ne sont que la partie visible de l'iceberg. En effet, une partie importante des connaissances sur les génomes se situe dans les publications, et sont de nature textuelle. Ces masses de données sont difficiles à analyser de façon automatique, car les connaissances sont exprimées en langage naturel, dont la structuration est loin d'être celle utilisable directement par la machine. Ces masses de données ne peuvent être traitées manuellement en étant exhaustif, étant donnée la quantité importante de l'ordre du millions de publications disponibles concernant les génomes (Requête sur *PubMed* [13] : 1017149 entrées pour "gene", 881571 entrées pour "DNA" en octobre 2006).

En ce qui concerne les données bibliographiques, il se trouve que la plupart des outils automatiques d'extraction d'informations à partir de publications, identifient les relations entre les termes ou des concepts biologiques. Ces relations peuvent être simples, comme par exemple des co-occurrences entre deux termes au sein d'une même phrase, d'un même paragraphe, d'un article, ou d'un corpus [100]. Elles peuvent être plus complexes, de nature sémantique, comme l'extraction d'information a partir de texte, en réponse à des requêtes exprimées sous forme de patrons de concepts [22]. Appliqué à une problématique biologique, on peut, par exemple, utiliser ces systèmes, afin d'extraire tous les termes désignant des protéines [94], et en extraire toutes les interactions biologiques entre ces protéines qui apparaissent dans un corpus [105]. Dans tous les cas, ces systèmes construisent des réseaux de termes, ou de concepts, qui permettent de caractériser une publication ou un corpus, dans un langage accessible à un traitement informatique, et plus propice à la synthèse et à l'analyse.

2.1.3.3 Conclusion

Dans les travaux qui a ont été présentés dans le premier chapitre, on peut constater que, excepté pour quelques outils comme *SARAGene* [102], les outils de visualisation de données génomiques sont la plupart du temps développés pour répondre à une problématique biologique très précise, et concernant des données spécifiques. A chaque organisme, une nouvelle base de données, à chaque nouvelle fonctionnalité, une nouvelle application. Ce cloisonnement a eu pour conséquence une certaine difficulté de la part des utilisateurs à appréhender les données génomiques de façon globale et intégrée, empêchant les comparaisons entre les données séparées par cette frontière applicative.

On remarque aussi que les outils de visualisation associés aux bases de données génomiques se focalisent un niveau de détail très fin, ce qui empêche la prise de recul, qui pourrait pourtant aboutir à des questions biologiques importantes, comme ce fut le cas pour les enzymes orphelines [78]. La principale cause provient du fait que la plupart de ces outils d'exploration sont des *web services*. Ceci a certes permis de partager une quantité de données importante, mais ces outils restent limités en terme de visualisation de données, en particulier si ces dernières sont massives, puisque Internet a été conçu pour présenter l'information essentiellement de façon textuelle.

2.2 Contexte de réalité virtuelle

La fleur de l'illusion produit le fruit de la réalité

P. Claudel

Choisir le contexte de réalité virtuelle (aussi appelé contexte immersif) afin de visualiser des données, est une réponse partielle à la problématique associée à la représentation visuelle de données massives. En effet, certaines caractéristiques spécifiques comme la taille importante du champs de visualisation, et la stéréoscopie, qui donne accès à une dimension supplémentaire dans la présentation des données, permettent d'explorer une quantité plus grande de données [55]. D'autres raisons ont aussi conduit à choisir ce cadre, comme le fait que certaines données génomiques sont par nature tridimentionnelle, comme les données moléculaires qu'ils s'agissent de séquences d'ADN, d'ARN, ou de protéines. Or, passer d'un contexte tridimensionnel à un contexte de bureau, pose des problèmes, en terme d'interaction avec les données, car les modalités d'interaction sont différentes dans les deux cadres, en terme d'adaptation, entre les données représentées de façon tridimentionnelle, parfois en stéréoscopie, ou bidimensionnelle, sans stéréoscopie. Visualiser ces données dans un contexte unique, à savoir tridimensionnel et immersif, est donc un enjeu important. La suite de ce chapitre est destinée à décrire le cadre immersif qui entoure ce travail. En particulier, la présentation des dispositifs d'affichage et d'interaction, propres à ce contexte de réalité virtuelle immersive, est nécessaire pour comprendre les paradigmes de visualisation et d'interaction qui seront présentés dans le chapitre suivant.

2.2.1 La réalité virtuelle : définition

La notion de réalité virtuelle était implicitement esquissée par *Platon* dans son mythe de la caverne, ainsi que par *René Descartes* qui envisage, dans son "Discours de la méthode", l'hypothèse que les témoignages de ses sens pourraient n'être qu'une série d'illusions coordonnées par un esprit extérieur (malin voir démoniaque). *Wikipédia* donne une définition plus actuelle de réalité virtuelle : "La réalité virtuelle est une simulation informatique interactive et immersive d'environnements 3D réels ou imaginaires". Cette discipline est née de l'intégration de dispositif d'affichage de haute résolution, et des techniques de stéréoscopie qui permettent de donner à l'utilisateur l'illusion d'une troisième dimension. L'utilisateur a ainsi l'impression d'être immergé dans un monde artificiel construit numériquement. C'est l'aspect immersif de la réalité virtuelle. L'aspect interactif se traduit par l'utilisation de dispositifs

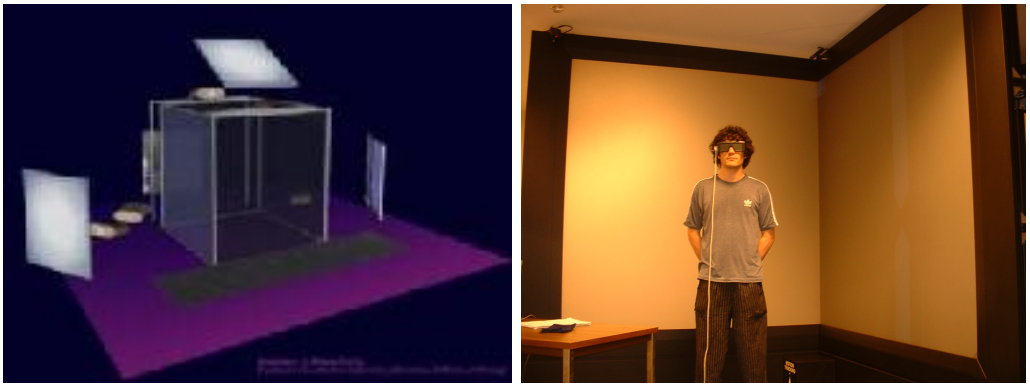
d'interaction, qui permettent à l'utilisateur de percevoir, de sentir, ou de modifier le monde artificiel dans lequel il est immergé.

2.2.2 Les techniques de simulation de l'immersion

2.2.2.1 Une surface d'affichage plus grande que le champs de vision

L'impression d'immersion n'est possible que si les bords de la surface d'affichage ne sont pas visibles par l'utilisateur, afin lui donner l'illusion, comme dans la réalité, que la seule limite est celle de son champs visuel. Il y a deux familles de techniques :

- La famille des *Head Mounted Display* (*HMD*), où le dispositif d'affichage est asservi aux yeux de l'utilisateur.
- La famille des CAVE ou simili-CAVE, où on utilise des surfaces d'affichage suffisamment importantes, et disposées de façon adéquate, de manière à donner l'impression à l'utilisateur d'être à l'intérieur d'une "bulle" d'affichage, à gauche dans la figure 2.4.



(a) Shéma du dispositif d'affichage d'une simili-CAVE, avec quatre écrans rétro-projetés

(b) Un utilisateur dans le dispositif *MUSE*

FIG. 2.3 – Le dispositif immersif *MUSE*

Le dispositif *MUSE* (Multi-User Stereoscopic Environment) que j'ai pu utiliser durant ce travail, dans le cadre de l'action transversale *VENISE* (Virtual ENvironment for Immersive Simulation and Experiments) [18], est un équipement immersif à deux écrans rétro-projetés de deux mètres de côté formant un angle droit, comme illustré à droite dans la figure 2.4.

2.2.2.2 La stéréoscopie

Cependant, la condition précédente, qui consiste à entourer l'utilisateur de dispositifs visuels, n'est pas suffisante pour lui donner l'illusion d'une immersion. Dans la nature, nous percevons le monde en 3 dimensions grâce à la vision binoculaire, qui permet à notre cerveau à partir de deux points de vue décalés, de reconstruire le monde en 3 dimensions. Dans une moindre mesure, l'audition binaurale permet de localiser la provenance d'une source sonore. Ainsi, une façon de donner l'impression d'être immergé dans une scène, est de présenter à chaque oeil un point vue différent de la scène, pour donner au cerveau de l'utilisateur l'illusion de perception d'une scène réelle en 3 dimension.

La réalité virtuelle utilise notre système visuel, pour simuler la stéréoscopie, en fournissant à nos deux yeux, une vision artificiellement décalée du monde virtuel. Plusieurs techniques permettent d'envoyer un signal différent à chaque oeil. Pour les HMDs, le problème ne se pose pas car il y a un écran par oeil, et chaque écran peut donc envoyer un signal différent à chaque oeil. En ce qui concerne les simili-caves ou les écrans traditionnels, il faut appliquer un filtre, pour ne présenter à chaque oeil que le signal qui lui est destiné. Il existe 3 grands types de filtres :

- les filtres fréquentiels, comme le filtrage couleur des lunettes de type anaglyphe (Rouge/Cyan).
- les filtres polarisés, qui permettent de filtrer la lumière en fonction de sa polarisation, verticale ou circulaire.
- les filtres temporels, qui permettent d'obstruer alternativement la vision de chaque oeil, à une fréquence suffisante pour donner une illusion de continuité.

(a) Lunettes avec filtrage anaglyphe (b) Lunettes avec filtrage polarisant (c) Lunettes avec filtrage alternatif

FIG. 2.4 – Lunettes pour la stéreovision

D'autre part, il existe deux méthodes pour dessiner les objets d'une scène tridimentionnelle en stéréo-vision :

- La stéréo simple, qui considère que l'utilisateur ne bouge pas dans le dispositif. La stéréo est donc construite à partir d'un point de vue unique et invariable dans le dispositif.

– La stéréo adaptative, qui, à partir de la position et de l'orientation des yeux de l'utilisateur, recalcule en permanence le point de vue à envoyer chaque oeil. Ce type de stéreo nécessite de connaître de la position et l'orientation du regard de l'utilisateur, à l'aide de technologie de tracking. Cette dernière est de bien meilleure qualité, car elle mime complètement la réalité.

Dans *MUSE*, la stéréoscopie peut être simple ou adaptative, avec un filtrage alternatif ou polarisée, de sorte que ce dispositif peut être reconfiguré pour étudier des interactions coopératives *in situ* entre deux utilisateurs. Un dispositif de tracking est positionné sur les lunettes, de manière à pourvoir suivre la position et l'orientation du regard de l'utilisateur nécessaire à la réalisation de la stéréoscopie adaptative.

Dans la nature, nous utilisons aussi l'ouïe pour percevoir notre environnement. En ce qui concerne la stéréoscopie auditive, elle est plus difficile à réaliser. La perception de la position d'une source est dû à un déphasage temporel, mais il faut aussi prendre en compte plusieurs autres paramètres spécifiques, comme la morphologie et la résonance crânienne, propre à chaque individu.

2.2.3 Les périphériques d'interaction

L'échelle du dispositif, l'immersion de l'utilisateur au sein de son application, la posture souvent debout et active de l'utilisateur, rend rédhibitoire toute réutilisation de dispositifs d'interaction classique d'un cadre de bureau, comme le clavier ou la souris. De nombreux dispositifs d'interaction ont donc été développés et commercialisés, pour répondre à la spécificité de ce cadre, et la recherche demeure très active dans ce domaine.

A travers les dispositifs d'interaction, il existe deux modalités de communication avec l'application : les dispositifs qui envoient des données à l'application, comme les capteurs de position d'orientation (figure 2.5), ou les gants de données (figure 2.7), ou les dispositifs de reconnaissance vocale. Ces dispositifs permettent d'agir sur le monde virtuel, par l'intermédiaire de l'application qui va interpréter ces données en entrée. Du point de vue de l'application, ces modalités de communication sont appelées modalités en entrée.

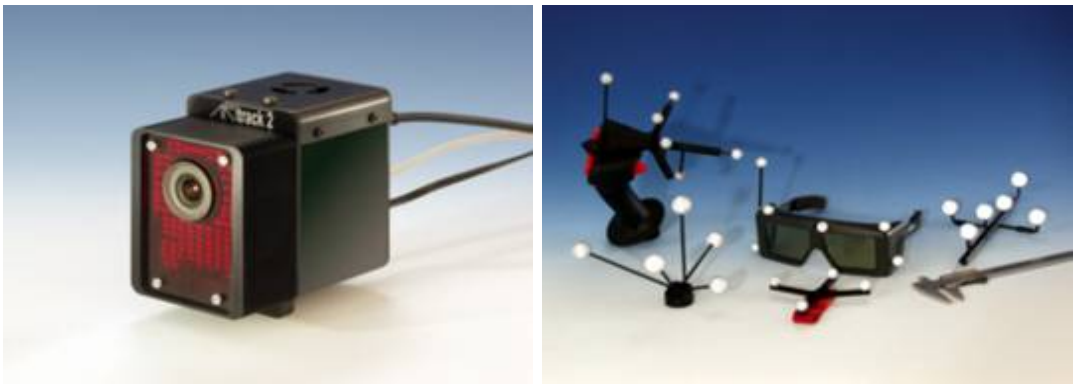
D'autres dispositifs reçoivent des données de l'application, pour les présenter à l'utilisateur selon 3 modalités : le retour visuel, par l'intermédiaire des projecteurs, retour sonore, ou bien retour haptique (figure 2.8). Ces modalités en sortie, permettent à l'utilisateur, à travers les dispositifs d'interaction, de sentir, de percevoir le monde virtuel dans lequel il évolue.

Certains dispositifs, comme les bras à retour d'effort, ou les gants haptiques ou vibrotactiles, sont à la fois en entrée et en sortie. Les mouvements de l'utilisateur qui manipule le bras sont transmis à l'application (entrée), l'application envoie des

informations au bras pour calculer un retour d'effort (sortie) devant être perçu par l'utilisateur.

2.2.3.1 Le tracking de l'utilisateur

Pour qu'une rétroaction soit possible entre l'utilisateur et un environnement virtuelle dans une une simili-cave, il est nécessaire de pouvoir suivre le positionnement et les mouvements de cet utilisateur. La technologie de tracking utilisée dans *MUSE*, permet à partir de plusieurs caméras infrarouges (*cf.* figure 2.5) qui filment quelques réflecteurs infrarouges habilement positionnés sur l'utilisateur pour constituer un référentiel, afin de suivre la position et l'orientation de ce référentiel attaché à l'utilisateur.



(a) Caméra infrarouge ArtTrack© (b) Réflecteurs infrarouges

FIG. 2.5 – Système de tracking ArtTrack©, utilisé dans *MUSE*

2.2.3.2 La souris 3D

Par analogie à la souris traditionnelle qui évolue sur un plan dans la main de l'utilisateur, et dont l'avatar sur l'écran est un pointeur, une souris 3D (*cf.* figure 2.6) permet de pointer les objets dans l'espace, en trackant la position et l'orientation d'un dispositif que l'utilisateur tient dans sa main. Le pointeur est par exemple une ligne qui permet de sélectionner un objet virtuel lorsque celle-ci entre en collision avec ce dernier, en appuyant sur un de ses plusieurs boutons, comme dans le cadre classique.

2.2.3.3 Les gants de données

Un autre dispositif couramment utilisé dans un environnement immersif est le gant de données, qui permet à l'application de connaître la configuration de la main

(a) Souris 3D sans fil, avec système ArtTrack© (b) Wanda™, Ascension Technology Corporation©

FIG. 2.6 – Souris 3D

de l'utilisateur, dont un des buts est de permettre des interactions directes avec les objets virtuels perçus en relief par l'utilisateur. Pour ce faire, un avatar de la main de l'utilisateur est en général présenté dans le monde virtuel. L'application peut ainsi interpréter une position et une configuration de cet avatar par rapport à un objet virtuel, comme une action à appliquer à cet objet. Certains gants intègrent aussi des modalités haptiques, avec des vibreurs au bout de chaque doigt, ou en intégrant des dispositifs à retour d'effort plus sophistiqués, comme les dispositifs présentés dans la figure 2.7.

(a) *CyberTouch*^{TM}, avec retour vibrotactil (b) *CyberGrasp*^{TM}, avec retour d'effort

FIG. 2.7 – Gants de données

2.2.3.4 Les interfaces haptiques

Enfin, les interfaces « haptiques » ou « à retour d'efforts » sont utilisées pour reconstituer, à l'aide de moteurs qui impriment des mouvements en sens contraires, certaines sensations physiques liées à la topologie du monde virtuel. La détection de collision entre l'avatar du bras et des objets dans le monde virtuel permettent de ressentir les limites physiques ou la forme d'un objet. Le bras permet aussi de rendre ressentir des champs de forces, des textures à la surface des objets. La figure 2.8 montre le dispositif de retour haptique actuellement diponible dans *MUSE*.

FIG. 2.8 – Virtuose de chez *Haption*

2.2.4 Une gestion distribuée des périphériques

Pour gérer l'ensemble de ces périphériques et des services nécessaires à la mise en place d'un contexte immersif de réalité virtuelle, un environnement logiciel *EvServer* [107] [106] a été développé dans le cadre de l'action transversale *VENISE* [18] par Damien TOURAINE dans le cadre de sa thèse. Les caractéristiques principales de l'*EvServer* résident dans sa gestion des périphériques. Plusieurs ordinateurs en réseau gèrent des périphériques d'interaction, et collaborent pour mettre ces périphériques à la disposition de l'application cliente, par l'intermédiaire d'évènements qui transitent sur le réseau. Cette architecture matérielle et logicielle a été utilisée afin de développer les paradigmes d'interaction et d'exploration immersifs décrits dans le chapitre suivant.

2.3 Conclusion

Nous avons, dans la première partie de ce chapitre, présenté une caractérisation des données génomiques, hétérogènes selon plusieurs points de vue, mais qui partagent des points communs qui seront utilisés afin de pallier cette hétérogénéité. Ces caractéristiques, ajoutées aux habitudes de visualisation des biologistes qui transparaissent dans la sélection d'outils présentée dans le premier chapitre, nous ont permis de définir les modalités de représentation visuelle adaptées à ces données, paradigmes de visualisation qui seront présentés dans le chapitre suivant. Par ailleurs, nous avons choisi le contexte immersif pour explorer les données génomiques, d'une part car certaines de ces données, sont, par nature tridimentionnelle, et d'autre part parce que les données génomiques sont massives. Le cadre immersif a démontré dans de multiples travaux [55] [64] [102], qu'il pouvait répondre à cette double problématique. Nous avons, donc, dans la seconde partie de ce chapitre, décrit les spécifités du contexte immersif, dans lequel nous expliquons les techniques utilisées pour donner l'illusion d'immersion. Nous avons fait une sélection des dispositifs d'interaction disponibles dans le dispositif que nous avons utilisé, sur lesquels nous nous appuieront dans le prochain chapitre, pour définir les paradigmes d'exploration et d'interaction entre l'utilisateur et ses données.

Chapitre 3

Représentation, exploration et interaction

Si tu veux progresser vers l'infini, explore le fini dans toutes les directions

F.W Goethe

L'enjeu de ce travail est de concevoir des paradigmes d'exploration et d'interaction avec les données génomiques, capables de couvrir leur variété et de gérer leur caractère massif, et adaptés à la fois au cadre immersif et aux besoins des biologistes. Nous décrirons dans ce chapitre, comment nous avons utilisé les spécificités des bases/banques de données génomiques textuelles ou factuelles et leurs points communs, pour concevoir puis développer un paradigme de visualisation qui puisse répondre à ces besoins. Ce chapitre sera aussi l'occasion de décrire plusieurs paradigmes d'interaction avec ces données utilisant les dispositifs spécifiques au contexte immersif, présentés dans le précédent chapitre.

3.1 Modèle générique de représentation des données génomiques

Malgré l'aspect hétérogène de format, de contenu et de qualité dont font preuve les banques de données génomiques, nous avons vu que la structuration globale des bases de données factuelles est relativement commune pour une grande partie d'entre elles. Elles ne traitent qu'un seul type d'objet biologique à la fois, et chaque instance est associée à un ensemble d'attributs descriptifs. Ce qui relient les instances au sein d'une base, mais aussi ce qui structure ces bases entre elles, sont dans la plupart des cas des relations binaires, d'identité pour les alias et les références croisées entre

les banques, de similarité de séquence, mise en évidence par leur alignement, de similarité fonctionnelle, d'appartenance à une même voie métabolique, ou de relation de nature sémantique, comme dans *GeneOntology* [3]. La plus grande partie des données génomiques, qu'elles soient textuelles ou factuelles, s'organisent donc en réseaux, à plusieurs échelles. Compte-tenu de la nature fortement relationnelle de ces banques de données, la structure d'une base de données peut dans la plupart des cas être assimilée à un graphe, dans lequel les sommets sont les instances des objets biologiques d'intérêt concernant cette base. Les arêtes de ce graphe traduisent les relations ayant un fondement biologique entre ces instances. Ces graphes, sont parfois des arbres, comme c'est le cas pour les arbres phylogéniques, ou pour certaines ontologies, mais dans la plupart des cas, ce sont des graphes qui ne présentent pas de caractéristiques topologiques particulières.

Nous avons vu aussi que les données génomiques, sont des données multivariées. En effet, les bases de données génomiques, excepté pour les bases intégratives, sont souvent centrées sur des objets biologiques d'intérêt de même type, associés à un ensemble d'attributs descriptifs, qualitatifs, ou quantitatifs, constituant une liste homogène d'attributs à tous les objets au sein d'une même base.

La représentation qui consiste à considérer les banques/bases de données comme une liste d'objets munis d'attributs, n'est cependant pas suffisante. Dans les bases, les objets sont explicitement reliés entre eux par des relations ayant un fondement biologique. Les relations ne sont pas nécessairement dépendante des attributs décrivant ces objets, comme celle qu'on construit dans les méthodes classiques d'analyse de données multidimensionnelles. Ces relations peuvent être, comme c'est souvent le cas, des résultats d'expérience, qui relient, par exemple, un substrat et son produit par la découverte d'une nouvelle enzyme, ou en matière d'extraction d'information, un verbe qui relie deux termes entre eux. Les relations sont donc porteuses d'informations, que l'on considéra dans le cas général comme indépendantes de la liste des attributs des objets qu'elles relient. De même que pour les objets, ces relations peuvent être décrites par un ensemble d'attributs de nature qualitative ou quantitative.

La structure de donnée qui émerge après une analyse sur des données génomiques variées à plusieurs échelles, est une structure de graphe multivarié (ou multidimensionnel), dans lequel les sommets et les arêtes jouent un rôle équilibré, les sommets et les arêtes représentant des données multivariées. En effet, dans cette représentation, les informations ne sont pas concentrées dans les sommets du graphe, et les arêtes, au même titre que les sommets, sont porteuses d'informations. Chaque sommet et chaque arête de ce graphe est associé à un vecteur de valeurs descriptives, de nature qualitative ou quantitative. Un vecteur de valeurs pour un sommet est une sélection parmi les attributs décrivant l'objet de la base que représentera ce sommet. Le type des vecteurs est le même pour tous les sommets du graphe, compte tenu de

l'homogénéité de la liste des attributs décrivant les objets d'une même base. Pour une arête, un ensemble de valeurs est une sélection de relations entre deux objets de la base. De même, la sémantique des ensembles de valeurs est la même pour toutes les arêtes, au sein d'un même graphe. Cette structure est décrite plus précisément ensuite, et une instance d'un tel graphe, appliquée à des données biologiques, est présentée dans la figure 3.1.

Un graphe multivarié est défini par le sextuplé (V, E, C, D, f, g) avec
- V un ensemble fini de sommets du graphe.
- E une partie de $V \times V$ qui constitent les arêtes de ce graphe.
- C une partie de $\mathbb{A}^n$, avec $\mathbb{A}$ un ensemble d'attributs qualitatifs ou quantitatifs.
- D une partie de $\mathbb{B}^m$, avec $\mathbb{B}$ un ensemble d'attributs qualitatifs ou quantitatifs.
- $f : V \rightarrow C$ une fonction qui associe à chaque sommet un vecteur d'attributs.
- $g : E \rightarrow D$ une fonction qui associe à chaque arête un vecteur d'attributs.

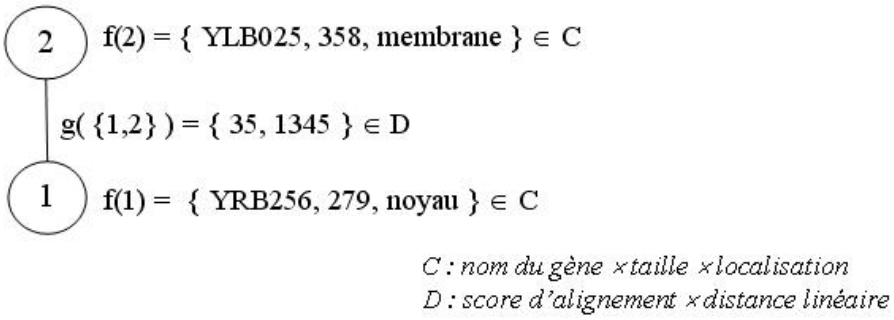


FIG. 3.1 – Graphe multivarié appliqué à des données biologiques, où les gènes 1 et 2 dont on connaît le nom, la taille en bp, et la co-localisation cellulaire, sont liés par un critère similarité de séquence, et de distance linéaire sur le chromosome

3.2 De la représentation informatique à la représentation graphique

Il s'agit désormais de définir des moyens de passer de cette représentation informatique à une représentation graphique. Nous avons présenté dans la section précédente une structure de données générique, que nous pensons refléter la structure sous-jacente des banques de données génomiques variées, avec l'utilisation de graphes multivariés, dans lesquels chaque sommet et chaque arête possède un vecteur de valeurs quantitatives ou quantitatives.

3.2.1 Représentation sous forme de graphes 3D

Cette interprétation de la structure globale des banques de données génomiques, les avantages d'un modèle à base de graphe, l'usage des biologistes, nous ont naturellement conduit à choisir une représentation visuelle de ces données sous forme de graphe. Dans la mesure où le cadre immersif donne accès à une troisième dimension, permettant de visualiser plus de données, ces graphes seront dessinés dans un espace tridimensionnel, afin de proposer à l'utilisateur une immersion au sein de ses données. La navigation autour et à l'intérieur de ce graphe, associé à la stéréo-vison, permet en général d'éliminer la gène souvent occasionnée lors de la visualisation de graphe 3D dans un cadre de bureau.

Par ailleurs, les représentations graphiques sous forme d'arbre ou de graphe sont déjà très utilisées par les biologistes, que ce soit pour visualiser des réseaux métaboliques [68] [33], des réseaux d'interaction protéiques [47], des réseaux d'expression, sous forme de graphes [88] ou d'arbres dendogrammes [45], et ne rompt donc pas avec leur usage. L'utilisation d'algorithmes efficaces issues de la théorie des graphes, offre de nombreuses approches pour simplifier ces graphes, et constitue une réponse possible à certaines problématiques associées à l'aspect massif des données. Par ailleurs, cette théorie offre des méthodes variées, répertoriées dans [30] ou [63], pour dessiner les arbres ou les graphes.

3.2.1.1 Inventaire des attributs graphiques d'un graphe 3D

Suite à ce choix, nous avons effectué un inventaire dans le tableau 3.2 des éléments graphiques qui constituent un graphe 3D, en s'inspirant de la liste des éléments applicables à ce contexte proposée par Colin Ware [110].

Une valeur quantitative dans un vecteur décrivant un objet, lorsqu'il est symbolisé par un sommet du graphe, peut être représentée, par la position de ce sommet, son volume, sa composante rouge, verte ou bleue, ou sa transparence. Plusieurs valeurs numériques d'un vecteur caractérisant un objet peuvent donc être visualisées

Sommet		Arête	
qualitatif	quantitatif	qualitatif	quantitatif
étiquette texte		étiquette texte	
	position : x, y, z		
	volume		épaisseur
			longueur
couleurs prédéfinies, tramage	composantes r, v, b	couleurs prédéfinies, tramage	composantes r, v, b
formes prédéfinies (point, cube, sphère, ellipse, tétraèdre...)		formes prédéfinies (ligne, cylindre droit, cylindre parabolique...)	
	transparence		transparence

FIG. 3.2 – Éléments graphiques atomiques d'un graphe 3D

simultanément. De même, une valeur quantitative dans un vecteur décrivant une relation entre deux objets, relation symbolisée par une arête dans le graphe, peut être représentée, par son épaisseur, sa longueur, sa composante rouge, verte ou bleue, ou sa transparence.

En ce qui concerne les valeurs qualitatives, associées aux objets ou aux relations, il est possible de les traduire visuellement par des étiquettes textes, des couleurs prédéterminées, ou bien des formes prédéterminées. La liste des formes prédéterminées est cependant spécifique selon s'il s'agit de représenter un sommet (point, cube, sphère, ellipse, tétraèdre) ou une arête (ligne, cylindre, paraboloïde).

Cet inventaire nous a permis de faire le point sur les ressources visuelles disponibles, qui pourront être utilisées simultanément et qui constitueront les briques de bases d'un graphe 3D, indépendamment de la nature des données.

3.2.2 Séparation entre données et représentations

Dans le précédent chapitre, nous avons insisté sur une caractéristique importante des systèmes de visualisation de données génomiques. Ces systèmes ont été conçus afin de visualiser des données spécifiques, avec méthode de représentation graphique des données souvent imposée, afin de faciliter une tâche biologique précise. L'évolution rapide des formats, et du contenu des bases, des modalités de ré présentation, diminue de manière assez draconienne la durée de vie de ces outils, qui deviennent très rapidement caduques. Par ailleurs, cette spécificité a aboutit à un cloisonnement important entre des données génomiques, qu'il est cependant, d'un point de vue biologique, très pertinent d'étudier ensemble. Enfin, chaque équipe de

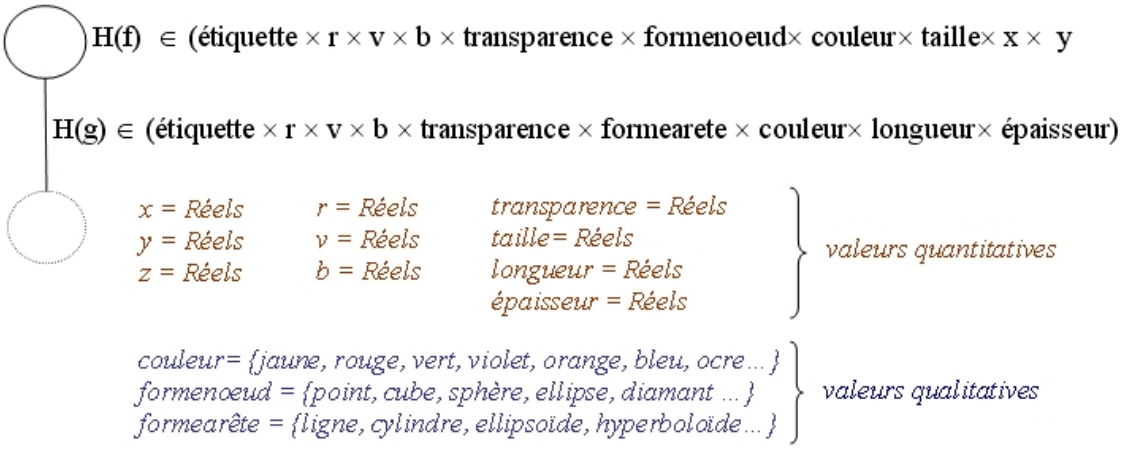


FIG. 3.3 – Espace d'arrivée de la fonction H de transformation d'un graphe multivarié en un graphe 3D

recherche, est habituée à un style visualisation qui n'est pas celle de l'équipe de recherche voisine, pour réaliser une tâche similaire sur des données identiques.

Nous avons donc identifié un besoin important auquel nous avons répondu en séparant complètement les données génomiques à analyser, de leur représentation visuelle. Pour ce faire, nous avons vu dans la section précédente que ces données, même très spécialisées, pouvaient souvent être synthétisées par des graphes multivariés. Nous avons aussi vu les modalités de traduction ce graphe multivarié, vers une représentation visuelle, par la transformation des attributs biologiques de nature quantitative ou qualitative d'objets et des relations de ce graphe, par des attributs graphiques de sommets et d'arêtes.

Pour assurer la séparation entre les données et leur représentation, nous avons défini donc deux couches : la couche *donnée* qui est une représentation du contenu des données à analyser (graphe multivarié), la couche *représentation* qui permet à l'utilisateur de choisir les modalités de représentation de ces données. Une fonction H (voir figure 3.3) transforme les attributs bioinformatiques quantitatifs et qualitatifs d'un graphe multivarié, par des attributs graphiques qualitatifs et quantitatifs répertoriés dans le tableau 3.2, qui peuvent être instanciés par des valeurs décrites dans la figure 3.3.

3.2.2.1 La couche *donnée*

Nous présentons dans les deux sous-sections suivantes, les modalités techniques qui permettent à l'utilisateur de coder ses données sous forme de graphe multivarié

(couche *donnée*), et comment il spécifie la fonction H, qui permet de transformer ces données en données graphiques (couche *représentation*).

La couche *donnée* fédère les données génomiques selon une représentation unique sous forme d'un graphe multivarié. Chaque objet est référencé par un identifiant unique (*id*). Ces objets sont reliés par des relations binaires (*id1 id2*). L'utilisateur associe à chaque objet (sommet) et à chaque relation binaire (arête), un vecteur d'attributs descriptifs de ces objets et de ces relations (*value0..valueN*). Ces vecteurs contiennent les valeurs quantitatives ou qualitatives descriptives de chaque objet biologique d'intérêt. Pour tous les objets, ces vecteurs doivent être de même type, c'est à dire de même taille, et les N^{emes} valeurs de chaque vecteur doivent être de type identique. C'est aussi le cas pour les vecteurs associés aux relations. Certaines valeurs peuvent ne pas être renseignées.

```
<data>
  <object id="0" value0="1R" value1="0.1" value2="right"/>
  <object id="1" value0="1L" value1="0.2" value2="left"/>
  ...
  <object id="22" value0="12R" value1="3.7" value2="right"/>
  <object id="23" value0="12L" value1="0.4" value2="left"/>
  ...

  <relation id1="23" id2="22" value1="22.0"/>
  <relation id1="1" id2="28" value1="15.0"/>
  ...
  <relation id1="11" id2="22" value1="22.0"/>
  <relation id1="12" id2="22" value1="9.0"/>
  <relation id1="22" id2="28" value1="17.0"/>
</data>
```

FIG. 3.4 – Exemple de codage des données génomiques en graphe multivarié en format XML

3.2.2.2 La couche *représentation*

La couche *représentation* permet de définir la fonction H (*cf.* figure 3.3), c'est à dire comment les attributs des sommets (*object*) et des arêtes (*relation*) du graphe multivarié décrits dans la couche *donnée*, seront traduits en attributs graphiques d'un graphe 3D, dont un inventaire a été réalisé au début de ce chapitre 3.2. L'utilisateur indique comment chaque valeur des vecteurs descriptifs des sommets et des arêtes, est traduite par ces attributs graphiques inventoriés précédemment. Les valeurs quantitatives peuvent être associées à des attributs graphiques quantitatifs

(*size* pour les sommets, ou *length* pour les arêtes), et les valeurs qualitatives à des attributs qualitatifs (*label, objectcolor*, pour les sommets). Une valeur par défaut peut être donnée comme ici la forme ou la couleur (*objectform et color*, pour les objets, *relationform et color* pour les arêtes).

```
<view>
  <object objectform="cube" color="red">
    <label map="value0"/>
    <size map="value1"/>
    <objectcolor map="value2">
      <objectdomaincolor value="left" mapto="red"/>
      <objectdomaincolor value="right" mapto="blue"/>
    </objectcolor>
  </object>
  <relation relationform="cylinder" color="green">
    <length map="value1"/>
  </relation>
</view>
```

FIG. 3.5 – Définition de la fonction H en format XML

Voici dans la figure 3.6 une instance de représentation de la couche *donnée* décrite précédemment dans la figure 3.4, par la couche représentation de la figure 3.5.

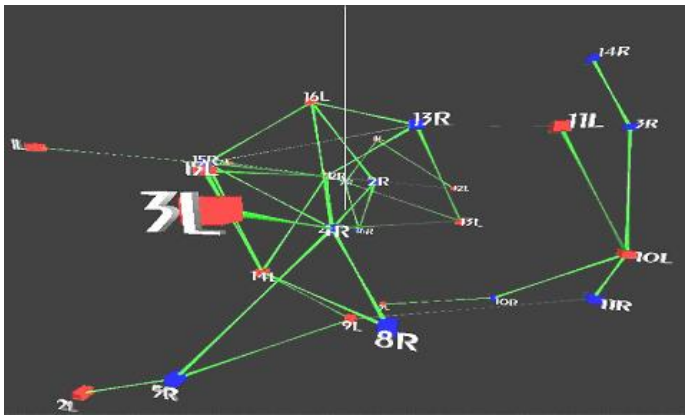


FIG. 3.6 – Illustration de la traduction du graphe multivarié vers un graphe 3D

On peut constater que nous n'avons pas utilisé des formats classiques disponibles dans la littérature, comme *GML* [16], *GraphML*, ou ceux de *Graphviz* [5], mais un format très simplifié de description de graphe multivarié. Ces formats permettent certes de décrire très précisément un graphe, son contenu, et la façon de le dessiner,

mais ne permettent pas facilement de définir un graphe multivarié, d'une part, ni de séparer cette description, indépendamment de toutes considérations graphiques, de sa représentation visuelle, d'autre part, comme nous souhaitons le faire ici. On peut cependant souligner un effort important effectué dans le format *GraphXML* [62], qui donne la possibilité, au sein d'un fichier unique, de séparer la structure topologique d'un graphe (sommets, arêtes et hiérarchie de graphe), et la couche géométrique et graphique. Cette séparation est techniquement réalisée par l'indépendance des balises spécifiant la structure du graphe, et des balises définissant les modalités de représentation géométriques et graphiques, comme la position, la taille, la forme et la couleur des sommets, ou le type des arêtes (droite, courbe, orthogonale).

Cette séparation entre la couche *donnée* est primordiale pour les raisons suivantes : l'utilisateur choisit comment visualiser ses données, et lui permet surtout, sans avoir à les modifier, d'avoir plusieurs représentations visuelles différentes. L'utilisateur peut ainsi comparer plusieurs visualisations, chacune focalisée sur un critère différent. Certains attributs graphiques, comme la longueur des arêtes, sont les piliers qui sur lesquels reposent l'organisation spatiale des données dans la représentation visuelle. L'association variable d'attributs biologiques ou bioinformatiques des données à associer à cet attribut graphique, permet d'explorer l'organisation globale sous jacente de banques de données génomiques, par rapport à plusieurs critères biologiques ou bioinformatiques : co-localisation cellulaire, similarité de séquence, similarité fonctionnelle, proximité phylogénique, co-expression, appartenance à un même chemin métabolique, etc.

3.2.3 Normalisation de valeurs quantitatives

La transformation des données vers une représentation pose cependant problème au niveau de la traduction des valeurs quantitatives par des attributs graphiques quantitatifs. En effet, le domaine de départ constitué par des données quantitatives, n'est pas le même que le domaine d'arrivée, les attributs graphiques quantitatifs. Certains attributs quantitatifs comme, les composantes qui définissent une couleur (rouge, vert, bleu), ou la transparence, sont des valeurs réelles bornées entre 0 et 1. De même, la taille des objets graphiques qui constituent les sommets, doit nécessairement être bornée. En reflétant exactement la valeur de l'attribut auquel cette taille a été associée, on obtient dans la scène des objets trop petits ou trop grands ; de même pour la largeur des arêtes. Une normalisation linéaire est donc effectuée lors de la transformation d'une valeur quantitative par un attribut graphique quantitatif, qui dépend des valeur maximales et minimales des attributs dans le domaine de départ (*donnée*) et d'arrivée (*représentation*). Les bornes dans l'ensemble d'arrivée est arbitrairement fixée pour éviter d'obtenir des objets trop gros ou trop petits, ou trop transparents. Ce problème de la transformation de métriques en attributs

graphiques est traité et discuté dans [61], et propose une normalisation pour prendre en compte la répartition pas nécessairement uniforme des données à visualiser.

3.2.4 Gestion et affichage des étiquettes

Par ailleurs, un des attributs graphique listé, associable à un sommet ou à une arête est l'étiquette. L'affichage d'étiquettes posent de multiples problème en visualisation d'information. Elle doivent être lisibles, facilement associables aux objets qu'elles sont censées décrire, et ne doivent pas occulter la visualisation des autres objets. Fekete et Plaisant propose dans [48], une taxonomie des techniques d'affichage des étiquettes, ainsi qu'une méthode par affichage radiale d'étiquette autour d'un nuages d'objets graphiques, zone d'intérêt indiquée par l'utilisateur (focus). Pour faciliter l'association des étiquettes avec l'objet qu'elle décrivent, une ligne de la couleur de l'objet relie celui-ci à une étiquette, entourée d'une rectangle de cette même couleur. Dans notre application, nous avons décidé de les présenter sous forme d'objet 3D avec une largeur, une hauteur une profondeur. En effet, l'utilisation de *billboards*, technique qui consiste pour afficher du texte à le plaquer sur un polygone sous forme d'une image, toujours orienté face à l'utilisateur, provoque une gène par atténuation de l'effet stéréoscopique. Cette méthode nécessite aussi la génération d'une image par étiquettes textuelles, et donc de plusieurs milliers d'images dans le cas de gros graphes. Enfin, la sélection d'objet 3D avec la souris 3D est plus facile sur étiquettes 3D, que pour les objets sans volume. La librairie de génération de police 3D utilisée est *GLF* [4]. Cependant, les étiquettes 3D sont des objets complexes, l'affichage de plusieurs milliers de d'étiquette est assez lourd. Pour accélérer l'affichage, nous avons utilisé une capacité de la plupart des cartes graphiques, qui permet d'accélérer l'exécution d'un ensemble de primitives graphiques souvent répétés, et donc d'accélérer l'affichage d'objets 3D, par un préchargement de ces primitives dans une liste appelée *display list*. Chaque lettre de l'alphabet est au préalable associée à une *display list*, pour être plus tard dessinée dans la scène de manière plus rapide. D'autre part, seules les étiquettes lisibles sont affichées. En fonction de la taille de la police et de la distance de l'étiquette et l'utilisateur déterminée empiriquement, l'étiquette est considérée comme lisible ou pas, et est ensuite affichée, ce qui réduit considérablement le nombre d'étiquettes à afficher, et améliore par conséquent la fluidité du dessin. Enfin, plusieurs idées non encore implémentées, et proposées dans [48] sont intéressantes : afficher uniquement les étiquettes d'une zone d'intérêt permet de décharger la visualisation, et l'utilisation d'une couleur d'étiquette identique à celle de l'objet, si cette couleur n'est pas sémantiquement liée à une autre information, permettrait de faciliter l'association. Cependant, le dessin radial et l'utilisation d'une ligne reliant l'étiquette à l'objet n'a pas été reprise. Bien adaptée au structure de type nuage de point, cette méthode crée de nouveaux liens entre des objets graphiques, qui s'ajoutent à ceux déjà nombreux de la représentation du graphe (arête).

C'est la forte proximité spatiale qui permet d'associer l'étiquette à l'objet.

3.3 Traduction d'une proximité biologique par une distance spatiale

Cependant, derrière ces modalités de traduction d'une représentation informatique vers une représentation graphique, se cache un problème de taille : la traduction d'une valeur quantitative associée à la longueur d'une arête, revient à fixer une contrainte de distance entre les deux sommets adjacents à cette arête. Si ce type de contraintes est appliqué à toutes les arêtes d'un graphe, elles sont souvent impossibles à résoudre lors du dessin du graphe représentant ces données dans un espace euclidien, comme le montre le schéma de la figure 3.7, sur lequel des contraintes de distances sont impossibles à respecter dans le plan, comme dans l'espace.

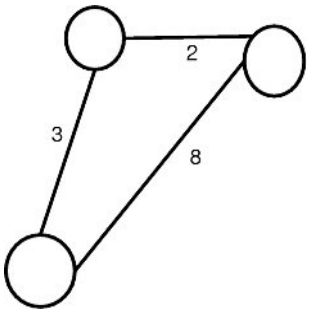


FIG. 3.7 – Un exemple de contraintes de distance insolubles

Pour s'affranchir de ce problème difficile [44], il faut trouver une heuristique de dessin qui permettent de résoudre partiellement ces contraintes de distance. Il faut aussi garder à l'esprit que cette méthode doit être applicable au contexte tridimensionnel et interactif associé au cadre immersif.

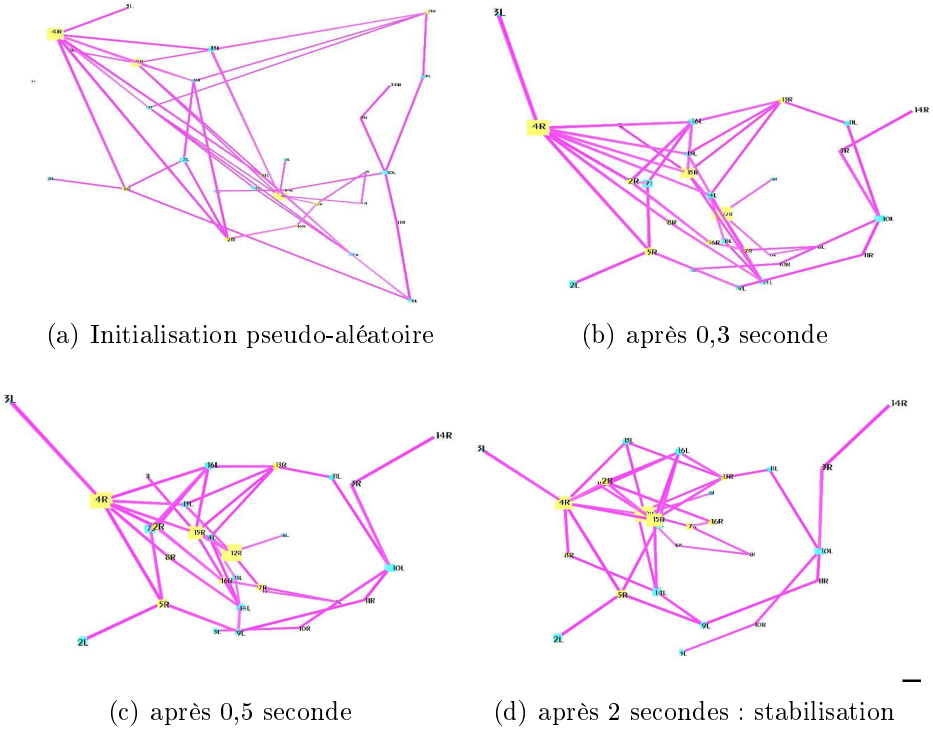
3.3.1 Approximation par simulation de forces

Il existe plusieurs méthodes pour résoudre ce problème, comme les approches basées sur l'optimisation combinatoire de Kamada et Kawai [71], les algorithmes génétiques [46], mais la plupart dérivent d'une approche par analogie physique proposée par Eades et al. en 1984 [42]. Nous avons exclu les méthodes de résolution par algorithme génétique, car celles-ci nécessitent de se constituer une population de nombreux graphes afin de sélectionner le meilleur selon des critères d'esthétisme, ou d'approximation d'une contrainte. Le graphe à dessiner est donc dupliqué de nombreuses fois, et est donc difficilement applicable à de gros graphes, et par conséquent, pas adapté à la gestion de données massives.

La méthode proposée par Eades permet de dessiner des graphes dans le cas général, en simulant un système physique de type masse-ressort, décrite dans l'algorithme 1. Dans cette simulation, les sommets sont des masses, mobiles, et les arêtes sont des ressorts. Les sommets sont initialement placés aléatoirement (ligne7). A chaque arête est associée une force attractive pour rapprocher deux sommets connectés jusqu'à une distance choisie. Des forces de répulsion (ligne 4) sont appliquées entre tous autres sommets du graphe, pour éloigner les sommets non connectés, et/ou compenser la force attractive (ligne 6) précédemment appliquée, ceci afin améliorer la lisibilité du graphe. La somme de ces forces est ensuite appliquée à chaque sommet du graphe, et modifie leur position (ligne 17). Ce processus est répété de façon itérative $|V|_{iter}$ fois (ligne 8), et le calcul d'une fonction d'énergie globale du système *temp* (ligne 17), permet, de diminuer la contribution de toutes les forces, d'évaluer la qualité de l'approximation du dessin, et est parfois utilisée comme condition pour la terminaison pour l'algorithme. La figure 3.8 est une illustration de cette approche.

Algorithme 1 Algorithme de dessin de graphe en $\theta(N^3)$

1:
Entrées : Un graphe avec $G(V, E)$ avec des longueurs d'arêtes entre u et v notée l_{uv}
2:
Sorties : Dessin λ de tous les sommets $u \in V$, de position λ_u
3: calcule la force de répulsion en fonction de la distance réelle d
4: **Fonction** $f_r(d)$:=**retourne** $-k/d$
5: calcule la force de ressort en fonction de la distance réelle d et de la distante désirée l
6: **Fonction** $f_s(d, l)$:=**retourne** $k(d - l)/l$
7: Positionner les sommets de manière aléatoire
8: **Pour** $0 \leq t \leq |V|_{iter}$ **Faire**
9: **Pour tout** $v \in V$ **Faire**
10: **Pour tout** $u \in V$ **Faire**
11: $\Delta \leftarrow \lambda(u) - \lambda(v)$
12: **Si** $(u, v) \in E$ **Alors**
13: $F_v(t) \leftarrow F_v(t) + (\Delta/\|\Delta\|) \times f_s(\|\Delta\|, l_{uv})$
14: **Sinon**
15: $F_v(t) \leftarrow F_v(t) + (\Delta/\|\Delta\|) \times f_r(\|\Delta\|)$
16: **Fin Si**
17: $\lambda(v) \leftarrow \lambda(v) + temp \times F_v(t)$
18: **Fin Pour**
19: **Fin Pour**
20: **Fin Pour**



(a) Initialisation pseudo-aléatoire (b) après 0,3 seconde

(c) après 0,5 seconde (d) après 2 secondes : stabilisation

FIG. 3.8 – Visualisation des itérations d'un algorithme de dessin de graphe de type *FDP* (Force Directed Placement)

Nous avons préféré les méthodes par analogie physique aux méthodes déterministes, car elles furent l'objet de nombreuses optimisations afin de dessiner de plus en plus rapidement des graphes. De plus, compte-tenu de sa nature itérative, elle est très adaptée au contexte interactif, car l'utilisateur, à chaque itération, peut effectuer des modifications de contrôle sur le dessin. Cette approche est particulièrement pratique, puisque elle peut être arrêtée à tout moment, pour une visualisation intermédiaire ou un modification du graphe par exemple, pour être reprise, à partir des résultats précédents.

Ces caractéristiques convergent avec les besoins des biologistes, qui modifient souvent leur base par ajout de nouvelles données, par suppression de données, par mise à jour d'attributs, souhaitant voir en temps interactif les répercusions de ces modifications sur l'organisation de leurs données dans la visualisation.

De plus, cette solution s'adapte facilement au dessin de graphe dans un espace tridimensionnel. Dans un cadre traditionnel de bureau, on fixe une zone dans laquelle le graphe doit être dessiner, afin d'avoir une vue d'ensemble de l'intégralité du graphe. Un des problèmes de cette méthode dans le plan est de dessiner le graphe à l'intérieur de limites physiques. Dans le contexte de réalité virtuelle, l'immersion au sein du graphe 3D, associée à la navigation dans ce graphe, permet de s'affranchir de cette contrainte. La navigation permet en effet de changer de point de vue, pour avoir une vision globale ou locale du graphe.

Cependant l'application brutale de cette méthode, dite *FDP* pour "Force Directed Placement", comme dans l'algorithme 1, est problématique à cause de sa complexité. Les forces de répulsion appliquées entre toutes les paires de sommets du graphe, tirent vers le haut la complexité d'une seule itération de l'algorithme en $\theta(N^2)$, N étant le nombre de sommets. En effet, le nombre d'itérations nécessaires pour aboutir à une terminaison, est dans la plupart des cas, proche de N, ce qui rend sa complexité en $\theta(N^3)$. Cette approche a donc été de nombreuses fois raffinée, afin de diminuer sa complexité ou d'améliorer la qualité du dessin. Fruchtermann et Reingold [50] ont proposé une structure de grille qui permet de limiter la répulsion des sommets en l'appliquant uniquement à leur voisinage, réduisant ainsi la complexité du phénomène de répulsion en moyenne à $\theta(N)$, et donc la complexité globale en $\theta(N^2)$.

Cette approche *FDP* est une des méthodes utilisées pour représenter des données multidimensionnelles (*MDS* pour MultiDimensionnal Scaling). La nature de ces données est décrite dans [32], ainsi que des descriptions détaillées des méthodes déterministes (résolution par des techniques d'algèbre linéaire) ou non déterministes (résolution de type masse-ressort) pour les analyser ou les visualiser. L'objectif des techniques de visualisation ces données, est de les plonger d'un espace multidimensionnel, vers un espace bidimensionnel ou tridimensionnel, en respectant une topologie des données imposées par une métrique. D'abord appliquée de manière brute [36] sur les données multidimensionnelles (complexité comme en $\theta(N^3)$), l'algorithme de placement été raffiné en déterminant au préalable deux ensembles de taille bornée pour chaque sommet. Le premier ensemble P pour un sommet, est constitué des sommets les plus proches (ligne 9), et le second ensemble L est constitué de quelques sommets distants (ligne 11). Cette optimisation s'appuie sur le fait que ce sont les forces entre les sommets les plus similaires qui jouent un rôle significatif, tout en conservant quelques contraintes d'éloignement pour les moins similaires. Cette optimisation en appliquant à chaque sommet du graphe (ligne 17) un nombre constant de forces ($|P| + |L|$), permet de diminuer la complexité de chaque itération à $\theta(N)$, diminuant de fait la complexité totale de l'algorithme à $\theta(N^2)$. Il n'y a pas de notion de répulsion, puisque le graphe est complet, mais uniquement des forces de ressort (ligne 1 et 31).

Algorithme 2 Algorithme de dessin de graphe en $\theta(N^2)$

Entrées : Un graphe complet $G(V, E)$ avec une matrice de similarité entre les sommets u et v notée l_{uv}

Sorties : Dessin λ de tous les sommets $u \in V$, de position λ_u

Fonction $f_s(d, l)$:=**retourne** $k(d-l)/l$
Positionner les sommets de manière aléatoire
$p \leftarrow constante$, $v \leftarrow constante$
Pour tout $v \in V$ **Faire**
    $P \leftarrow \emptyset$, $L \leftarrow \emptyset$, $cpt \leftarrow 0$
    Tant que $cpt < l + p$ **Faire**
        Choisir aléatoirement un sommet u
        Si $l_u v < Maximum(l_{xv}), x \in P$ **Alors**
            $P_v \leftarrow P_v + u$
        Sinon
            $L_v \leftarrow L_v + u$
        Fin Si
        $cpt \leftarrow cpt + 1$
    Fin Tant que
Fin Pour
Pour $0 \leq t < N_{iter}$ **Faire**
    Pour tout $v \in V$ **Faire**
        $L_v \leftarrow \emptyset$, $cpt \leftarrow 0$
        Tant que $cpt < l + p$ **Faire**
            Choisir aléatoirement un sommet u
            Si $l_{uv} < Maximum(l_{xv}), x \in P$ **Alors**
                $P_v \leftarrow P_v + u$, $L_v \leftarrow L_v + x$
            Sinon
                $L_v \leftarrow L_v + u$
            Fin Si
            $cpt \leftarrow cpt + 1$
        Fin Tant que
    Fin Pour
    Pour tout $u \in L_v \cup P_v$ **Faire**
        $\Delta \leftarrow \lambda(u) - \lambda(v)$
        $F_v(t) \leftarrow F_v(t) + temp \times (\Delta / \|\Delta\|) \times f_s(\|\Delta\|, l_{uv})$
    Fin Pour
Fin Pour

Nous avons adapté l'algorithme précédent à notre situation. Par rapport aux données multidimensionnelles, nous ne disposons pas de matrice de similarité, qui fournit une distance entre tous les sommets. Cette distance pourrait être calculée en utilisant les algorithmes de plus court chemins entre toutes les sources qui ne dispose pas d'une mesure de similarité, mais la complexité de cette méthode reste trop élevée, en $\theta(NE \log N)$ pour l'algorithme de Dijkstra. Cette différence peut paraître subtile, mais elle est primordiale, car la notion d'ensemble proche, et d'ensemble éloigné s'en trouve profondément modifiée dans la mesure où nous ne disposons pas de critère de distance entre chaque paire de sommets. Nous avons donc redéfini ces deux ensembles dans l'algorithme 3. Un ensemble proche P sont les sommets adjacents les plus proches (ligne 18), et un ensemble éloigné L d'un sommet est un ensemble de sommets non connectés à ce dernier. Cependant, nous avons constaté lors de l'application de cette approche à un graphe, que le phénomène de répulsion était minimisé, car la distance entre les sommets choisis aléatoirement pour se repousser était en général trop grande pour s'appliquer. Sans ces forces de répulsion nécessaires à la lisibilité, la qualité du dessin s'en trouve fortement réduite. Nous avons donc voulu réduire l'ensemble L pour un sommet u, aux sommets proches spatialement de u dans le dessin. Pour ce faire, il fallait trouver une méthodologie efficace qui permette d'attribuer à un sommet un ensemble de voisins proches dans le dessin. Nous avons donc introduit une méthode de découpage de l'espace de dessin par octree (*cf.* annexe E), qui permet d'obtenir un ensemble S_v de sommets dans la même subdivision de l'espace de v dans le dessin (ligne 12). La construction de cet octree (ligne 6) est appelée après le positionnement aléatoire de l'algorithme (ligne 5). On vérifie cependant lors du choix d'un voisin proche spatialement, qu'il ne fait pas partie des voisins adjacents de v (ligne 13). On notera aussi que la listes d'adjacence de chaque sommet est triée par insertion dichotomique à la lecture du graphe, de complexité $\theta(\log(Degre_{Moy}))$ par arête. On utilise cette propriété, pour extraire rapidement les p plus proches voisins (ligne 18).

La reconstruction de l'octree, pour prendre régulièrement en compte le déplacement des sommets est ensuite appelée de temps en temps, mais un nombre constant (O_{MAX}) de fois (ligne 21). L'algorithme 3, comme celui de Chalmers [36], est de complexité quadratique, puisque L et P sont de taille constante, et puisque la complexité d'insertion de tous les sommets dans l'octree en ($\theta(N)$) n'augmente pas la complexité globale, car cette construction n'est effectuée qu'un nombre constant de fois.

Algorithme 3 Algorithme avec gestion des répulsions par octree en $\theta(N^2)$

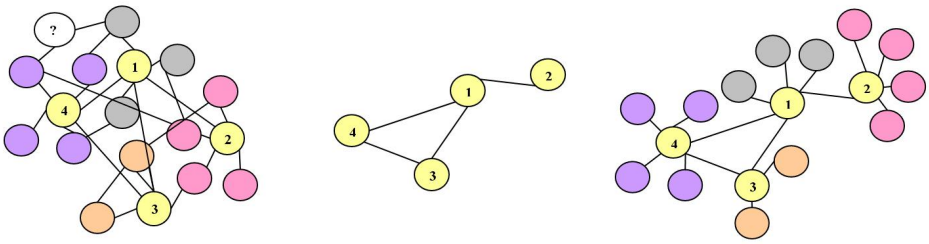
Entrées : Un graphe avec $G(V, E)$ avec des contraintes de longueur notée l_{uv}

Sorties : Dessin λ de tous les sommets $u \in V$, de position λ_u

Fonction $f_r(d)$:=**retourne** $-k/d$
Fonction $f_s(d, l)$:=**retourne** $k(d-l)/l$
Positionner les sommets de manière aléatoire
$RemplirOctree(\lambda, K)$ K la profondeur de l'octree
$p \leftarrow constante$, $l \leftarrow constante$
Pour tout $v \in V$ **Faire**
    $P_v \leftarrow \emptyset$, $L_v \leftarrow \emptyset$
    $cpt \leftarrow 0$, $pcpt \leftarrow l$
    Tant que $cpt < l$ **Faire**
        Choisir aléatoirement un sommet u dans S_v, voisinage spatial de v
        Si $u \notin A_v$ **Alors**
            $L \leftarrow L + u$
            $cpt \leftarrow cpt + 1$
        Fin Si
    Fin Tant que
    $P_v \leftarrow$ les p plus proches sommets parmi les adjacents à v
Fin Pour
Pour $0 < t < N_{iter}$ **Faire**
    Appeler de temps en temps $ReconstruireOctree(\lambda, K)$
    $L \leftarrow \emptyset$, $cpt \leftarrow 0$
    Tant que $cpt < l$ **Faire**
        Choisir aléatoirement un sommet u dans S_v, voisinage spatial de v
        Si $u \notin A_v$ **Alors**
            $L \leftarrow L + u$
        Fin Si
    Fin Tant que
    Pour tout $v \in V$ **Faire**
        Pour tout $u \in P$ **Faire**
            $\Delta \leftarrow \lambda(u) - \lambda(v)$
            $F_v(t) \leftarrow F_v(t) + (\Delta/\|\Delta\|) \times f_s(\|\Delta\|, l_{uv})$ force ressort
        Fin Pour
        Pour tout $u \in L$ **Faire**
            $\Delta \leftarrow \lambda(u) - \lambda(v)$
            $F_v(t) \leftarrow F_v(t) + (\Delta/\|\Delta\|) \times f_r(\|\Delta\|)$ force de répulsion
        Fin Pour
        $\lambda(v) \leftarrow \lambda(v) + temp \times F_v(t)$
    Fin Pour
Fin Pour

3.3.1.1 Optimisation

L'approche de dessin de graphe à partir de données multidimensionnelles a été plusieurs fois raffinée afin d'atteindre une complexité sub-quadratique. En 2002, puis en 2004, Morisson et Chalmers proposent une technique hybride [81], qui consiste d'abord à positionner un sous graphe astucieusement choisi de taille $\sqrt{(N)}$, appelé *noyau*, avec l'algorithme en $\theta(N^2)$, pour que ce pré-positionnement s'effectue en temps linéaire. Cette technique est illustrée par l'algorithme 4. Les sommets non positionnés sont ensuite placés autour d'un sommet proche appelé sommet *père*, processus illustré par la figure 3.9. La recherche de ce sommet père (ligne 4) est une opération d'une complexité de $\theta(\sqrt{(N)})$ dans l'algorithme de 2002, puisque le *noyau* contient $\sqrt{(N)}$ sommets ; complexité réduite en 2004 à $\theta(\sqrt{\sqrt{N}})$. Cette opération, effectuée pour chaque sommet, majore la complexité globale de l'algorithme à $\theta(N\sqrt{\sqrt{N}})$. Plus récemment, Jourdan et Melançon dans [69], ont proposé un algorithme plus rapide en $\theta(N \log N)$, en reprenant la notion de *noyau*, mais en appliquant une approche récursive multi-échelle.



(a) Extraction du *noyau* (en jaune)

(b) Dessin du *noyau*

(c) Positionnement autour du *noyau* (en jaune) des sommets restants

FIG. 3.9 – Optimisation de l'algorithme de dessin par extraction d'un *noyau* de $\sqrt{N}$ sommets

Choix du *noyau*

Nous proposons dans la suite une adaptation de ces optimisations à la structure d'octree, choisi dans l'algorithme précédent. On suppose que le graphe à dessiner est connexe. Pour la même raison que dans l'algorithme précédent (le graphe à dessiner n'est pas complet), le *noyau* ne peut s'extraire de manière aléatoire, car ceci ne garantit pas l'obtention d'un graphe connexe. Le graphe qui constituera le *noyau* doit être connexe pour assurer un positionnement global cohérent avec le graphe initial. Un graphe *noyau* composé de deux composantes connexes aboutirait à un placement indépendant des composantes connexes, spatialement éloignées par l'action des forces de répulsion. Il faut donc une méthode d'extraction de *noyau* qui garantisse la connexité de ce dernier. Par ailleurs, le dessin du *noyau* avec l'algorithme 3 de complexité quadratique, doit s'effectuer avec une complexité linéaire, ce qui impose une taille de $\sqrt{N}$ sommets. Ce *noyau* est extrait par un choix aléatoire d'un sommet du graphe, par un parcours du graphe à partir de ce sommet, en accumulant chaque sommet visité dans le *noyau*, jusqu'à ce que celui-ci contienne $\sqrt{N}$ sommets. Une autre alternative est celle proposée par Fricke dans [49], et introduit une force d'attraction centrale vers le barycentre du dessin, Elle permet d'éviter un éloignement trop important des composantes connexes, mais aboutit cependant à un positionnement relatif arbitraire de ces composantes.

Algorithme 4 Algorithme optimisé avec pré-positionnement d'un *noyau* en $\theta(N\sqrt{N})$

Entrées : Un graphe complet $G(V, E)$ avec une matrice de similarité notée l_{uv}
Sorties : Dessin λ de tous les sommets $u \in V$, de position λ_u

1: $Kern \leftarrow$ choisir aléatoirement $\sqrt{|V|}$ sommets de G
2: Dessiner le graphe $Kern$ avec l'algorithme 2
3: **Pour tout** $v \in V$ **Faire**
4: Trouver $s \in V_{Kern}$ le sommet le plus proche de v
5: Placer s près de v
6: **Fin Pour**
7: Appeler une iteration de l'algorithme 2 un nombre constant de fois

Positionnement des sommets restants

Une fois le *noyau* de taille $\sqrt{N}$ positionné (ligne 2) à l'aide de l'algorithme 3, il s'agit désormais de placer les sommets restant autour de ce *noyau*, le plus rapidement possible. Le goulot d'étranglement de l'algorithme optimisé de Chalmers est la recherche pour chaque sommet du sommet *père* le plus proche dans matrice de similarité, parmi les sommets du *noyau*. Dans notre cas, il n'y a pas de mesure de similarité entre tous les sommets. Un sommet proche est donc simplement pour nous un sommet adjacent. Cependant, chaque sommet non dessiné du graphe ne possède pas nécessairement un sommet du *noyau* dans sa liste d'adjacence (voir figure 3.9(a), le noeud en blanc). Nous avons donc pris le problème dans l'autre sens, en partant des sommets du *noyau*. Pour chaque sommet du *noyau*, on dessine un sous-arbre de taille constante (ligne 6). On dessine ainsi une partie des sommets. Ensuite, chaque sommets v qui n'a pas été dessiné, géré par l'ensemble R, constitue la racine d'un arbre de taille constante, dans lequel on va rechercher le premier sommet u déjà dessiné (ligne 9, appel de l'algorithme 6). On place alors v près de u (ligne 11). Cet algorithme ne garantit pas le placement de tous les sommets, et les sommets restants sont placés aléatoirement (ligne 13). Ce placement sera raffiné par plusieurs itérations de l'algorithme 3, un nombre constant d'itérations (ligne 17).

Algorithme 5 Algorithme avec gestion par octree et pré-positionnement de *noyau*

Entrées : Un graphe avec $G(V, E)$ avec des contraintes de longueur l_{uv}
Sorties : Dessin λ de tous les sommets $u \in V$, de position λ_u

1: $Kern \leftarrow ExtraireNoyau(G)$
2: Dessiner le graphe $Kern$ avec l'algorithme 3 précédent
3: $R \leftarrow V$, $prof \leftarrow constante$
4: **Pour tout** $v \in V_{Kern}$ **Faire**
5: $\quad R \leftarrow R - v$
6: $\quad DessinerSousArbre(v_{Kern}, prof)$
7: **Fin Pour**
8: **Pour tout** $v \in R$ **Faire**
9: $\quad u \leftarrow ChercherDansSousArbre(v, prof)$
10: $\quad$ **Si** $u \neq \emptyset$ **Alors**
11: $\quad\quad$ Placer v près de u, , $R \leftarrow R - v$
12: $\quad$ **Sinon**
13: $\quad\quad$ Placer v aléatoirement , $R \leftarrow R - v$
14: $\quad$ **Fin Si**
15: $\quad RemplirOctree(O, v)$
16: **Fin Pour**
17: Appeler une iteration de l'algorithme 3 un nombre constant de fois

Algorithme 6 Algorithme ChercherDansSousArbre

Entrées : Un sommet v, $prof$ la profondeur maximum de l'arbre
Sorties : Le premier sommet u dessiné du sous arbre de v de profondeur $prof$

```
1: Si prof ≠ 0 Alors
2:     retourne ∅
3: Fin Si
4: s ← ∅
5: Si v ∈ R Alors
6:     retourne v
7: Sinon
8:     Pour tout u ∈ P_v Faire
9:         s ← ChercherDansSousArbre(u, prof − 1)
10:        Si s ≠ ∅ Alors
11:            retourne s
12:        Fin Si
13:    Fin Pour
14:    retourne ∅
15: Fin Si
```

Algorithme 7 Algorithme DessinerSousArbre

Entrées : Un sommet v, $prof$ la profondeur maximum de l'arbre
Sorties : Dessin λ de tous les sommets du sous arbre de profondeur $prof$

```
1: Si prof ≠ 0 Alors
2:     Pour tout u ∈ P_v Faire
3:         Si u ∈ R Alors
4:             Placer u près de v
5:             R ← R − u
6:             AjouterOctree(O, u)
7:         Fin Si
8:         DessinerSousArbre(u, prof − 1)
9:     Fin Pour
10: Fin Si
```

3.3.1.2 Complexité

On ne garantit pas ici de positionner tous les sommets près d'un sommet du noyau, et c'est le raffinement sur l'intégralité du graphe qui permet de placer ces sommets correctement. Cependant nous avons constaté, que les nombre de sommets placés aléatoirement par cet algorithme, étaient étonnemment très réduit, sans avoir réussi à faire une relation entre ce phénomène et la structure du graphe. De toute façon, contenu du contexte interactif, le raffinement continue pendant toute l'interaction pour que le dessin soit réactif aux modifications du graphe. Cet algorithme constitue donc une étape rapide de pré-positionnement d'une complexité en $\theta(N \times p^k)$, où p est la taille maximum d'un ensemble minimal de voisin d'un sommet, et k la profondeur maximum d'un arbre de dessin et de recherche. En effet, après le positionnement du *noyau*, opération d'une complexité en $\theta(N)$, on parcourt tous les sommets du *noyau*, parcours d'une complexité en $\theta(\sqrt{N} \times p^k)$, et on dessine pour chaque sommet un sous-arbre de taille constante. Ensuite, ensuite pour chaque sommet non placé, on cherche dans un sous-arbre dont ce sommet est la racine, un sommet dessiné, opération d'une complexité maximum en $\theta(N \times p^k)$. La complexité de ce pré-positionnement est donc $\theta(N \times p^k)$, et il faut donc choisir une profondeur k très petite, 3 par exemple, qui garantisse que la taille des arbres ne dépasse pas N. L'avantage, c'est qu'on peut choisir de faire varier cette complexité selon la profondeur k et le nombre des voisins le plus proches p. Cette approche est une méthode hybride entre une méthode descendante (*top-down*) et ascendante (*bottom-up*).

3.3.1.3 Discussion

Nous avons remarqué que le choix du *noyau* influait fortement sur la rapidité du placement pour obtenir une qualité équivalente, selon la distribution des sommets choisis. Si ce *noyau* constitue un sous-graphe *noyau* trop localisé par rapport aux contraintes de distance, c'est à dire si son diamètre est trop petit, le positionnement autour de ce *noyau* est de moins bonne qualité qu'avec un *noyau* de diamètre plus grand. Un critère a donc été choisi pour maximiser ce diamètre sans augmenter la complexité de l'extraction. En extrayant le *noyau*, on parcourt et on intègre d'abord les sommets adjacents les plus éloignés en terme de similarité, en parcourant la liste d'adjacence triée à l'envers. Par ailleurs, maximiser le degré moyen du *noyau*, permet de maximiser le nombre de sommets non dessiné des sous-arbres de chaque sommet du *noyau*. Ceci diminue fortement la taille de l'ensemble restant à placer R, et diminue de fait la complexité globale. Un second critère serait donc d'intégrer en priorité dans le *noyau* les sommets qui ont le plus de voisins. Un *noyau* idéal serait un *noyau* qui optimise ces deux critères.

En ce qui concerne l'octree, dans l'algorithme 3 initial, la qualité du dessin peut être fortement améliorée en distribuant les appels de l'octree plutôt au début du

positionnement. En effet, à la fin du placement la variation de la position des sommets est moins forte, et reconstruire l'octree devient moins utile. La stabilisation du dessin est en effet fonction d'une température *temp* proportionnelle la somme des différences au carré de la longueur désirée pour chaque arête et de leur distance euclidienne réelle, comme chez Kamada et Kawai [71]. Une bonne approximation est une évaluation qui minimise cette somme. On utilise donc comme étalon la différence de température entre les deux premières itérations Δ_{max}, qui correspond à une probabilité d'appel de 1. Soit O_{max}, le nombre constant d'appels à l'octree, et $O_{restant}$ le nombre d'appels qu'il reste à effectuer, et Δ_{temp} la dernière variation de température, on appelle l'octree avec la probabilité P suivante :

$$P = \frac{O_{restant} \times \frac{\Delta_{temp}}{\Delta_{max}}}{O_{max}} \tag{3.1}$$

On appelle évidemment la construction de l'octree que si $O_{restant}$ est positif, pour garantir un nombre d'appels constant.

3.3.1.4 Dessin réactif à l'interaction

La force de cette méthode est sa nature itérative, qui permet de dessiner le graphe à chaque itération. Ce dessin itératif nous permet d'obtenir un rendu dynamique qui réorganise en permanence le graphe en fonction des ordres donnés par l'utilisateur. Ainsi, il peut suivre *de visu* l'impact sur l'auto-organisation du graphe d'une modification, d'un ajout, d'une suppression, d'une arête ou d'un sommet, ou d'un filtrage d'information. Ce processus ne s'arrête donc pas lorsqu'une une stabilisation est détectée. Par ailleurs la gestion par octree permet en temps interactif d'élaguer les zones non visibles pour l'utilisateur, ce qui permet de diminuer d'une part les sommets et les arêtes à dessiner, et d'autre part, de n'effectuer la simulation de force que sur les sommets visibles. Un autre avantage important de cette méthode, est son adaptation rapide aux dispositifs de retour de force disponible dans le cadre immersif, puisque comme pour les sommets, c'est aussi un ensemble de forces qui permet de spécifier le comportement de ce type de dispositif.

3.4 La navigation

A l'intérieur du graphe 3D, qui peut être volumineux, la première des interactions nécessaires à l'exploration est celle qui est associée à la navigation dans la scène. Les données visualisées sont parfois massives et bien souvent, l'utilisateur n'en visualise qu'une petite partie. C'est la raison pour laquelle nous avons deux modalités de navigation : la navigation locale, qui consiste pour l'utilisateur à se rapprocher d'une

donnée éloignée mais visible, et les déplacements globaux, qui permettent d'atteindre des données hors du champs visuel.

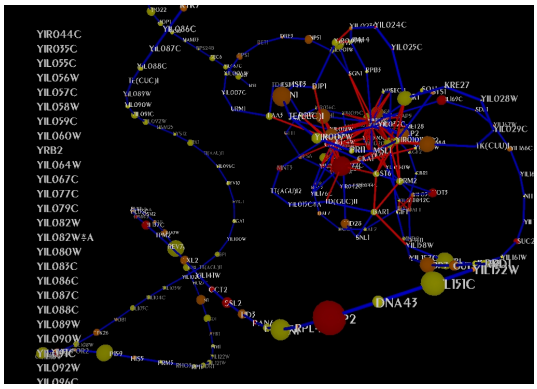
3.4.1 Les déplacements locaux

Les déplacements locaux s'effectuent par la sélection sur un sommet du graphe, arête, ou étiquette. Cette sélection s'effectue à l'aide de la souris 3D, dont l'orientation et la position sont captées par le système infrarouge du système *ArtTrack* décrit dans le chapitre 2. La caméra, métaphore de la fenêtre de visualisation sur la scène pour l'utilisateur, se déplace alors progressivement vers l'objet sélectionné, suffisamment lentement pour que l'utilisateur conserve son contexte de travail. En règle général, il est apparu que cette modalité de navigation est utilisé d'une manière particulière, par saut successif d'une données à l'autre, comme navigation par saut de puce.

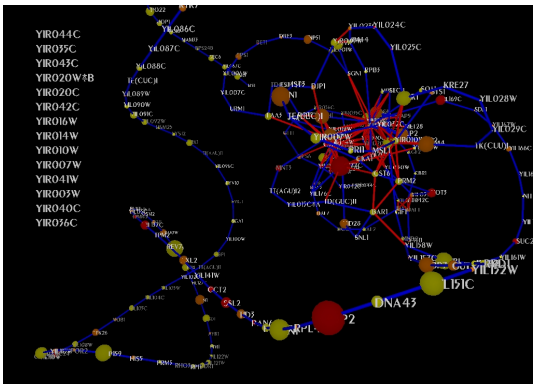
3.4.2 Les déplacements globaux

Les déplacements globaux s'apparentent à une recherche puis à un déplacement. La sélection est en effet impossible puisque les objets d'intérêt sont hors du champs visuel. La recherche est effectuée à partir des étiquettes visuelles des objets du graphe. Cependant il est hors de question d'utiliser un clavier, qu'il soit réel ou virtuel, pour rechercher ces étiquettes. Il est peu pratique et très intrusif lors d'une tache d'exploration d'avoir en permanence un clavier proche de l'utilisateur. D'autre part, l'utilisation d'un clavier virtuel est d'une efficacité très réduite.

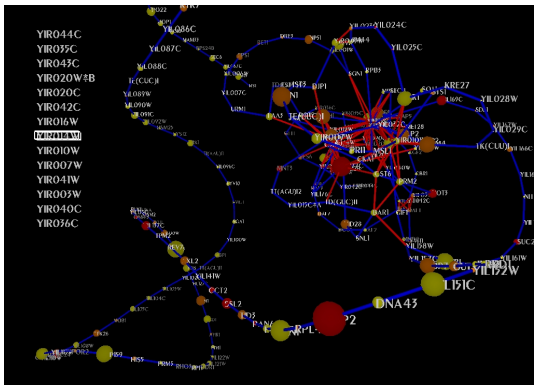
Nous avons développé une technique utilisant la reconnaissance vocale et la sélection afin de minimiser le nombre des interactions. L'utilisateur épelle la première lettre de l'étiquette de l'objet recherché. La liste des étiquettes commençant par cette lettre est affichée sur un côté de l'écran. L'utilisateur a alors le choix de continuer à épeler pour diminuer la liste, ou de sélectionner avec la souris 3D l'étiquette recherchée dans la liste, comme illustré dans la figure 3.10. L'utilisation conjointe de ces deux modalités permet de rendre cette recherche très efficace, puisque même sur plusieurs milliers d'objets, le nombre de lettres épelées ne dépasse rarement trois avant sélection.



(a) Passage en mode recherche, affichage de la liste des sommets dont les étiquettes commencent par le caractère prononcé



(b) Diminution dynamique de la liste en fonction des caractères prononcés



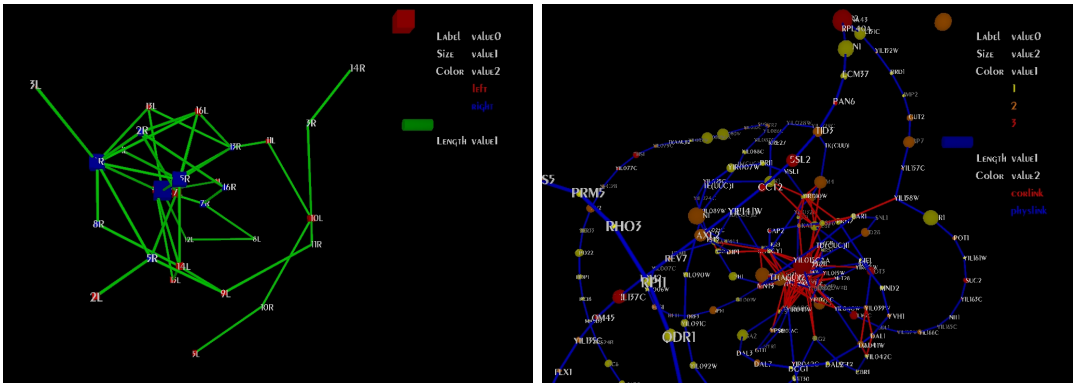
(c) Désignation de l'objet désiré

(d) Mouvement de caméra vers cet objet

FIG. 3.10 – Processus de navigation globale vers un sommet éloigné ou caché

3.4.3 La légende

Un retour d'information important lorsqu'il s'agit de gérer des données génomiques est de rappeler à l'utilisateur la sémantique des valeurs associées aux caractéristiques graphiques de chaque objet dans la représentation (cf. figure 3.11). Par ailleurs, la légende permet aussi d'effectuer d'autres opérations qui seront décrites dans les sections suivantes, comme l'ajout ou la modification d'attribut.



(a) Graphe de duplications (b) Graphe de co-régulations

FIG. 3.11 – Affichage avec la légende à droite de la sémantique des attributs graphiques

3.5 Un filtrage interactif des données

Une interaction importante pour diminuer la surcharge visuelle due à la complexité d'une masse importante d'information, est d'appliquer des filtres visuels par rapport à un type d'objet, où une caractéristique. Dans une interface de bureau, l'interaction classique consiste par exemple à configurer un filtrage à l'aide de boutons radio ou de boutons de sélection, qui permettent de sélectionner les types d'information à afficher.

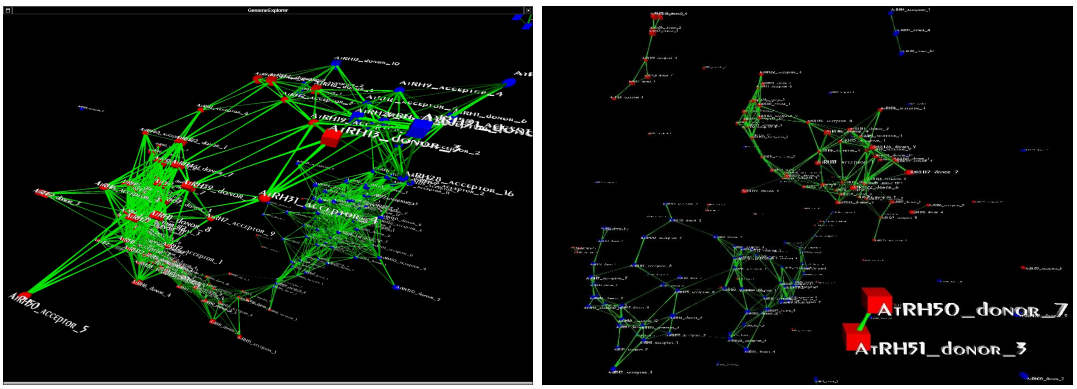
3.5.1 Filtrage sélectif

Un paradigme alternatif est intéressant en 2D pour visualiser des informations géographique variées, par exemple, topographiques, routières, pluviométriques, touristiques, sur une seule et même carte. A la place de boutons radio, un filtre est représenté par une métaphore de film semi-transparent, positionnable sur la carte. Plusieurs filtres peuvent se superposer ou se positionner côte à côte, pour visualiser seulement un type informations de la carte, dans la zone sur laquelle le film est positionné. Ce paradigme a été adapté au cadre immersif, selon les modalités suivantes :

le film semi-transparent est associé à un filtrage d'une caractéristique biologique sélectionnable depuis la légende. Ce film est attaché à l'avatar de la main contrôlé par le gant de données, de la manière suivante : le plan de ce film est formé par l'index et le pouce de l'avatar. L'activation de ce film est déclenché lorsque l'ouverture entre le pouce et l'index est maximal, c'est à dire quand le pouce et l'index sont écartés d'un angle de plus de 75 degrés.

3.5.2 Un filtrage par seuillage interactif

Un autre filtre permet de filtrer les arêtes par rapport à leur longueur. Toute les arêtes d'une longueur supérieure à celle choisie, deviennent invisibles et ne sont plus intégrées dans le processus dynamique de positionnement des sommets. Ce filtrage n'est pas directement effectué sur la longueur des arêtes dans l'espace euclidien, mais sur la mesure de similarité, reliée à cette longueur. Cette interaction est particulièrement intéressante, puisqu'elle permet de simplifier des graphes parfois complexes et difficilement lisibles, en sous-graphe plus petits, dans lesquels seules les arêtes les plus significatives sont conservées. Par exemple, dans un graphe représentant par un réseau d'alignement de séquences protéiques (sommets), le score associé à chaque alignement (arête), représenté par la longueur de ces arêtes, permet de refléter l'organisation dans l'espace les séquence protéines. Ce filtrage permet de ne conserver que les meilleurs alignements, et d'ignorer les moins significatifs.



(a) Avant un filtrage sur la longueur des arêtes (b) Après un filtrage sur la longueur des arêtes

FIG. 3.12 – Répartition spatiale de séquences nucléiques d'introns-exons selon un critère de courbure

3.6 La modification des données et l'annotation

Un besoin fort des biologistes lorsqu'il s'agit de gérer leur données est de pouvoir les annoter. Dans le cadre de graphe 3D, les opérations de base sont la mise à jour d'informations relatives à un sommet ou une arête, l'ajout d'un sommet ou d'une arête, et leur suppression.

3.6.1 La mise à jour, l'ajout, la supression

La suppression d'une arête s'effectue par sélection de celle-ci puis par commande vocale. La suppression d'un sommet est effectuée de la même façon, cependant, toutes les arêtes associées à ce sommets sont elles aussi supprimées.

La mise à jour est une modification d'un des attributs d'un objet. L'objet dont on veut modifier une valeur est sélectionné dans le graphe, puis une commande vocale de modification déclenche l'opération. La valeur à modifier de l'objet concerné est sélectionnée dans la légende. Lorsque cet attribut est une valeur qualitative, elle est mis à jour en sélectionnant la valeur désirée parmi la liste des valeurs disponibles dans la légende. Lorsque l'attribut est une valeur quantitative, un *slider* (jauge) réactif à la sélection par la souris 3D, permet de choisir une valeur numérique.

Pour l'ajout d'un sommet, le processus est identique à la mise à jour, à ceci prêt que la commande vocale est une commande d'ajout. Un sommet est ensuite affiché au centre de la scène, avec des valeurs par défaut, qui peuvent être modifiée par les modalités précédemment décrite. De même, lorsqu'il s'agit d'une arête, excepté le fait que les deux sommets entre lesquels on veut ajouter l'arête sont au préalable sélectionnés avant d'envoyer la commande vocale d'ajout.

3.6.2 Le problème du "Text Input" dans un cadre immersif

Lorsqu'il s'agit d'entrer des données de nature textuelle, comme dans le cas des étiquettes, ou de l'ajout d'annotations, l'utilisation d'un clavier est en général difficilement envisageable dans un cadre immersif pour les raisons suivantes :

- la posture souvent debout dans une simili-CAVE n'est pas compatible avec les claviers classiques.
- Le fait de quitter des yeux la scène virtuelle rendue en stéréoscopie, pour regarder l'objet réel que constitue le clavier est désagréable
- L'ambiance lumineuse est souvent assez sombre, ce qui rend la saisie sur clavier difficile.

Ce problème, identifié dans la communauté de la réalité virtuelle par le problème du "text input", est un problème en soi, et fait l'objet de nombreux travaux [34][113]. Lorsque la contrainte de ne pas utiliser de clavier est forte, une solution possible

est d'utiliser le clavier virtuel. Une autre solution est la reconnaissance vocale de termes, mais l'absence de systèmes vraiment efficaces et suffisamment entraînés à reconnaître des termes concernant les données génomiques, rend son utilisation très peu efficace. Faute de mieux, quand il n'était pas possible de faire autrement, nous avons donc utilisé le clavier, pour entrer des informations textuelles non prédéfinies.

Une solution idéale à mettre en place serait d'enregistrer ces annotations textuelles de façon vocale, de traiter ces annotations *a posteriori* par un processus automatique de reconnaissance de la parole, pour les transformer en annotations textuelles, afin de pouvoir les analyser ultérieurement dans un contexte non immersif. En effet, la présentation des annotations sous forme de restitution vocale dans un cadre non immersif est évidemment exclue. Cependant, la restitution de ces annotations dans une exploration ultérieure des données dans un contexte immersif, pourra être sous forme sonore, en restituant les enregistrements vocaux, ou visuels, par l'intermédiaire d'étiquettes 3D, au choix de l'utilisateur.

3.7 Sonification de données

Lorsque le canal visuel est surchargé, il est utile de faire appel un autre canal sensoriel, pour transmettre des informations afin de diminuer cette surcharge. Il est particulièrement intéressant, dans un dispositif de réalité virtuelle, d'utiliser l'ouie. En effet, l'être humain, perçoit le son de façon spatialisée, au même titre que la vue.

Dans ce cadre, une des interactions intéressante est une métaphore du sonar, que nous avons appelé sonar virtuel. L'utilisateur pointe un sonar dont le référentiel est attaché à la souris 3D, de forme cylindrique et transparent, et dont la longueur et la largeur sont réglables. Ce sonar permet d'englober un objet ou un groupe d'objets dans l'espace. Plusieurs alternatives de rendu sont alors possibles. Une caractéristique qualitative d'un sommet du graphe, peut, au même titre que pour les attributs visuel, être associée à un son. On peut alors rendre uniquement le son correspondant à la caractéristique majoritaire dans l'ensemble d'objets pointés par le sonar. Une autre alternative a été testée, l'amplitude du son correspondant à chaque caractéristique est proportionnelle à la distribution de chaque caractéristique dans l'ensemble des objets atteint par le sonar. Ce retour d'information sonore est intéressant car il permet de transmettre une information en plus par sommet.

3.8 Une architecture logicielle pour visualiser les données massives

La nature des très simple des composants graphiques de base du graphe 3D (sphère, ligne, cylindre, paraboloïde hyperbolique) permet d'en afficher quelques milliers sans poser de problème de performances purement graphiques. Plusieurs milliers de sommets et d'arêtes peuvent être affichés en temps interactif avec stéréoscopie.

Pour les graphes plus massifs, à partir d'une dizaine de milliers de sommets et d'arêtes, jusqu'à quelques millions de sommets et d'arêtes, la gestion des données graphiques est plus problématique, en terme de stockage dans la mémoire mais aussi de performance dans un contexte immersif. Une architecture spécifique est nécessaire, avec une machine pour gérer le graphe et le placement, fournisseur de données graphiques à une machine pour gérer la scène (figure 3.13). Cela permet, d'une part, de permettre le calcul *offline*, c'est à dire sans affichage, du placement du graphe, et d'autre part, de n'envoyer au client graphique que les données perceptibles par l'utilisateur, sa position par rapport aux objets dans la scène étant connue.

En réponse à une requête du client graphique, le serveur n'envoie que la liste des objets qui se trouvent inclus dans une sphère de vision dont le centre est la position de l'utilisateur dans la scène, sphère symbolisant les limites pour lesquels les objets ne sont plus visibles, car trop éloignés de l'utilisateur. Ainsi, à chaque translation de l'utilisateur, le client graphique émet une requête de mis à jour au serveur de données en lui envoyant sa position, et le serveur de données lui transmet la position des sommets et des arêtes du graphe inclus dans la sphère. Le test qui permet d'extraire uniquement les sommets à l'intérieur est grandement accéléré par les gestions de l'espace par octree, utilisé dans les algorithmes de positionnement coté serveur.

Avec cette architecture, deux modes d'exploration sont possibles :

- le mode statique, dans lequel le placement des sommets de graphe a été effectué offline au préalable de l'exploration. Dans ce cas, les sommets et les arêtes du graphe ont une position fixe. Dans ce mode, une optimisation est effectuée en n'envoyant au client que les objets qui entrent dans cette sphère. Le client détruit ainsi les objets qui sortent de la sphère, conserve et affiche ceux qui restent, et ne construit graphiquement que ceux qui entrent dans la sphère.
- le mode dynamique, dans lequel le placement des sommets de graphe est observé en temps interactif. L'avantage de ce mode est que la suppression ou l'ajout de sommets et d'arêtes dans le graphe, induit une modification topologique du graphe, puisque le placement est calculé en permanence. Dans ce mode, le client envoie au serveur une requête de mise à jour à intervalles réguliers, en ms, configuré par l'utilisateur. L'optimisation du mode statique n'est

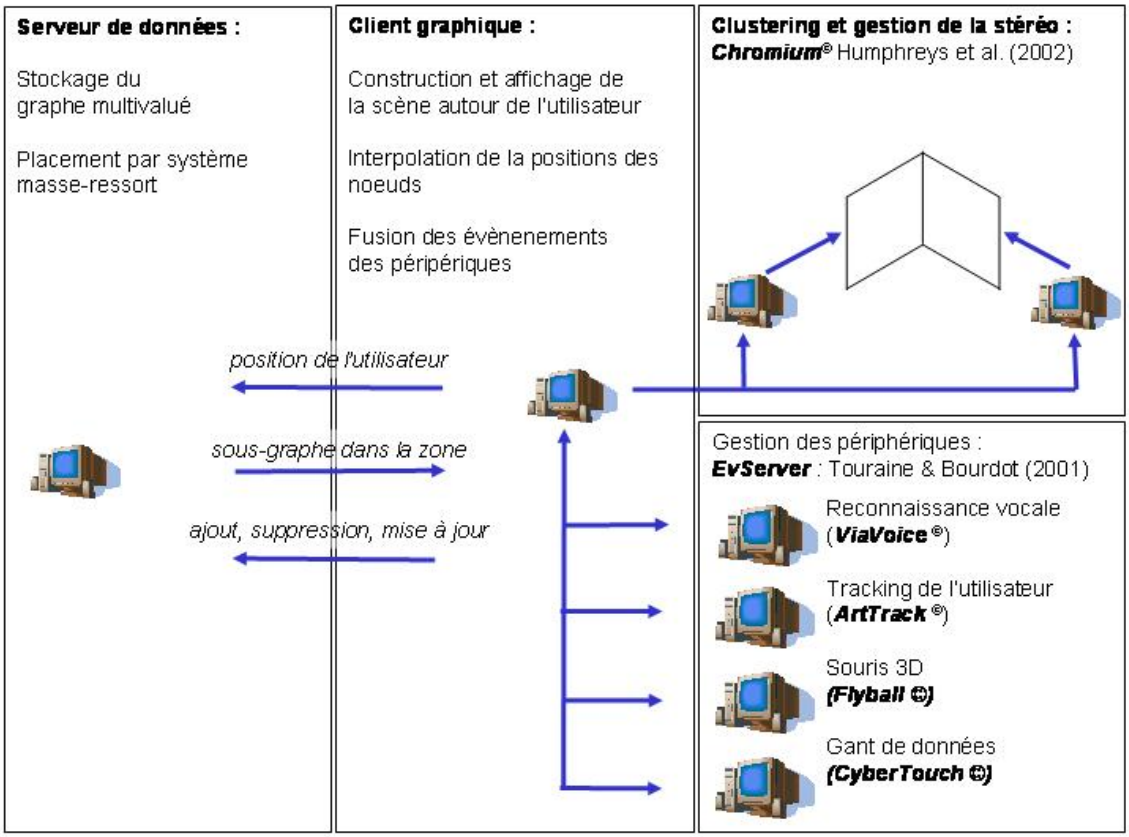


FIG. 3.13 – Architecture de Genome3DExplorer

pas possible puisque la position des sommets est variable, le serveur répond donc par la liste de tous les objets dans la sphère. Cependant, le client ne modifie que la position des objets graphiques dans le cas ou ils existent déjà. Pour éviter les changements brusques de positions des objets graphiques côté client, une interpolation linéaire des positions des sommets du graphe est effectuée entre chaque réception.

3.9 Interopérabilité et retour d'information

Nous avons vu au début de ce chapitre que la représentation informatique des données que nous avons choisi, était un graphe multivarié. Cette structure reflète à la fois la structure sous-jacente des bases de la plupart des bases de données génomiques factuelles, mais correspond aussi à la structure des résultats issus de

l'extraction d'informations à partir des données bibliographiques textuelles. Nous avons aussi vu dans les sections précédentes les modalités de traduction de ce graphe multivarié, vers une représentation informatique. Nous avons défini deux couches correspondantes la couche *donnée* qui est une représentation du contenu des données à analyser, la couche *représentation* qui permet à l'utilisateur de choisir la modalités de représentation de ces données.

Nous avons défini une troisième couche, la couche *sémantique*, fournissant des méta-informations sur les données. L'objectif de cette couche est de faciliter l'interopérabilité semi-automatique avec des bases de données externes (cf. figure 3.15). Le problème de l'interopérabilité et plus généralement de l'intégration de données hétérogènes est un problème en soit [57]. Cette couche permet d'associer aux données du graphe multivarié, des méta-informations, comme la nature de leur identifiant unique (ici *private*), c'est à dire un identifiant propre à des données privées ne correspondant à aucune autre base de données génomiques, ou bien publique s'il est décrit dans une base, leur nom (*name*), leur type d'un point de vue bioinformatique (*boitype="DNA Sequence"*). Elle permet aussi de décrire la nature de chaque valeur des vecteurs associés aux objets et aux relations (*is_ a="chromosomename"*), et de leur donner un type (*domain="float"* ou *domain="enumleft,right"*).

Cette description permet l'interopérabilité car elle peut être être traduite automatiquement par une requête afin d'interroger des plates-formes fournissant des services et des données génomiques, comme *GeneBank*, *SRS*, ou *GenoMEDIA*. Le résultat d'une requête, émise dans un format spécifique à la base interrogée, peut lui aussi être traduit automatiquement en données de graphe multivariée, fournit en entrée à notre application d'exploration immersive 3.15. Cette approche a été testée et validée dans l'expérience sur les données d'expression dans le chapitre suivant, en intégrant des données de puces à ADN, augmenté d'information externes fourni par la plate-forme d'intégration de données génomiques *GenoMEDIA* [57].

Par ailleurs, le typage des objets bioinformatiques qui constituent les sommets du graphe (niveau *overview* selon Shneiderman) est nécessaire afin de rendre un retour d'information visuelle adapté, lorsque l'utilisateur sélectionne un sommet dans le but d'obtenir une description détaillée de cet objet (niveau *zoom* selon Shneiderman [95]). Ce typage bioinformatique rend possible la configuration de l'application à utiliser pour obtenir une représentation visuelle de cet objet, en fonction de sa nature bioinformatique : séquences d'ADN (DNA sequence), séquences d'ARN (RNA sequence), séquences protéiques (protein sequence), structures protéiques 3D (PDB), en utilisant des applications de visualisation tierces. Dans ce processus, l'utilisateur ne définit que la nature des données à représenter (couche sémantique), et les modalités de représentation de ces données (couche représentation).

```
<semantic>
  <object uid="private" is_a="chromosomesequence"
          biotype="DNA sequence" organism="Yeast">

    <value0 is_a="chromosomename" domain="string"/>
    <value1 is_a="chromosomeside" domain="enum{left,right}"/>
    <value2 is_a="chromosomesize" domain="float"/>
  </object>

  <relation>
    <value0 is_a="commonblockscore" domain="float"/>
  </relation>
</semantic>
```

FIG. 3.14 – Description de la nature des données à analyser

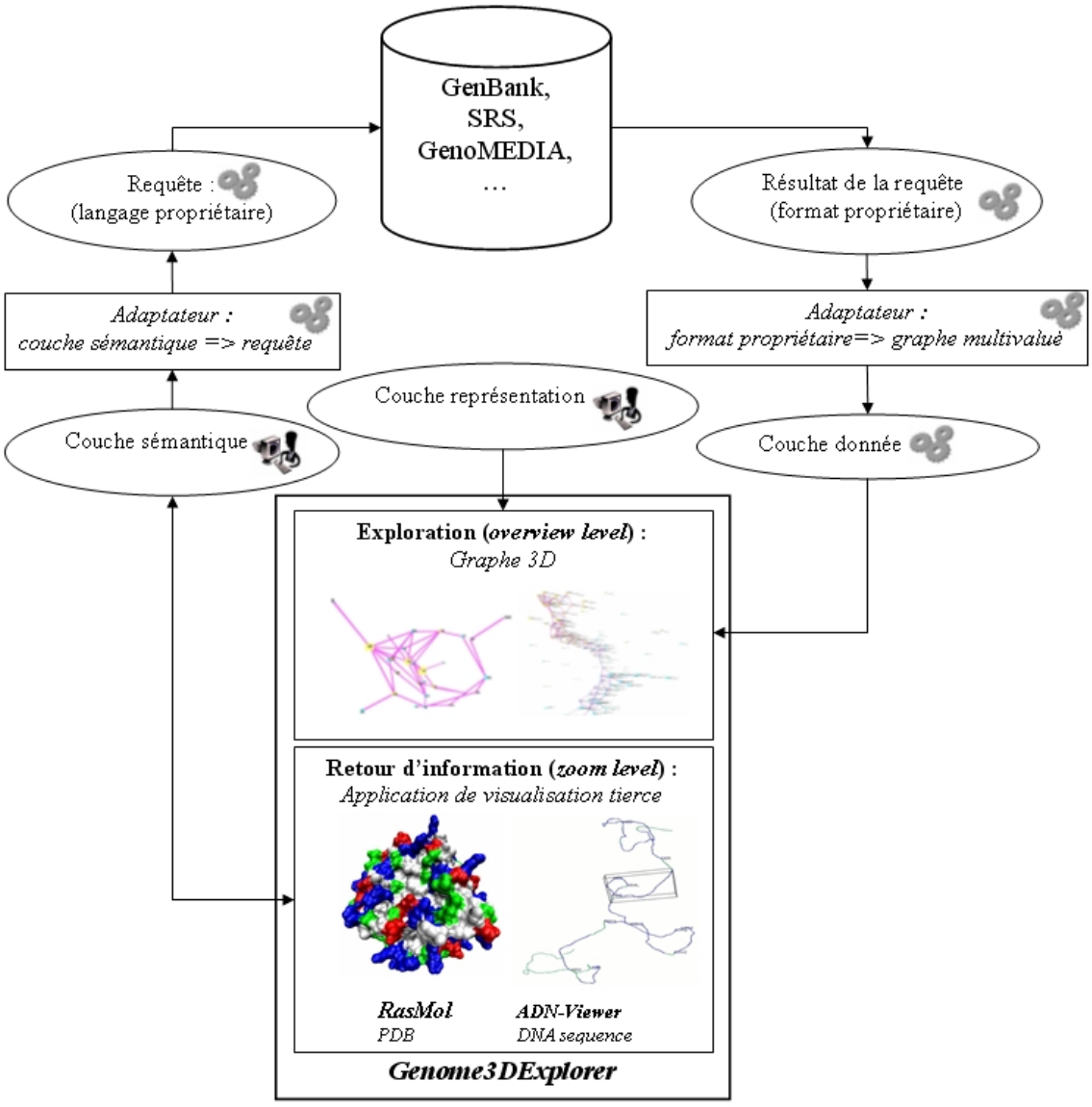


FIG. 3.15 – Interopérabilité entre des fournisseusrs de données génomiques et *Genome3DExplorer*

3.10 Conclusion

Nous avons défini dans ce troisième chapitre nos choix en terme de conception du paradigme d'exploration et de visualisation, avec trois couches différentes. La couche *donnée* définit les données relationnelles multivariées, constituées d'un ensemble d'objets multivariés, et d'un ensemble de relations binaires multivariées entre ces objets. La couche *représentation* permet de choisir les modalités de traduction des données par des caractéristiques visuelles, sonores ou haptiques. Enfin, la couche *sémantique*, permet de décrire la nature biologique des données à explorer, et rend possible l'interopérabilité et l'interrogation de banque de données génomiques extérieures, comme *GenoMEDIA* [57], *GeneBank* [2] ou *Swissprot* [15]. Cette séparation entre les couches *donnée* et *representation* a grandement participé à augmenter l'utilisabilité de *Genome3DExplorer*, dans la mesure où elle permet sur de mêmes données, de disposer de plusieurs représentations, ce qui permet de tester rapidement avec l'utilisateur laquelle de ces représentations est la plus adaptée. Elle a permis d'intégrer des données externes grâce à la couche sémantique, comme nous le verrons dans le chapitre suivant.

Nous avons par ailleurs proposé une adaptation de l'algorithme de Chalmers [36]. Cette approche est conçue pour dessiner des données multidimensionnelles sous forme de graphe. Notre adaptation permet de dessiner des graphes sans topologie particulière, possédant des contraintes de longueurs sur ses arêtes, grâce à l'utilisation d'un octree pour gérer l'espace. Nous avons ensuite optimisé cette méthode en s'inspirant des méthodes d'optimisation par extraction de *noyau*. L'utilisation d'octree permet de diminuer la complexité du phénomène de répulsion, mais permet aussi d'optimiser la recherche des sommets du graphes visibles pour l'observateur coté serveur, et de n'appliquer qu'à ces sommets la simulation des forces. Cette approche permet de gérer la réactivité du dessin aux modifications en temps interactif.

Cette méthode de dessin se prête particulièrement aux graphes peu connectés, ce qui est le cas de la plupart des graphes construits à partir des données biologiques. Pour illustrer, dans le pire des cas où les données sont très volumineuses, le *Decrypthon* (plusieurs Go), présenté dans le chapitre 4, est constitué d'un réseau de 600 millions d'alignements (arêtes), effectués sur 500000 protéines (sommets). Le graphe que constitue cette banque est donc assez creux, malgré le degré moyen de ce graphe, ainsi que sa taille, qui restent trop élevés pour être intégralement dessiné par les méthodes classiques de dessin de graphe, qui serait, s'il était dessiné, trop complexe pour être exploité en l'état.

Enfin, nous avons décrit quelques paradigmes d'interaction développés dans le contexte immersif, dont le filtrage dynamique sur la longueur des arêtes, qui permet de simplifier le graphe pour qu'il soit exploitable, en conservant les relations biologiques entre les objets d'intérêt les plus significatives.

Chapitre 4

Applications et résultats

La connaissance s'acquiert par l'expérience, tout le reste n'est que de l'information

A. Einstein

L'expérience est une observation provoquée dans le but de faire naître une idée

C. Bernard

Dès le début de ce travail, l'utilisateur final a toujours été mis au centre la boucle de conception, grâce à l'utilisation sur place d'un prototype, afin de tester et de valider les paradigmes de visualisation et d'exploration précédemment définis. Ceci a permis de préciser, au fur et à mesure leurs besoins et de leurs usages, et de confronter cette méthode d'exploration à des données biologiques. Plusieurs expériences d'exploration de données génomiques variées ont donc été menées en collaboration avec des biologistes, décrites dans le quatrième et dernier chapitre de cette thèse.

4.1 Analyse de l'architecture des génomes

Nous présentons dans cette section, trois expériences d'exploration de données ayant pour but de mieux comprendre l'organisation des génomes. Nous avons étudié grâce à *Genome3DExplorer*, associé parfois à d'autres outils en amont comme *ADN-viewer* dédié à l'analyse de l'architecture des génomes. Cette section présente l'étude avec *Genome3DExplorer* trois phénomènes biologiques, qui organisent et structurent les génomes : les duplications et les co-régulations chez *Saccharomyces cerevisiae*, et l'épissage chez *Arabidopsis thaliana*. Ces explorations et leurs analyses ont été effectuées en collaboration avec des biologistes et des bioinformaticiens.

4.1.1 Analyse des duplications interchromosomiques chez la levure

Une des premières exploration de données avec *Genome3DExplorer* a été effectuée afin d'analyser les duplications de séquences inter-chromosomiques chez la levure. La levure est un eucaryote dont le patrimoine génétique est porté par 16 chromosomes. Au cours du temps, ces chromosomes ont échangé des portions de leur séquence d'ADN. Certains de ces chromosomes partagent aujourd'hui des sous-séquences très similaires, voire identiques. La figure 4.1 est une carte partielle des duplications chez 5 des 16 chromosomes de la levure. Le nombre en chiffre romain est le numéro du chromosome, les rectangles colorés représentent les blocs dupliqués identifiés par un numéro et une couleur. Enfin, les annotations verticales au dessus de chaque bloc sont les noms des gènes appartenant à ce bloc, et qui existent dans d'autres blocs. On peut observer que le chromosomes I et VIII partagent un bloc en rouge identifié par le numéro 1, que le IX et le V partagent un bloc bleu identifié par le numéro 27. Ces données proviennent du travail de l'équipe de Wolfe sur les duplications [112], et la carte complète de ces duplications est fournie en annexe C.

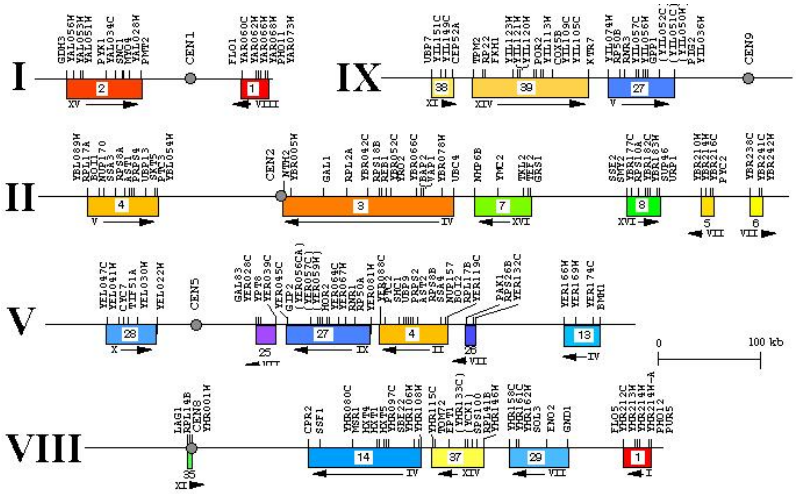


FIG. 4.1 – Visualisation usuelle des blocs dupliqués dans 5 des 16 chromosomes de la levure

Les données brutes traitées dans cette analyse sont donc constituées d'un ensemble de 32 séquences, les bras gauche et droit des 16 chromosomes de la levure, et d'une mesure normalisée de duplication entre chaque couple de séquences. La synthèse de ces données est le fruit du travail de Justine Merle effectué durant son stage de licence, et encadrée par Oriane Matte-Tailliez. L'objectif de la visualisation de ces données est d'obtenir une vision synthétique des échanges inter-chromosomiques du patrimoine génétique de la levure, et complémentaire à la représentation tradition-

nelle 4.1, qui permet difficilement d'appréhender l'organisation globale des échanges inter-chromosomiques.

Comme dans la carte partielle 4.1, l'objet d'intérêt est ici la séquence chromosomique, représentée par un sommet, et la relation qui relie ces objets est la mesure de duplication. Dans le graphe, l'utilisateur a choisi de représenter ces 32 séquences par des cubes, dont la couleur représente le côté de la séquence sur le bras droit (en bleu) ou gauche (en rouge) du chromosome. Cette mesure de duplication inter-chromosome a été associée à la longueur des arêtes qui relient les sommets, de sorte que si deux chromosomes ont en commun un nombre important de sous-séquences, ils seront spatialement proches, dans la représentation tridimentionnelle fournie par *Genome3DExplorer* dans la figure 4.2.

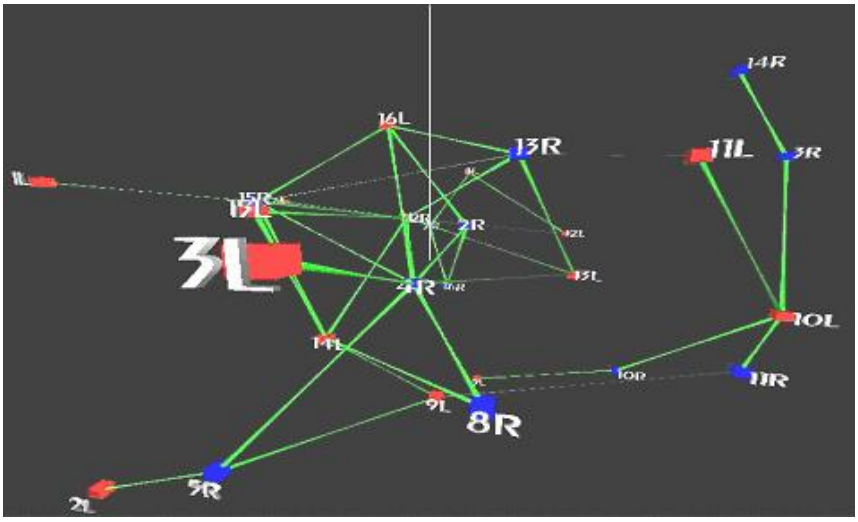


FIG. 4.2 – Visualisation synthétique des duplications entre les 32 séquences portées par les bras gauche et droit des 16 chromosomes de la levure : chaque cube représente la moitié d'une séquence de chromosome, identifiée par une étiquette qui donne le numéro du chromosome, en bleu la moitié droite, en rouge, la moitié gauche - une arête entre deux séquences signifie que les chromosomes partagent des séquences communes, et la longueur de cette arête est inversement proportionnelle au nombre de duplications

Cette représentation des données est complémentaire à celle utilisée dans la figure précédente 4.1. En effet, si cette carte des duplications offre une vision assez détaillée des blocs dupliqués chromosome par chromosome, un parcours séquentiel bloc par bloc est nécessaire afin d'identifier la quantité de patrimoine partagé dans chaque couple de chromosomes. Dans notre représentation synthétique des échanges inter-chromosomiques (figure 4.2), on peut directement constater que certaines séquences sont au centre des échanges comme *4R*, alors que d'autres sont en périphérie, comme pour *14R*.

De nouveaux questionnements peuvent ainsi émerger à partir de cette représentation des données. Par exemple, existe-t-il un rapport entre le positionnement relatif des chromosomes dans la cellule et les échanges inter-chromosomiques qu'ils ont subi, d'autant plus que des études récentes ont montré que la position relative des chromosomes est très conservée après la division cellulaire [54].

Enfin, l'aspect synthétique de cette visualisation des échanges inter-chromosomiques, permet d'envisager une comparaison visuelle très rapide entre deux souches différentes de levure, en supposant que la mesure de duplication entre chaque séquence soit assez sensible pour mettre en relief les différences de duplication entre deux souches, et que le positionnement initial des sommets du graphe soit effectué de manière déterministe.

4.1.2 Caractérisation des introns/exons chez *Arabidopsis thaliana*

Un autre phénomène biologique, appelé épissage, est étudié dans cette section. Certains gènes codants des eukaryotes supérieurs qui aboutiront par transcription puis traduction à une protéine, sont constitués des sous-séquences, nommées introns et exons. En effet, après le processus de transcription, puis de traduction qui mène du gène à la protéine, les biologistes ont constaté qu'il y avait des trous dans les séquences protéiques, c'est à dire que certaines parties de la séquence correspondant à un gène, n'apparaissaient pas dans la séquence protéique. Ils en ont déduit que certaines parties d'un gène sont traduites, (exons) et d'autres non (introns). Ce phénomène est appelé épissage, et est illustré par la figure 4.3.

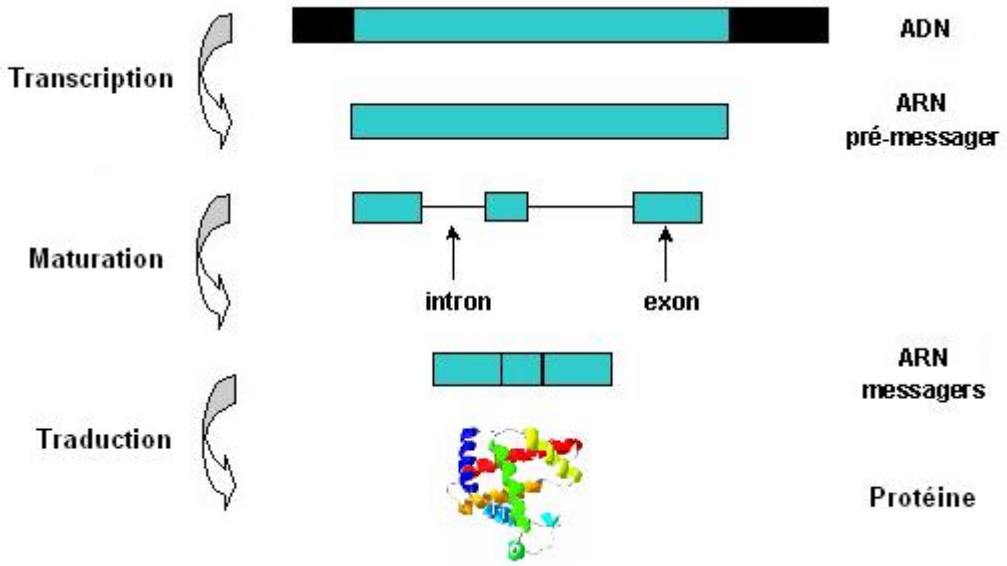


FIG. 4.3 – Processus et types de séquences impliqués dans l'obtention d'une protéine à partir de la séquence d'ADN d'un gène

4.1.2.1 Objectif

L'objectif de cette expérience est de savoir si une caractérisation des sites d'épissage chez *Arabidopsis thaliana* peut être effectuée en considérant des critères locaux de courbures de la double hélice que constitue la séquence d'ADN. Ce travail a été mené en collaboration avec Claire Toffano-Nioche, Foued Ben Hadjali et Joan Hérisson.

4.1.2.2 Données

Durant le processus qui, à partir d'une séquence d'ADN, aboutit à la production d'une protéine, seuls les exons sont traduits en éléments qui constituent la protéine. Discriminer les introns des exons constitue une problématique biologique importante. Plusieurs méthodes permettent d'étudier la structure de telles séquences, et utilisent les données expérimentales. Quand les données expérimentales ne sont pas disponibles, on utilise des outils logiciels de prédiction [60] [80]. Cependant ces méthodes sont dépendantes de la définition de la structure des gènes durant la phase d'annotation d'un génome [83] [76] [40], c'est la raison pour laquelle il est important d'apporter d'autres critères d'annotation, afin d'améliorer l'efficacité des prédictions *in silico*.

Le jeux de données de cette étude est constitué de séquences d'introns et d'exons centrées autour des sites d'épissage de gènes d'*Arabidopsis thaliana*, le site d'épissage étant la frontière entre un intron et un exon. A partir de son génome complet provenant de *AGI* [20], nous avons défini deux ensembles de données concernant les sites d'épissage. Nous avons extrait deux types de séquences centrées sur ces sites d'épissage : les séquences dites "donneurs", (exons puis introns), et les séquences dites "accepteurs" (introns puis exons).

4.1.2.3 Méthode

Ces deux ensembles de séquences concernant les sites d'épissage, sont envoyés à l'outil *ADN-viewer* afin de calculer leur trajectoire tridimentionnelle à partir de leur séquence textuelle d'ADN. Une fonction de courbure intrinsèque est ensuite calculée à partir de la trajectoire de chaque séquence. Ensuite, une métrique, basée sur la corrélation de *Pearson* qui permet d'évaluer la similarité de courbure entre deux séquences, permet de regrouper, avec une approche de type *K-Means*, en donnant un score de comparaison entre les courbure de séquences. La corrélation de *Pearson* utilisée pour comparer deux fonctions discrètes de t échantillons F et G de courbure deux séquences est la suivante :

$$cor(F,G) = \sum_{1 \leq t \leq N} \frac{(F(t) - \bar{F})(G(t) - \bar{G})}{\mu_F \mu_G}$$

En parallèle, cette métrique a été associée de manière inversement proportionnelle, à la longueur des arêtes du graphe, chaque sommet de ce graphe représentant une séquence "donneur" ou "accepteur", afin d'être explorée avec *Genome3DExplorer*. La chaîne des processus de traitement des données est illustrée dans la figure 4.4.

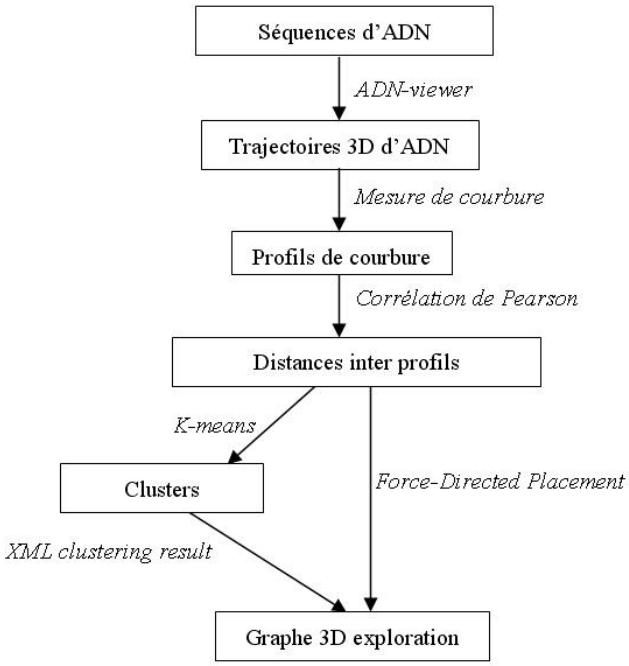


FIG. 4.4 – Etapes de traitements des données brutes à la visualisation

De la séquence textuelle à la trajectoire tridimensionnelle

L'ADN est généralement étudié en tant que séquence primaire (succession de A, de C, de G et de T). Une nouvelle façon d'aborder cette molécule est de compléter ces informations textuelles par une représentation de la conformation spatiale de cette séquence calculée *in silico*. Cette conformation spatiale est une trajectoire tridimentionnelle que constitue par la molécule d'ADN nue. Cette représentation peut être calculée et visualisée grâce à l'outil *ADN-viewer* [56][64], à différentes échelle, atomique, génique, ou chromosomique. Le principal intérêt de cette approche est que deux séquences textuelles d'ADN différentes peuvent avoir une conformation spatiale proche. L'opération de construction de la conformation spatiale de l'ADN à partir de la séquence étant donc une opération non bijective, il devient intéressant de mener des études pour découvrir d'éventuelles relations entre la courbure intrinsèque de l'ADN et certains phénomènes biologiques, comme l'épissage dans notre cas.

Extraction du profil de courbure d'une séquence

A partir du calcul de la trajectoire, nous avons défini une mesure reflétant la notion de courbure intrinsèque de l'ADN, mesure qui sera notre critère pour classifier les données. Beaucoup de mesures permettent de calculer le degré de courbure d'une trajectoire tridimentionnelle. Nous en avons choisi une basée sur l'abscisse curviligne de la trajectoire. Pour chaque coordonnée représentant une paire de bases P_i, on calcule son degré de courbure de la façon suivante :

$$Courbure = \frac{1}{2\pi} \frac{\Delta_\varphi}{||\overrightarrow{v_i}|-|\overrightarrow{v_{i-1}}||}$$

La mesure de la figure 4.5 calcule le degré de courbure au point P_i par rapport aux points $P_i - granularity$ et $P_i + granularity$.

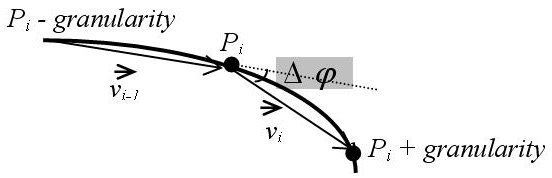


FIG. 4.5 – Mesure du degré de courbure à partir d'une trajectoire tridimentionnelle

Pour obtenir la courbure selon la méthode de l'abscisse curviligne, d'une trajectoire 3D en utilisant cette mesure, il faut choisir la granularité, c'est à dire la taille de la fenêtre en nombre de nucléotides, qui permet de construire les vecteurs $\overrightarrow{v_{i-1}}$ et $\overrightarrow{v_i}$, et de calculer l'angle entre ces derniers. La granularité, choisie à 40bp puis 70bp, correspond à la moitié de la taille d'une séquence pour laquelle la courbure est négligeable (environ 120bp), c'est à dire quasiment rectiligne (voir figure 4.6).

Mesure de distance entre les profils de courbures pour la clusterisation

Dans l'objectif de regrouper les séquences qui ont un profil de courbure similaire, afin de savoir si ce critère permet de caractériser les séquences "donneur" et les séquences "accepteur", nous avons utilisé la corrélation de *Pearson* comme mesure de distance entre deux profils de courbure.

Clusterisation

Afin de visualiser un éventuel regroupement des séquences en fonction de leur profil de courbure, nous avons utilisé une approche *K-means*. Un module de clusterisation générique par la méthode de *K-Means* a été développé en collaboration avec Foued Ben Hadjali durant son stage de maîtrise. Ce module est générique car il permet de clusteriser des ensembles d'objets de n'importe quel type, pour peu qu'une

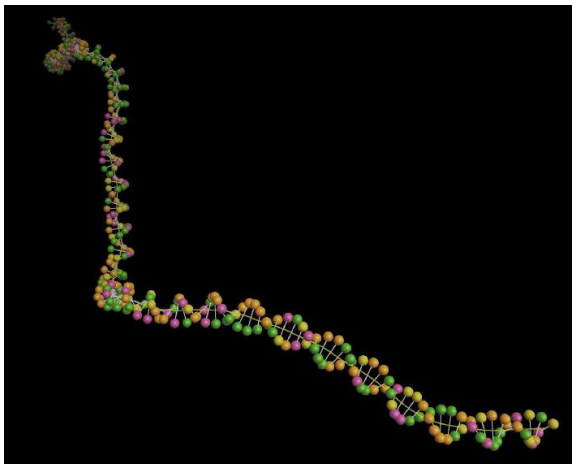


FIG. 4.6 – Modélisation et de visualisation de trajectoire tridimentionnelle d'ADN avec d'*ADN-viewer* : on observer en premier plan qu'une sous-séquence dont la longueur est inférieure à 100 bp (paire de base) a une trajectoire presque rectiligne

fonction de distance ait été implémentée, qui est calculée selon des critères spatiaux ou biologiques. Ce module a été développé dans le but de minimiser les appels à la fonction calculant cette distance. En effet, certaine fonction de distance entre deux objets biologiques, comme le score d'alignement de deux séquences, peuvent être très coûteuses si elles sont appelées de nombreuses fois, comme c'est le cas dans *K-Means*. En faisant varier le nombre de groupes désirés, et en conservant le nombre de groupes pour lequel la rapport entre la somme des distances inter-groupe et des distances intra-groupe est maximale, nous avons trouvé que le nombre de groupes qui optimise ce critère d'évaluation étaient de deux.

Visualisation

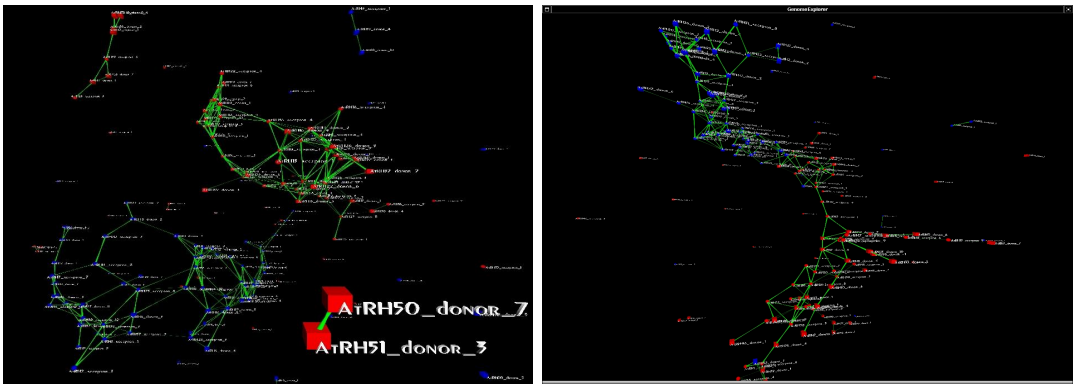
En parallèle avec ce résultat de clusterisation, nous avons visualisé la répartition spatiale de ces séquences selon le même critère de distance entre profils de courbure, et pour recouper ce résultat avec celui obtenu par la méthode *K-means* précédemment effectué, les groupes "accepteur" ou "donneur" étant identifiés par une couleur dans la représentation visuelle (*voir* figure 4.7).

4.1.2.4 Résultats et discussion

Dans la représentation, chaque séquence est représentée par un sommet, la mesure de corrélation détermine la longueur des arêtes. La couleur permet de différencier les séquences de type "accepteur" ou "donneur" des séquences initiales.

D'un point de vue méthodologique, on peut directement observer que le résultat

de la clusterisation en deux groupes, identifié par la couleur dans la figure 4.7, est cohérent avec la répartition spatiale des noeuds du graphe calculée indépendamment par *Genome3DExplorer*, à partir des mêmes données de distances entre le profil de courbure des séquences données en entrée. Cette méthodologie utilise les avantages des deux méthodes de clustering. Pour *K-means*, un nombre arbitraire de clusters doit être fixé *a priori*. On dispose généralement d'une mesure d'évaluation de la qualité du clustering (rapport entre la distance inter-cluster et la distance intra-cluster), mais le clustering doit être reproduit autant de fois que le nombre de groupes à tester. Par ailleurs, au sein d'un même groupe, contrairement à *Genome3DExplorer*, la position relative des éléments de ce groupe n'est pas connue. Enfin, les groupes formés par *K-means*, sont disjoints. En ce qui concerne *Genome3DExplorer*, le regroupement n'est pas une notion explicite, mais est une conséquence de la répartition spatiale des sommets du graphe calculée par l'algorithme de placement. Le filtrage interactif qui permet de réduire le nombres d'arêtes en éliminant les moins significatives, (graphiquement les plus longues, et biologiquement celles qui relient les noeuds qui ont un profil de courbure éloigné), a été particulièrement utile. En effet, initialement, même en éliminant les corrélations de courbure non significatives (inférieures à 0.5), le graphe était très connecté. La visualisation n'était donc pas exploitable en l'état. Ce filtrage a permis de continuer l'analyse en diminuant le seuil pour lequel un corrélation est jugée significative (0.05 et 0.15 dans les images 4.7, difficile à déterminer de manière automatique), et en simplifiant le graphe.



(a) Seuil de corrélation choisi à 0.05 (b) Seuil de corrélation choisi à 0.15

FIG. 4.7 – Répartition spatiale des séquences selon leur profil de courbure

D'un point de vue biologique, il apparaît que dans les deux groupes, résultats de l'opération de clusterisation, la répartition des séquences introns et exons est équilibrée. On peut donc en conclure que la courbure, calculée pour cette granularité spécifique de 60bp, n'est pas un critère significatif pour discriminer les introns des exons, ni les donneurs et les accepteurs. Cependant, quelques sous-graphes, consti-

tués uniquement par des donneurs, sont intéressants d'un point de vue biologique, et suggèrent que certaines classes de sites d'épissages, qui partagent un profil de courbure très proche. Cependant, ce résultat, décevant d'un point de vue biologique, peut être modéré par le phénomène suivant : la plupart du temps, les trajectoires 3D calculées à partir du modèle di-nucléotide, inférieure à 100bp, sont quasi rectilignes. Il apparaît que la granularité (40bp) choisie pour calculer la courbure se situe un peu trop prêt de la frontière pour laquelle on peut mesurer une courbure significative sur les séquences. Ainsi, pour être certain que ce critère ne peut caractériser les ou donneurs/accepteurs, il faudrait réitérer cette étude sur des séquences plus longues, de l'ordre de 250bp, en incluant l'environnement autour des exons d'une taille moyenne de 80bp chez *Arabidopsis thaliana*.

4.1.3 Co-régulations intra-chromosomiques chez la levure

Dans cette section, nous allons étudier une exploration de données concernant le troisième phénomène énoncé au début de ce chapitre, les co-régulations de gènes chez *Saccharomyces cerevisiae*. Le but de cette expérience est d'étudier l'organisation des gènes des chromosomes III et IX de la levure, considérant des critères de co-régulation associés et des critères géométriques. En effet, dans la cellule, plusieurs gènes s'expriment systématiquement même moment, régulés ensemble par un facteur biologique commun, phénomène que les biologistes appellent co-régulation. Même si deux gènes sont très éloignés l'un de l'autre (plusieurs dizaines ou centaines de milliers de nucléotides) dans la séquence linéaire (textuelle) du chromosome, ils peuvent être spatialement beaucoup plus proches sur la trajectoire tridimensionnelle du chromosome. Or, on sait que la proximité spatiale des gènes joue un rôle prépondérant et incontournable dans certains phénomènes biologiques comme la co-régulation [74].

4.1.3.1 Problématique

Cette section concerne l'étude et la mise en oeuvre d'une nouvelle approche concernant l'analyse de l'organisation globale de la transcription, en particulier chez *Saccharomyces cerevisiae*. Les gènes co-régulés de levure tendent à être régulièrement espacés le long des chromosomes [75]. L'intervalle diffère d'un bras de chromosome à l'autre. Aussi ces régularités suggèrent un modèle généralisé de boucle d'ADN, qui permettrait aux gènes co-régulés de se co-localiser dynamiquement dans l'espace. Ainsi, l'association des facteurs de transcription aux sites de liaison de l'ADN serait optimisée par l'augmentation de la concentration locale de ces deux partenaires. Habituellement, ces analyses se font avec une représentation plate (ordre dans la séquence linéaire), alors que l'on peut imaginer que la trajectoire tridimensionnelle de l'ADN basée sur les règles de la conformation 3d pourrait apporter un nouveau point de vue sur ces données de régulation. Il est donc intéressant d'étudier dans quelle mesure la structure de l'ADN nu contribue au rapprochement spatial des gènes transcriptionnellement co-régulés.

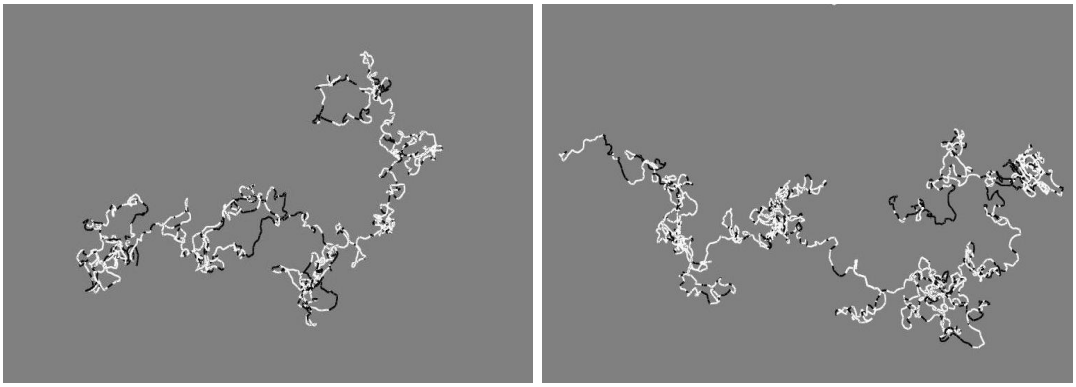
Plusieurs travaux portent sur l'analyse et la compréhension des réseaux de régulation transcriptionnelle basés sur des études à grande échelle chez *Saccharomyces cerevisiae* [77, 58, 104], mais l'organisation tridimensionnelle des gènes impliqués dans ces réseaux est largement inconnue. Toutefois, certains travaux [75, 74] tentent de comprendre comment les propriétés de structure tridimensionnelle sont impliquées dans les mécanismes fonctionnels. Un travail récent [75] montre que *Saccharomyces cerevisiae* présente une organisation spatiale périodique du génome qui concorde avec son réseau de régulation transcriptionnelle.

Dans cette étude, nous menons une exploration des relations entre l'organisation spatial du génome et la régulation transcriptionnelle chez *Saccharomyces cerevisiae*

guidée par la visualisation 3d. Notre but est de comprendre si l'architecture du génome présente une prédisposition en faveur de ses fonctions biologiques, et en particulier avec sa régulation transcriptionnelle. Notre approche est de considérer les gènes co-régulés linéairement très éloignés (dans la séquence textuelle) mais spatialement proches, pour observer savoir si le phénomène de co-régulation et peut être relié à l'architecture spatiale du génome.

4.1.3.2 Données *in silico*

La trajectoire tridimensionnelle de chaque chromosome de *Saccharomyces cerevisiae* a été calculée par *ADN-Viewer*. Les informations relatives au contenu génomique, comme le noms de gènes, leurs positions, dans la séquence, ont été obtenues par la plate-forme logicielle *GenoMEDIA* décrite dans le chapitre 1. On peut visualiser la trajectoire de ces deux chromosomes dans la figure suivante 4.8. Une autre mesure importante obtenue à partir de cette modélisation de la trajectoire tridimentionnelle de l'ADN, est le degré d'accessibilité pour un gène, permet d'évaluer selon les critères spatiaux, dans quelle mesure, un gène est accessible à des facteurs biologiques extérieurs (cf. annexe D).



(a) Le chromosome III (b) Le chromosome IX

FIG. 4.8 – Les trajectoires tridimensionnelles annotées des chromosomes III et IX de *S. cerevisiae*. Les phases de lecture ouvertes (ORF) apparaissent en blanc et les zones intergéniques en noir

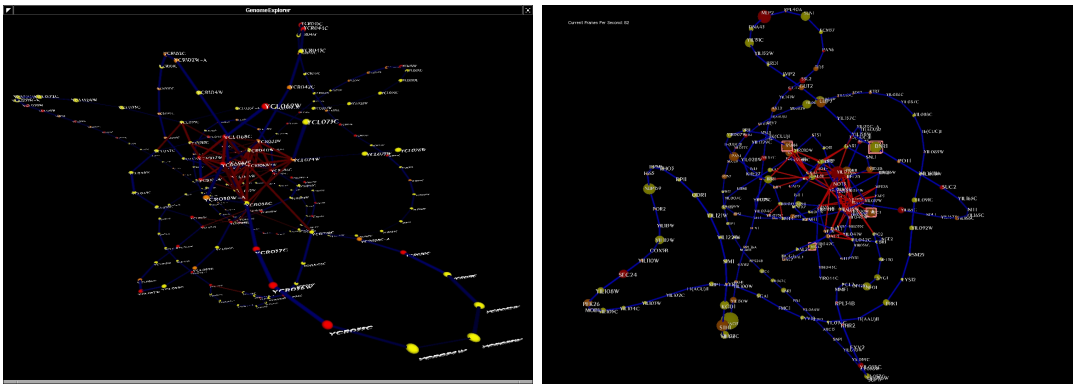
Données *in vitro* de co-régulation

Deux gènes sont co-régulés car ils sont en général contrôlés par un même facteur commun, dit de transcription. Les données utilisées que nous avons utilisé ici proviennent du travail de Lee et al. [77], qui ont identifié 106 régulateurs par Chromatin

ImmunoPrecipitation par (Immunoprécipitation de la chromatine, dite aussi ChIP), ainsi que les gènes qu'ils régulent. Dans cette étude, la métrique utilisée entre deux gènes, est le nombre de régulateurs commun à ces deux gènes.

4.1.3.3 Résultats

Pour déterminer si la proximité spatiale entre deux gènes joue un rôle dans le phénomène de co-régulation, nous devons d'abord étudié, à partir de la trajectoire de séquence d'ADN, la proximité des gènes dans l'espace pour les gènes co-régulés. Pour cela, nous représentons chaque gènes par un sommet dans le graphe. Nous avons reproduit l'ordre des gènes dans la séquence, dans le graphe décrivant les données, en reliant les gènes voisins dans la séquence par une arête, en bleu dans la figure 4.9. Les gènes co-régules selon les données de Lee [77], spatialement proches d'après modélisation de la trajectoire 3D de l'ADN (*cf.* 4.8), mais linéairement éloignés dans la séquences d'ADN, ont été reliés par une arêtes, en rouge dans la figure. Ce graphe a ensuite été visualisé par *Genome3DExplorer*. D'autre part, le degré d'accessibilité de la trajectoire d'un gène, (voir annexe D), est représenté par une couleur jaune (très accessible), orange (moyennement accessible) et rouge (peu accessible).



(a) Le chromosome III (b) Le chromosome IX

FIG. 4.9 – Visualisation de données de co-régulation intra-chromosomique, les gènes sont représentés par des sommets, jaune, orange ou rouge selon leur degré d'accessibilité. La séquence ordonnée des gènes sur l'ADN est représentée par les arêtes en bleu, et les relations de co-régulation entre les gènes spatialement éloignés par des arêtes rouges.

Pour les chromosomes III et IX, la figure 4.9, montre une structure symétrique en rosace, et exhibe une certaine périodicité dans l'arrangement spatial. Ce résultat est intéressant car il cohérent avec certains résultats sur le rôle de l'organisation tridimentionnelle dans le génome [75, 74]. Il est à souligner pour le chromosome III que les gènes co-régulés situés au centre du graphe présentent un degré d'accessibilité

en moyenne plus élevé que ceux se trouvant dans les boucles. Cette observation est intéressante puisque les gènes sont souvent co-régulés par un facteur extérieur, qui doit accéder physiquement à ce gène, pour le réguler. La relation entre le degré d'accessibilité et la co-régulation est donc une hypothèse biologique à étudier.

Par ailleurs, les gènes sont situés plus au centre et ont la plupart des relations qui correspondent aux zones télomériques ou sub-télomériques, situés linéairement à l'extrémité du chromosome : par exemple, sur le bras gauche du chromosome IX, les gènes HXT12 (YIL170C-YIL171W), YIL174W (pseudo-gène), YIL175W (pseudo-gène) et YIL177C (fonction inconnue), et, sur le bras droit, YIR040C et YIR041W. Cette répartition spatiale obtenue, uniquement guidée pour des données de co-régulation, est assez cohérente avec l'architecture du chromosome, et joue en faveur d'une hypothèse de phénomènes de co-régulation guidés par des critères spaciaux.

4.1.3.4 Discussion

Les données de structure issues de la modélisation avec *ADN-viewer* ont été enrichies par des données de régulation de Lee et al. [77]. Nous avons constaté que des régulations se font sur des groupes de gènes proches sur la séquence (*cluster*) et que d'autres régulations touchent des gènes très distants linéairement mais proches spatialement. De plus, pour les chromosomes III et IX, ce sont les régions télomériques ou sub-télomériques qui sont impliquées dans ces régulations-rapprochements.

Dans le contexte de cette étude, la visualisation 3d des données a été un outil puissant pour explorer le comportement biologique des structures tridimensionnelles. Le but de cette étude était de montrer qu'il y a probablement une relation entre les propriétés intrinsèques des structures tridimensionnelles calculées à partir de la séquence nucléotidique primaire et les propriétés des gènes incluant leur régulation. Si les gènes sont organisés linéairement [99], ils le sont également dans l'espace. En outre, certains travaux ont montré que les chromosomes préservent leur organisation au sein du noyau de génération en génération [54], ce qui joue fortement en faveur de l'hypothèse selon laquelle l'organisation spatiale génome joue un rôle très important dans le fonctionnement de la cellule, la nature ayant pour habitude de ne conserver que ce qui est utile.

4.2 Exploration d'un réseau d'alignements massif de séquences protéiques

Les séquences nucléiques ou séquences d'ADN, considérées en tant que séquence textuelle, ou en tant que trajectoire tridimentionnelle, ne sont pas les seules séquences étudiées pour comprendre l'organisation des génomes. Les séquences protéiques sont des polymères constitués d'une chaîne de quelques centaines à plusieurs milliers de 20 types de monomères de base, les 20 acides aminées, communs à la plupart des êtres vivants. Ces séquences sont aussi étudiées en tant que séquence textuelle (alphabet de 20 lettres correspondant au 20 acides aminées), et en tant qu'objet tridimensionnel. Nous expliquons dans cette section comment, nous avons exploré avec *Genome3DExplorer* des données massives d'alignements entre des séquences protéiques.

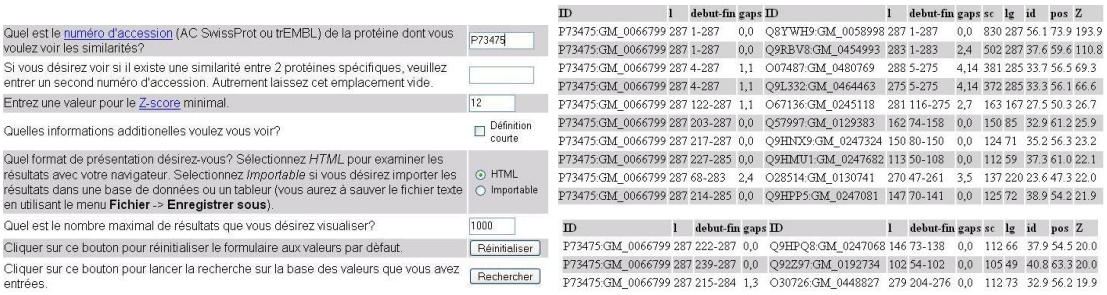
4.2.1 Données

La base de données que nous allons explorer est le *Décrypthon*, et résulte de la première comparaison exhaustive de toutes les protéines connues en 2000 identifiées chez les eucaryotes (animal, végétal, humain), en particulier celles de 76 génomes dont la séquence est entièrement connue. Cet outil doit notamment permettre aux chercheurs de reconnaître plus facilement la fonction biologique des séquences des gènes obtenues par les travaux de génomique, d'établir des familles de protéines qui ont conservé une importante similarité au cours de l'évolution, signe d'une fonction biologique commune, et d'identifier des régions de ces protéines déterminantes pour la réalisation de leur fonction biologique.

La base de données *Décrypthon* est mise à la disposition de la communauté scientifique pour lui permettre de progresser plus rapidement dans la compréhension des maladies génétiques et des maladies rares et dans le développement de nouvelles thérapeutiques. Initié par l'*AFM* (Association Française contre les Myopathies) dans le cadre du *Téléthon* 2001, *Décrypthon* a été rendu possible par la contribution de plus de 75 000 volontaires qui ont prêté bénévolement la puissance de calcul de leurs PC. Grâce à cette mobilisation sans précédent en France, les calculs (comparaison de 559275 séquences protéiques, au moyen de l'algorithme d'alignement de Smith-Waterman [98]) ont pu être effectués en moins de deux mois. Le projet a été mené en partenariat entre l'AFM, IBM et GENOMINING. Le résultat brut de ce travail est une liste de plusieurs centaines de millions d'alignements de protéines, alignement caractérisé par l'identifiant de deux protéines, une mesure de similarité de séquence entre ces protéines, et d'autres paramètres que nous n'avons pas considérés dans cette étude.

4.2.2 Problématique

Le point fort de cette banque de données est qu'il contient la globalité des alignements possibles entre toutes les séquences protéiques connues en 2000. Cet aspect global est très important, car il aurait permis des recoupements entre les connaissances provenant de différents organismes. Mais compte-tenu de la taille des données, la mise à disposition des résultats du *Decrypthon* et l'exploitation des données, sont restées très locales. En effet, l'utilisateur, à l'aide de l'outil d'interrogation , ne peut accéder qu'au voisinage direct d'une protéine donnée (*cf.* 4.10). La valeur ajoutée du résultat s'en retrouve diminuée, car il peut être obtenu par une simple requête effectuée en comparant cette protéine aux bases existantes en utilisant l'outil *blastP* [21], et un peu de patience.



(a) Formulaire de requête du *Decrypthon* (b) Présentation des résultats

FIG. 4.10 – Requête et résultat du *Decrypthon*

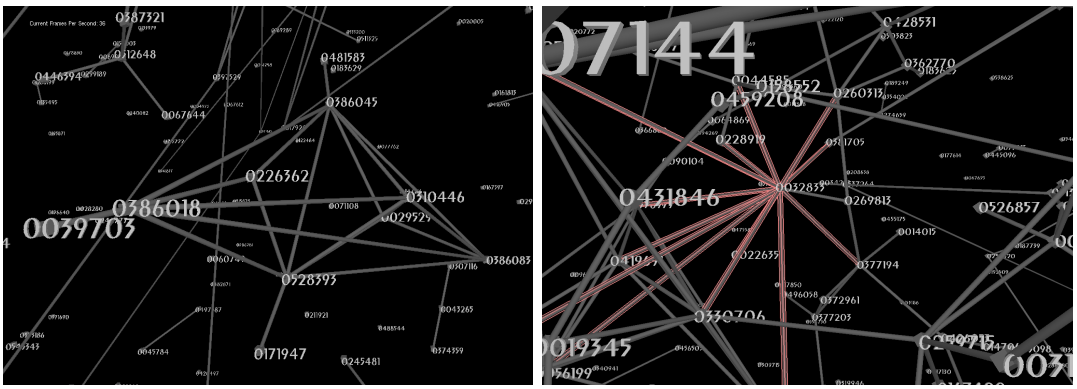
L'utilisateur peut certes rapatrier l'intégralité du résultat du *Decrypthon*, un volume conséquent de données brutes, environ 39 Gigas, et effectuer ensuite un traitement sur ces données, mais ce volume très important reste un frein à ce type d'approche. Nous avons donc essayer de trouver des solutions afin de permettre de tirer avantage de l'aspect global de ce calcul en utilisant *Genome3DExplorer* pour explorer au moins une partie du *Decrypthon*.

4.2.3 Méthode et résultats

Le *Decrypthon* est en fait un graphe énorme dont les noeuds sont les séquences protéiques et les arêtes le résultat d'alignement entre deux séquences, c'est pourquoi la traduction des données de cette banque, vers un format de *Genome3DExplorer* est simple. Cependant, la taille de ce graphe est très importante et a posé des problèmes en terme de limite de mémoire. La problématique principale associée à l'exploration du *Decrypthon* est celle du volume des données. Ce graphe contient environ 500000 noeuds et près de 600 millions d'arêtes. L'algorithme de placement présenté dans le chapitre précédent nécessite le chargement de la globalité du graphe pour effectuer

le positionnement relatifs des noeuds par rapport à leur zscore, et un simple calcul permet de conclure que les ordinateurs actuels ne peuvent charger autant de données dans leur mémoire vive. En étant minimaliste, c'est à dire en ne considérant que les informations essentielles comme le numéro des protéines et leur score d'alignement, et en comptant 32 bits par identifiant pour une protéine, et 32 bits pour le score d'alignement, il faut environ 10 GigaOctets de mémoire pour stocker ce graphe sous forme de liste d'adjacence.

Par ailleurs, le positionnement des noeuds pour dessiner le graphe, est un algorithme, d'une complexité quadratique, le temps nécessaire pour calculer ce positionnement est beaucoup trop important. Un filtrage a donc été réalisé pour traiter uniquement les données concernant *Homo Sapiens*, environ 10000 protéines. Nous avons ensuite utilisé le réglage interactif du seuil des scores d'alignement à visualiser, intégré dans le processus de positionnement, ce qui a permis aussi de simplifier le graphe en diminuant de façon draconienne le nombre d'arêtes. Ce réglage permet de ne conserver que les alignements les plus significatifs, et est donc plus synthétique. Il peut être réglé en fonction de la complexité des données en cours d'observation, et permet donc une exploration plus lisible de masse de données comme celle-ci, dont la figure 4.11 est un exemple.



(a) Identification d'un cluster de séquences protéiques

(b) Identification d'une séquence protéique constituant un *hub*

FIG. 4.11 – Exploration du réseau d'alignements de séquences protéiques : le *Decrypthon*

Une autre approche plus ciblée, en cours de réalisation, afin de pouvoir analyser plusieurs organismes, permet d'extraire le graphe voisinage d'une protéine, jusqu'à une profondeur donnée. C'est un bon compromis entre le besoin d'exploiter les données d'une manière globale, et l'importante difficulté de visualiser des graphes très lourds.

4.3 Exploration de données d'expression issues de puces à ADN

Nous avons décrit dans le premier chapitre, l'outil puissant que constitue les puces à ADN, et qui permet de mesurer l'expression de grands ensembles de gènes en fonction de conditions expérimentales variables. Nous allons étudier dans cette section, la dernière consacrée à l'étude de données génomiques factuelles, comme nous avons explorés ces données en utilisant *Genome3DExplorer*.

4.3.1 Objectif

Beaucoup de produit de gènes ne jouent pas qu'un seul rôle dans l'organisme, et peuvent donc faire partie intégrante de plusieurs chemins métaboliques. Cependant, la plupart des méthodes d'analyse de données de puces à ADN utilisées sont issus de la clusterisation hiérarchique ou de méthode *K-means*, qui ne peuvent refléter ces caractéristiques. En effet, dans ce type d'analyse, un élément ne peut appartenir qu'à un et un seul groupe distinct, même si de nouveaux travaux s'attachent à présenter de nouvelles solutions pour faire évoluer les usages dans ce domaine [88] [35]. De plus, ces méthodes d'analyse et leurs outils de visualisation associés, montre leur limite face à la taille toujours croissante des données de puces à ADN à analyser, qui peut aujourd'hui dépasser plusieurs milliers de gènes sur une seule puce. Enfin, ces outils d'analyse n'intègrent pas dans leur visualisation des données externes, avec lesquelles les utilisateurs vont pourtant croiser leurs résultats de clustering.

Dans cette expérience, nous répondu à cette problématique en utilisant *Genome3DExplorer* pour permettre une exploration de données de puces à ADN volumineuse dans un cadre immersif, et *GenoMEDIA*, afin d'augmenter les données de puces à ADN par des données externes.

4.3.2 Description des données de puces à ADN

Les données de puces à ADN sont constituées d'un ensemble de niveau d'expression de N gènes dans M différents contextes expérimentaux. Elles se présentent donc comme une matrice de nombres réels de taille $N \times M$, indexée par le nom des gènes pour chaque ligne de la matrice et les conditions expérimentales pour chaque colonne. Le profil est caractérisé par une ligne dans la matrice, et est une fonction discrète du niveau d'expression dans plusieurs contextes expérimentaux. Dans cette expérience, nous nous sommes intéressés aux patrons de co-expression, c'est à dire aux ensembles de gènes partageant des profils d'expression proches, chez *Saccharomyces cerevisae*.

4.3.3 Exemple d'inter-opérabilité avec d'autres sources d'informations

Pour pouvoir visualiser des données d'une puce à ADN, et des annotations externes concernant les gènes, il faut trouver un moyen d'intégrer les données brutes issues du traitements des images de puces à ADN, et des annotations externes contenues dans des bases/banques de données telles *GeneBank*. Pierre-Emmanuel GROS a développé dans notre équipe une plate-forme d'intégration de données et de services relatifs au génome, nommée *GenoMEDIA* [57]. Cette plate-forme permet d'intégrer à la fois des données génomiques hétérogènes et distribuées, mais aussi des services, qui permettent d'effectuer des calculs sur ces données. Ces données intégrées sont interrogeables de façon transparente pour l'utilisateur.

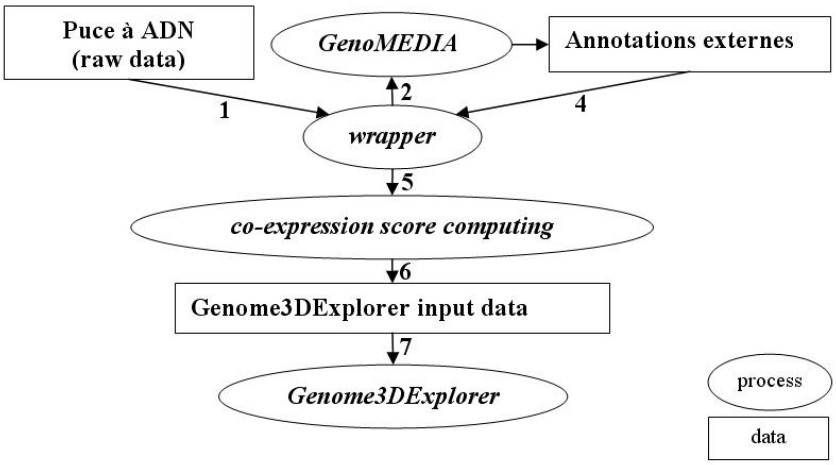


FIG. 4.12 – Processus de fusion de données de puce à ADN et de données externe issues de la plate-forme GenoMEDIA[]

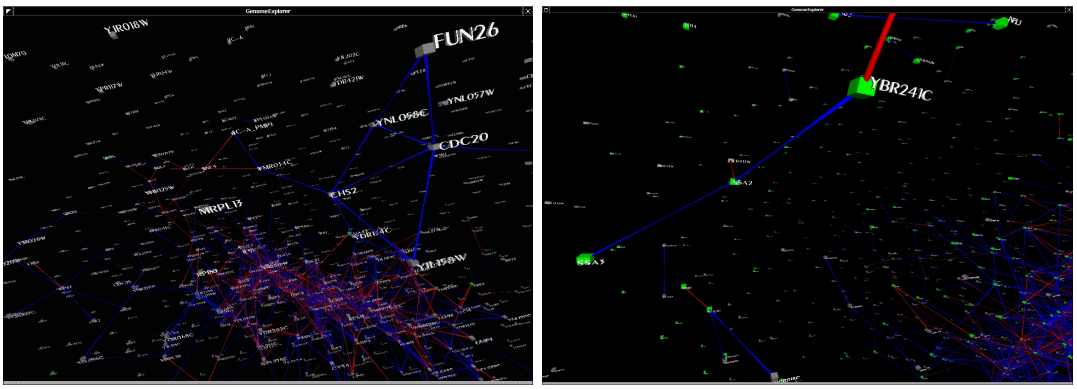
A partir de la liste des gènes [1] contenues dans les données brutes de la puce à ADN fournies en entrée, une requête SQL récupère les informations concernant ces gènes de la base à l'aide par l'intermédiaire la plate-forme de *GenoMEDIA* [2] [57]. Ces informations sont fusionnées à l'aide de ce même adaptateur [5], puis une corrélation entre chaque profil d'expression (ligne dans la matrice) de chaque gène sur la puce [6]. La formule de corrélation utilisée est la corrélation de *Pearson* :

$$cor(k,l) = \sum_{1 \leq j \leq M} \frac{(X_{kj} - \bar{k})(X_{lj} - \bar{l})}{\mu_k \mu_l}$$

Le résultat de cette corrélation est compris dans l'intervalle [-1..1]. Entre deux profils une corrélation proche de 1 indique que ces profils sont parallèles, et une corrélation proche de -1 indique que ces profils sont anti-parallèles. Une corrélation dans l'intervalle [-0.5 .. 0.5] est non significative. Ces données sont traduites en données relationnelles XML au format d'entrée de *Genome3DExplorer*. Dans cette expérience, l'objet d'intérêt est le profil d'expression d'un gène. Il est représenté par un sommet dans le graphe. Dans ce graphe, nous avons choisi de représenter le résultat d'une corrélation par la longueur de l'arête, de sorte que plus deux profils sont parallèle ou anti-parallèle, plus ils sont proches dans la représentation. Les profils parallèles sont relié par une arêtes bleu, et les profils anti-parallèles par une arête rouge. La couleur des sommets indique si la fonction du gène/profil est connue en vert, ou inconnue en gris, informations récupérées grâce à *GenoMEDIA*[57].

4.3.4 Résultats, validation et conclusion

Contrairement à la clusterisation hiérarchique et sa visualisation par dendogramme, utilisé dans la plupart des cas pour analyser ce type de données, l'appartenance d'un profil à un cluster n'est pas exclusive. De plus, par rapport à d'autres méthodes d'analyse de données de puces à ADN, cette solution permet d'obtenir visuellement un clustering selon différents critères de co-expression (parallélisme de deux profils d'expression, anti-parallélisme, décalage temporel), de façon simultanée. D'autre part, la visualisation en même de quelques attributs descriptifs de chaque gènes, permet d'effectuer des recoupements entre ces données externes, qui ne proviennent pas de l'analyse de la puce, et le regroupement de ces gènes selon leur profil d'expression dans la puce. Enfin, le filtage dynamique du seuil qui permet d'ajouter ou de retirer les arêtes les moins significatives, et de visualiser en temps interactif la répercussion de ces modifications sur l'organisation des données dans la représention. Cette fonctionnalité est très appréciée car elle permet de commencer une analyse systématique d'abord sur des regroupements de profils les plus significatifs, et ensuite, en relâchant le seuillage, d'approfondir la recherche.



(a) Sous-graphe de profils/gènes co-exprimés

(b) Sous-graphe contenant des profils de gènes cohérent avec des résultats biologiques (SSA1 et SSA2)

FIG. 4.13 – Exploration de réseau de co-expression chez la levure : un profil d'expression d'un gène est représenté par un sommet. Les sommets en vert correspondent aux profils/-gènes dont la fonction est connue. Les sommets sont reliés par des arêtes bleues si ces profils sont parallèles, et rouges si ces profils sont anti-parallèles

Quelques gènes identifiés dans un même cluster dans la représentation (figure 4.13), comme c'est la cas pour *SSA1* et *SSA2*, travaillent de manière synchronisée chez *Saccharomyces cerevisae*, font partie d'une même sous-famille de protéines chaperonnes (cytosolic HSP70 proteins) [111]. Une analyse systématique des gènes de fonction inconnue, appartenant à un groupe dans lequel certains gènes ont une fonction déjà identifiée, est en cours, pour valider complètement la méthode, avec pour objectif de faciliter la découverte de la fonction de gènes.

4.4 Exploration de connaissances bibliographiques

La solution développée pour explorer dans un contexte immersif les données génomiques, a aussi été conçue pour explorer les données de nature textuelles. L'expérience présentée dans cette section se situe en aval d'un outil d'extraction automatique d'informations à partir de texte nommé *ETIQ* [22].

4.4.1 Données

Le travail présenté ici a été réalisé en collaboration avec Oriane MATTE-TAILLEZ et Ahmed AMRANI, durant sa thèse. Le corpus étudié dans cette expérience est un corpus de publications concernant la maladie de Crohn [9]. Plusieurs résultats de sortie d'*ETIQ* [22], habituellement présenté sous forme textuelle, ont été formatés sous forme de graphes multivariés, dans un objectif d'exploration visuelle de ces résultats. *ETIQ* [22] permet, à partir d'un corpus de publications scientifiques spécialisées, d'extraire dans un premier les termes spécifiques au domaine du corpus [29], et ensuite d'extraire automatiquement les relations sémantiques entre ces termes écrit en langage naturelle, à partir de requête formulée sous forme de patrons de concepts.

4.4.2 Résultats

La figure 4.14 illustre l'organisation de concepts biologiques du corpus, avec une mesure de co-occurrence entre chacun de ces concepts, à différentes échelles (phrase, paragraphe, article). On peut ainsi directement observer dans la figure 4.14, les concepts *EPIDEMIO* (épidémiologie) et *DIS* (maladie) sont centraux dans ce corpus. En effet, il est, pour une grande part, constitué de données épidémiologiques sur cette maladie génétique.

La figure 4.14 est une vue d'ensemble du corpus, par la visualisation de ses principaux concepts, ce qui donne une vision synthétique de l'organisation et de la distribution relative des concepts au sein de ce corpus.

La figure suivante 4.15 est une vision plus détaillée, des termes issus de corpus, toujours organisée par une mesure de co-occurrence à l'échelle de la phrase. L'appartenance de chaque terme à chaque concept généraux est représenté par une couleur. La navigation dans le graphe est importante, car la vision, par rapport à la figure 4.14, est beaucoup plus détaillée, il n'est pas possible de percevoir l'organisation des termes de ce corpus d'un premier coup d'oeil, mais permet sans lecture, d'avoir une idée des relations entre les termes spécifiques du corpus au sein d'une même phrase.

Enfin, la figure 4.16 montre le résultat d'une analyse plus détaillée des relations entres les termes à l'échelle de la phrase. La relation entre deux termes, n'est pas une relation de co-occurrence comme dans l'analyse précédente, mais une relation

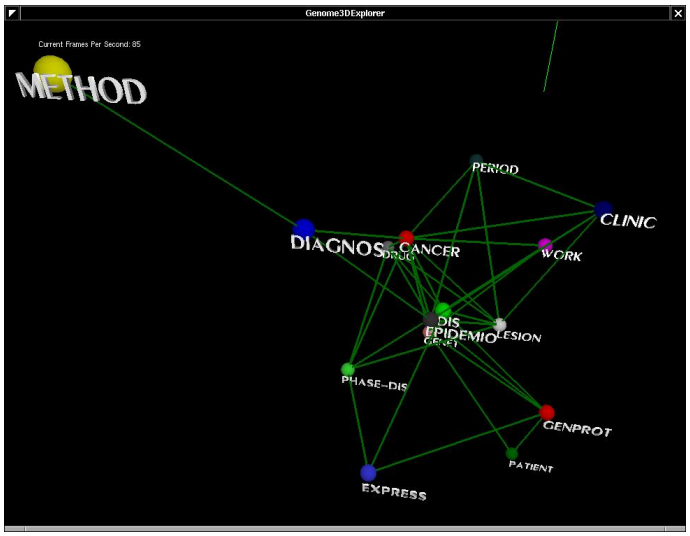
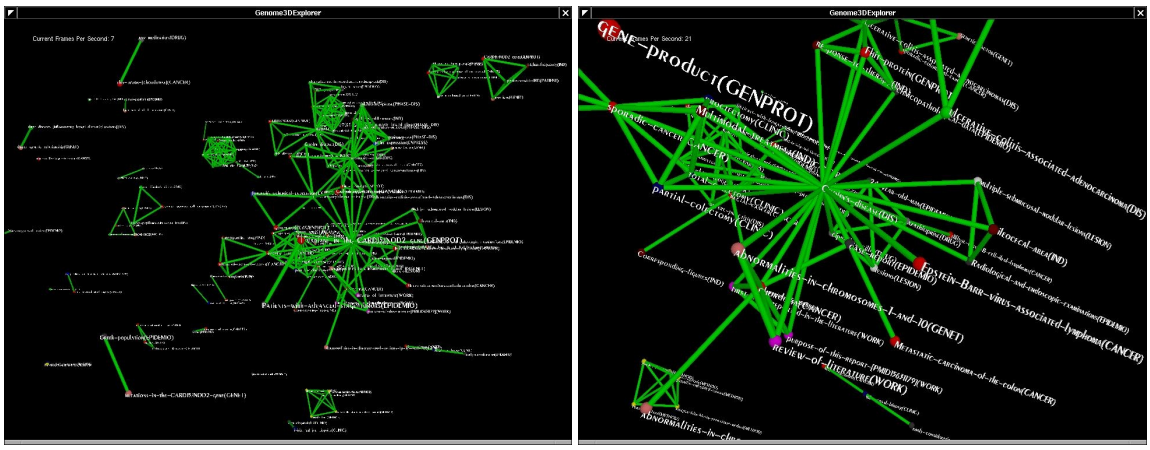


FIG. 4.14 – Organisation des concepts d'un corpus de publication concernant la maladie de Crohn, selon un critère de co-occurence (échelle : phrase)



(a) Vue globale

(b) Exemple de structure caractéristique d'un terme central dans le corpus

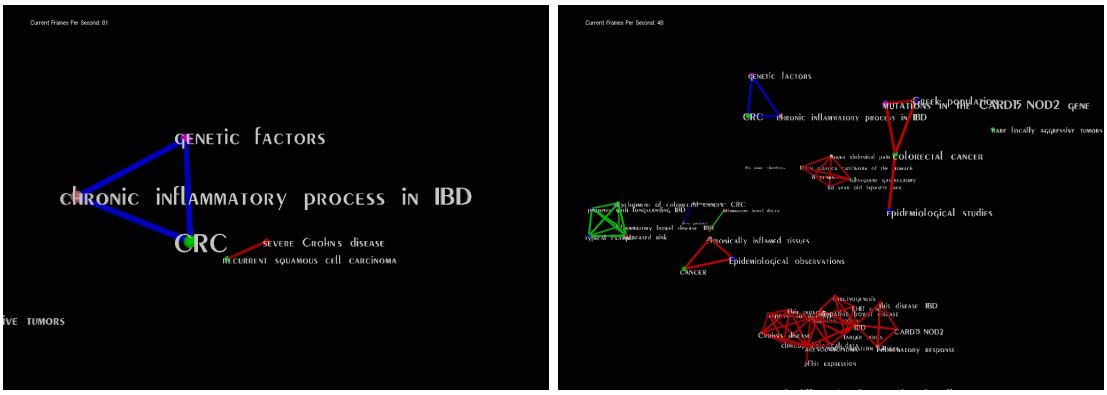
FIG. 4.15 – Organisation des termes d'un corpus de publications concernant la maladie de Crohn, selon un critère de co-occurrence (échelle : phrase) : chaque sommet est un terme, sa couleur indique sa classe conceptuelle, et les relations de co-occurrence sont représentées par des arêtes en vert

verbale sémantique qui relie deux termes. L'analyse s'est focalisée en reliant les concepts du texte par 3 grandes classes de relations verbales :

– la relation verbale d'induction (induit, provoque, déclenche...), en bleu dans la

figure,
- la relation verbale d'association, (est associé à, s'accompagne de, ...), en rouge dans la figure,
- et la relation verbale de synonymie (est un) en vert dans la figure.

L'appartenance d'un terme à une classe conceptuelle, est représentée par les mêmes modalités que dans l'image précédente, c'est à dire par la couleur, en choisissant autant de couleurs que de classes conceptuelles.



(a) Trois termes reliées par une même relation sémantique verbale

(b) Groupements de termes reliés par différentes relations sémantiques verbales

FIG. 4.16 – Organisation des termes d'un corpus de publications concernant la maladie de Crohn, selon des critères de relation sémantiques verbales : chaque sommet est un terme, sa couleur indique sa classe conceptuelle, et les relations sémantiques verbales entre ces termes sont représentées par des arêtes, colorées selon la classe de la relation sémantique verbale

Cette expérience montre la faisabilité d'explorer de façon efficace un corpus entier, à l'aide de l'intégration d'un outil d'extraction d'information à partir de textes, comme *ETIQ*, et d'un outil de visualisation adaptés comme *Genome3DExplorer*. Cette exploration peut être effectuée à plusieurs échelles, de la plus globale, en présentant l'organisation des grands concepts qui classifient les termes spécifiques à un corpus, à la plus détaillée, en visualisant à la fois les différents types de relations sémantiques verbales prédéterminées, et l'appartenance des termes à une classe conceptuelle.

4.5 Conclusion

Nous avons présenté dans ce dernier chapitre, une série d'expériences choisies, réalisées à l'aide de *Genome3DExplorer*, qui ont participé à sa conception et à son amélioration. Ces expériences ont été réalisées afin de confronter cet outil à la réalité

des données, le fossé entre la théorie et la pratique étant parfois assez important. Elles ont été réalisées par des biologistes ou des informaticiens, et prouvent que le concept de visualisation dynamique, qui permet une auto-organisation interactive des données génomiques variées, répond au besoin d'explorer visuellement des données pour découvrir ou faire naître de nouvelles hypothèses, et d'avoir une vision synthétique de grands ensemble de données. Par ailleurs, elles montrent aussi que *Genome3DExplorer* a répondu la nécessité d'offrir des modalités d'utilisation simple, compte tenu du public dont l'informatique n'est pas nécessairement l'activité principale.

Discussion sur l'usage de *Genome3DExplorer*

En ce qui concerne le contexte immersif, nous avons constaté que le principal avantage de cet environnement est de faciliter l'apprentissage, le travail collaboratif. En effet, la taille des dispositifs permet d'être à plusieurs utilisateurs en même temps, rend la désignation des objets, leur appropriation et la discussion plus facile. Dans un cadre pluridisciplinaire comme la bioinformatique, chacun des membres de la collaboration doit se former afin de comprendre la terminologie du domaine, et le fait d'avoir à sa portée un environnement immersif facilite la compréhension mutuelle, par l'émergence d'un vocabulaire partagé autour d'un objet virtuel commun. Par ailleurs, le contexte immersif, avec un espace de visualisation important, auquel s'ajoute l'aspect tridimensionnel, permet de visualiser plus de données, que dans la mesure où celles-ci ont été au préalable adaptées à ce contexte fortement interactif. Des travaux ont déjà montré les avantages du contexte immersif, pour répondre à la problématique associée aux données massives, ou de nature tridimensionnelle, comme des données de volumes en mécanique des fluides, des complexes moléculaires en biologie ou en chimie, ou de données génomique abstraites [55]. En ce qui concernent les données génomiques, *ADN-Viewer* [64][56] ou *SARAGene* [102] ont largement montré les avantages du contexte immersif. L'immersion donne accès aux compétences naturelles de l'être humain à se situer par rapport à son environnement, et permet de faire appel plus facilement à ses facultés de mémorisation visio-spatiale qu'il utilise dans la vie de tous les jours.

L'usage de *Genome3DExplorer* que des données variées a aussi permis d'étudier le comportement des utilisateurs pendant les phases d'exploration. Par exemple, en utilisant le système, nous avons constaté que pour la plupart des utilisateurs accordaient une importance variable aux attributs graphiques d'un graphe 3D. Cette importance est en partie liée à des différences de perceptions entre plusieurs critères visuels. Une hiérarchie est née de ces expérimentations. De manière générale, les propriétés visuelles invariantes à la position de l'utilisateur, comme la forme d'un sommet ou d'une arête, sa couleur,ou sa transparence, sont préférentiellement utilisées

pour rendre compte d' attributs importants. D'autres propriétés, comme la taille, sont considérés comme moins importantes, car lors des navigations, zoom ou prise de recul, ces propriétés ne sont pas invariantes, ce qui rend plus difficile l'interprétation des données qui ont été associées à ces modalités visuelles. Cependant, la longueur des arêtes, qui fait partie de la catégorie des propriétés variables selon la position de l'utilisateur, est paradoxalement considérée comme la caractéristique graphique la plus importante, afin de représenter un attribut quantitatif porté par une relation, car cet attribut guide l'organisation des données dans l'espace. Lors de la phase de choix des modalités visuelles, nous avons rapidement eu à prendre en compte de telles considération, de manière à faire correspondre de manière adéquate, les caractéristiques biologiques des objets à explorer, et les attributs graphiques qui leur seront associés, en fonction de d'importance variable que les utilisateurs accordent à certaines caractéristiques biologiques. Un autre besoin est né de l'étude de l'usage de *Genome3DExplorer*, et provient de la gêne occasionnée par la non reproductibilité du placement des sommets du graphe, compte tenu de la méthode de dessin utilisée. Or la reproductibilité d'une expérience sur paillasse par les biologistes, afin d'étudier certains phénomènes, une caractéristique essentielle à la crédibilité des résultats. Nous avons usé de nombreux arguments pour convaincre que ce placement, non reproductible à l'identique, n'était significatif que par le positionnement relatif des sommets du graphe, mais la demande forte d'obtenir à placement reproductible est restée tenace. Nous avons donc adapté *Genome3DExplorer* afin de ne plus choisir aléatoirement la position initiale des sommets. Nous avons extrait un arbre couvrant du graphe, puis nous avons utilisé une méthode de dessin radiale d'arbre analogue à celui présenté dans [43], et envoyé ensuite à l'algorithme dynamique de dessin du graphe pour prendre en compte les autres arêtes. Cette méthodologie permet d'obtenir une visualisation identique à chaque essai, en réponse aux exigences des utilisateurs.

Conclusion

Trop de morceaux de musique finissent trop longtemps après la fin

I. Stravinsky

L'objectif du travail réalisé durant cette thèse a été de proposer des solutions fournissant dans un contexte d'interaction immersive, une vision globale et synthétique d'une masse de données génomiques hétérogènes, qu'elles soient de nature textuelles ou factuelles. Cette visualisation devant être construite à des fins exploratoires, nous avons étudié plusieurs modalités de représentation visuelles de données génomiques utilisées dans certains outils d'analyse présentés dans le premier chapitre. Nous avons dans un second chapitre caractérisé les données génomiques, identifié et utilisé leurs points communs, pour faciliter le traitement de l'hétérogénéité des données génomiques. Ce chapitre présente aussi le contexte immersif, qui a été choisi car il est particulièrement bien adapté à l'exploration de données massives. Nous avons ensuite, dans le troisième chapitre décrit les solutions conçues afin de répondre aux problématiques associées à l'exploration, dans un contexte immersif fortement interactif, de données génomiques, massives, et hétérogènes, ainsi qu'une description de l'implémentation de ces méthodes au sein de notre application *Genome3DExplorer*. Enfin, le quatrième et dernier chapitre se consacre à la concrétisation de ces solutions à des applications réelles en génomique. L'exploration de données génomiques variées, en proposant des approches méthodologiques nouvelles qui utilisent les points fort de *Genome3DExplorer*, et qui ont souvent abouti à l'émergence d'hypothèses biologiques, ce qui constitue finalement la raison d'être de ce travail.

Contributions

Ce travail de thèse s'inscrit dans un contexte que nous pourrions appeler *Visual Mining*, parallèle et complémentaire à l'approche du *Data Mining*. Dans les deux approches, le but est d'inférer de la connaissance, implicite dans les données brutes. Dans le *Data Mining*, ces connaissances sont inférées par des techniques d'apprentissage, alors que le *Visual Mining* est une approche plus exploratoire, qui exploite la perception visuelle pour extraire la connaissance latente dans ces données, en exploitant les compétences naturelles et de l'expertise de l'utilisateur. La principale contribution réside dans la séparation de la gestion des données génomiques sous forme de graphe multivarié, et de leur visualisation sous forme de graphe tridimensionnel. Cette séparation des données et des modalités de représentation, est très importante, car elle permet la comparaison de plusieurs visualisations ayant une organisation spatiale guidée par des critères bioinformatiques différents. Elle offre ainsi à l'utilisateur la possibilité d'obtenir plusieurs points de vue sur des données identiques, en changeant simplement les modalités de représentation, et facilite aussi l'utilisabilité de *Genome3DExplorer*. La représentation informatique choisie pour intégrer une variété importante de données génomiques, sous forme de graphe multivarié, et la séparation entre ce graphe et sa représentation, fait que cette solution s'adapte facilement à d'autres données, dans la mesure ou celles-ci constituent des données relationnelles multivariées. Enfin, cette séparation, facilite la multimodalité en sortie, car elle permet les changements dynamiques de représentation en cours de visualisation. Cette multimodalité est importante dans le contexte immersif, dans la mesure ou ou n'utilise pas seulement le retour visuel pour transmettre de l'information à l'utilisateur, mais aussi des retours sonores ou haptiques, et qu'il est difficile de déterminer à l'avance quelle modalité sera la plus adaptée, pour transmettre une information.

Une autre contribution a été celle de l'adaptation de l'algorithme de Chalmers de 96 [36] pour représenter des données multidimensionnelles (graphe complet), afin de dessiner des graphes sans caractéristiques topologiques particulières, et dans un contexte interactif. Dans ce contexte, les modifications interactives du graphe à dessiner, comme une suppression, un ajout, ou un filtrage sur les arêtes, ont des répercussions sur l'organisation des sommets du graphe dans le dessin. La structure d'octree utilisée permet à la fois de réduire la complexité du phénomène de répulsion durant le placement initial, et d'effectuer la simulation des forces uniquement sur la partie visible du graphe, sujette à l'interaction.

L'exploration de ces graphes 3D à positionnement interactif, permet d'avoir une analyse à plusieurs échelles : la première, globale, grâce à la mise en évidence de la structure ou de la topologie sous-jacente d'un ensemble de données volumineux, rendue possible grâce aux outils interactifs de simplification du graphe ; la seconde,

locale, grâce à la navigation qui permet de zoomer sur des points d'intérêt de ce graphe, avec les mêmes possibilités de faire varier interactivement un filtrage en fonction de la lisibilité. Ces caractéristiques en font un outil particulièrement intéressant pour les biologistes et bioinformaticiens, lorsqu'ils s'agit d'explorer une masse importante de données génomiques, sans pour autant avoir d'idée précise sur ce qu'ils recherchent. Cette solution de visualisation permet aussi d'avoir une interprétation plus directe des résultats issus de processus de calcul sur des masses de données importantes, par exemple dans l'approche consacrée au *Decrypthon*. Enfin, cette approche est complémentaire aux multiples outils traditionnels d'exploration de données disponibles dans les banques de données génomiques factuelles sur Internet, car elle offre une vision synthétique de l'organisation des données (niveau *overview* selon Shneiderman [95]), alors que ces derniers sont souvent positionnés à un niveau de granularité assez fin (niveau *zoom* and *details* selon Shneiderman).

Perspectives

La vie, c'est comme une bicyclette, il faut avancer pour ne pas perdre l'équilibre

A. Einstein

Durant ces trois années de thèses, ma démarche fut guidée par la nécessité de m'assurer en permanence de l'utilité des solutions proposées aux utilisateurs. Cet objectif n'a pu être atteint que par une approche de conception centrée sur ces utilisateurs, afin d'augmenter l'utilité de *Genome3DExplorer*, de la représentation tridimentionnelle, et plus généralement de l'exploration immersive, contexte dans lequel les applications restent souvent considérées commes des applications "jouets". Ainsi les solutions proposées dans ce travail, sont le résultat de l'application et de la généralisation implicite de nombreux scénarios, à travers les expériences présentées dans le chapitre 4. D'autre part, l'observation des utilisateurs a permis de faire émerger des schémas de pensée généraux sur la façon avec laquelle les biologistes abordent une problématique concernant l'analyse de leurs données. On distingue dans cette méthodologie de développement, l'application involontaire de principes de conception centrés utilisateur, formalisés dans [93], et appliqués à la conception d'outils bioinformatiques dans [65]. En continuant dans cette optique, il devient nécessaire d'effectuer une évaluation plus formelle de cette application, guidée par des critères d'utilisabilité précis [92]. Pour réaliser ces évaluations, il faut mettre en place des expériences avec un nombre d'utilisateurs suffisant, afin de tester, la rapidité de familiarisation, l'utilisabilité, l'efficacité, et l'utilité de chacun des paradigmes de visualisation et d'interaction. C'est ce que j'appellerais, par analogie au génie logiciel, les tests unitaires des paradigmes de visualisation et d'interaction. En parallèle, il faut aussi mettre en place une évaluation plus globale de l'application. Toujours par analogie avec le génie logiciel, il s'agit d'évaluer la valeur ajoutée de l'intégration de ces paradigmes d'interaction et de visualisation dans un contexte immersif, d'un point de vue utilisateur, et par rapport à une tâche d'exploration de données biologiques bien scénarisée.

Une des autres ouvertures possibles serait de donner à l'utilisateur la possibilité d'utiliser un modèle physique plus général. Le comportement dynamique des objets constitué par la visualisation en temps interactif du modèle physique de positionnement des sommets du graphe, permet la réorganisation de la topologie lors d'une modification de ce graphe. Le modèle physique sous-jacent à ce comportement dynamique n'est cependant pas paramétrable. Ce modèle physique, s'ajouterait aux trois couches définies dans le troisième chapitre, pour permettre non seulement de déterminer la position des noeuds de façon dynamique, mais aussi de pouvoir faire évoluer d'autres caractéristiques d'un noeud (symbolisée par la taille, la couleur, ou la transparence), en fonction des caractéristiques des sommets adjacents. Cette

dynamicité permettrait de visualiser des informations dans le temps, comme la visualisation d'un réseau métabolique, étudié plus spécifiquement dans [67] et dans [33], mais de façon dynamique, en reliant par exemple la quantité de chacun des produits de la voie métabolique à un caractère graphique (par exemple, la taille), et scriptant une simulation définie par les formules de réactions métaboliques. D'autres structures qui présentent aussi un aspect profondément dynamique sont concernées, comme des réseaux de Markov, ou les réseaux de neurones, sur lesquels l'apprentissage fait évoluer les valeurs numériques des sommet et/ou d'arêtes. La visualisation de tel système serait un outil important pour l'aide au design et à l'optimation de ce type de structure. En effet, l'absence de formalisation pour concevoir de tels réseaux, afin de résoudre un problème spécifique, rend l'élaboration de tel réseau très empirique, et le nombre de couches cachées pour les réseaux de neurones (respectivement, nombre d'états cachées pour les HMM) ainsi que la structure du réseau, sont souvent optimisés par tâtonnements. Ainsi, la visualisation du processus d'apprentissage faciliterait sans doute cette tâche de conception de ce types de réseaux qui reste aujourd'hui profondément empirique.

Une autre piste à explorer dans un futur proche, est d'appliquer un filtrage qui pourrait être complémentaire au filtrage dynamique par seuillage sur la longueur des arêtes utilisé actuellement, pour simplifier le graphe. Il existe beaucoup d'approches de clustering de graphe, comme l'approche par clusterisation hiérarchique, mais les adapter à un contexte de navigation interactive n'est pas forcément chose facile. C'est pourquoi, nous réfléchissons à une clusterisation au préalable du graphe en utilisant les nombreuses méthodes de clusterisation, comme la clusterisation hiérarchique, par convolution proposée par Auber et al. dans [27] et utilisée dans *Tulip* [25] [24], ou selon des critères topologiques du graphe globaux [23] ou locaux [26]. L'objectif est de pouvoir adapter localement et automatiquement la granularité du dessin du graphe en fonction de leur distance à l'utilisateur. Dans la communauté de la visualisation d'information, cette idée a été formalisée par Furnas dans [51] pour généraliser les systèmes de visualisation de type *fisheye* en proposant des indices structuraux, le *DOI* pour *Degree Of Interest*, qui dépend d'une distance, et d'un indice arbitraire d'importance, le *API* pour *A Priori Of Importance*, associé à chaque objet *a priori*. Cette idée a été reprise et adaptée au contexte de visualisation interactive de graphe par van Wijk dans [108]. Dans la communauté du graphisme, cette idée est celle associée aux nombreuses techniques de gestion du niveau de détails (*Level Of Details*) dans une scène, qui dépend par exemple de la position de l'utilisateur et/ou de la direction de son regard, et dont une description exhaustive est effectuée dans [79]. Outre la difficulté d'avoir une représentation, dans laquelle plusieurs niveaux de hiérarchie différents localement, sont simultanément et dynamiquement dessinés en fonction des déplacements de l'utilisateur, ce type de filtrage pose le problème de la fusion des informations portées par les sommets et les arêtes.

Pour les valeurs quantitatives, j'imagine qu'un sommet représentant un regroupement de sommets à une échelle plus fine, sera représenté par un sommet dont la taille est proportionnelle au nombre de sommets qu'il contient, et un attribut quantitatif de ce sommet serait une moyenne du même attribut quantitatif des sommets constituant ce groupement. On peut appliquer cette méthode aux arêtes. Mais en ce qui concerne les attributs qualitatifs, je n'ai pas d'idées claires sur la façon fusionner les informations. La clusterisation hiérarchique pourrait par ailleurs être intégrée à l'algorithme de dessin, en diminuant peut être la complexité de cet algorithme par des méthodes analogues au dessin multi-échelle proposées par Jourdan [69] ou Koren [73].

Enfin, lors d'exploration de données massives, l'utilisation de la multimodalité en sortie [109], permet dans certains cas de répondre au problème de la surcharge du canal visuel, en transmettant les informations à l'utilisateur sur d'autres canaux, sonores ou haptiques par exemple. L'utilisation de la multimodalité est facilitée par la séparation entre les données et leur représentation, qui permet aisément de changer le canal de transmission d'une information. Même si nous avons présenté dans le troisième chapitre un paradigme associé à différents des canaux perceptifs, cette voie n'a pas été assez exploitée pour transmettre à l'utilisateur une quantité d'informations plus importante que dans un cadre non immersif.

Réflexions

C'est lorsque nous croyons savoir quelque chose qu'il faut justement réfléchir un peu plus profondément

F. Herbert

On donne à ces éléments reconstruits le nom de "racine", sans qu'il soit nécessaire de penser que ces racines sont autre chose qu'un terme abstrait de comparaison qui mesure les ressemblances

S. Auroux

Les échanges avec les biologistes n'ont pas été faciles. De mon point de vue, la compréhension sur la nature des données, nécessaire à l'identification de leur besoins, a nécessité une formation permanente. Du coté des utilisateurs, la difficulté provient d'une difficile séparation entre ce que les utilisateurs considèrent comme un résultat, et une représentation des données. Les visualisations ont parfois été considérées comme résultats, et non comme un outil d'analyse et d'exploration, comportement qui peut aboutir à des interprétations biologiques non fondées. La prise de conscience, que la représentation n'est qu'une manière d'organiser, de structurer, de hiérarchiser des données, est donc primordiale, afin de ne pas relier hâtivement la représentation avec une quelconque réalité biologique. Pour illustrer cette confusion, prenont un exemple de l'actualité scientifique. La thèse qui soutient que toutes les langues sont issues d'une langue unique ancestrale, partage la communauté des chercheurs en linguistique et en archéologie. Il est intéressant de constater que la thèse majoritaire chez les linguistes, est née de l'utilisation de la classification hiérarchique, initiée par Sir William Jones, au $XVIII^{eme}$ siècle, qui constata des similitudes entre le sanskrit, le grec et le latin, en supposant une origine commune à ces trois langues. Les linguistes ont ensuite, et jusqu'à aujourd'hui, fait de classification hiérarchique une méthode incontournable pour étudier l'histoire des langues. En utilisant exclusivement cette méthode arborescente, en réduisant les relations entre les langues à une relation de parenté, n'est-il pas naturel, en remontant dans l'arborescence, qui ne possède par définition qu'une seule racine, d'aboutir à une langue ancestrale commune (monogenèse) [28]. Je suis convaincu que cette thèse, pourtant majoritaire chez les linguistes [89], en dehors de considérations religieuses ou ethnocentrée, est essentiellement le fruit de l'utilisation de cette classification hiérarchique. D'ailleurs, elle est peu étayée par des découvertes archéologiques, et aucune population unique n'a pour l'instant pu être reliée à cette langue ancestrale ou à une proto-langue indo-européenne. La thèse antagoniste est celle de l'apparition de plusieurs langues (polygenèse). Certaines études utilisent en effet d'autres méthodes [59] [38], pour modéliser le fait qu'une langue peut être le résultat de la contribution de plusieurs langues, et plus d'une seule, comme c'est la cas pour le phénomènes de *créolisation*

[97], plus cohérents à la fois avec l'étude des langues qui ont émergé récemment et avec l'absence de faits archéologiques pour étayer l'existence d'un berceau d'une langue ancestrale unique. La même question se pose pour les biologistes : *LUCA*, pour *Last Unique Common Ancetor*, racine de l'arbre de classification des espèces, peut t-il être utilisé en temps que résultat pour appuyer la thèse de monogenèse de l'apparition la vie, ou doit t-il être considéré comme un concept méthodologique détaché de toute réalité. Ces exemples montrent l'importance de l'impact d'une méthodologie sur les interprétations de faits scientifiques identiques, interprétations parfois contradictoires selon la méthodologie utilisée. Ce sont les faits, non les méthodes pour les interpréter, qui doivent guider la construction de nos modèles. Dans cette perspective, la solution proposée n'est qu'un générateur d'hypothèses, résultats de l'analyse visuelle d'une proposition d'organisation des données génomiques, qui n'auraient pas émergées sans cette représentation.

Table des figures

1.1 Interface de visualisation de *FLAGdb++* . . . 11
1.2 En haut, alignement de 12 sites de liaison *cI* et *cro*, en bas, représentation de cet alignement avec *Sequence LOGO* . . . 12
1.3 Consensus de séquence protéines de (Globine) . . . 13
1.4 Visualisation du génome complet d'*Escherichia coli* avec *Genome Structure ATLAS* . . . 14
1.5 Visualisation de biomolécules . . . 17
1.6 Empilement des plateaux selon modèle dinucléotidique de Bolshoy et al. . . . 19
1.7 Représentations d'une trajectoire d'ADN de 10 bp à l'échelle nucléique avec *ADN-viewer*[64] . . . 20
1.8 Représentation d'une trajectoire d'ADN avec ADN-Viewver . . . 21
1.9 Les puces à ADN . . . 23
1.10 Analyse de données d'expression par clusterisation hiérarchique . . . 24
1.11 *BioBiblioMetrics* : Réponse à la requête "DNA repair and DNA Transcription" . . . 26
1.12 Visualisation du résultat d'une requête avec *Lysa* et la séquence selectionnée avec *ADN-viewer* en haut à gauche . . . 27
1.13 *Sequence World* : Exploration tridimentionnelle d'une banques de données génomiques . . . 28
1.14 *SARAGene* : un système de visualisation de données génomiques dans un contexte immersif . . . 29

2.1 Croissance en millions de paires de bases de 3 banques de données génomiques . . . 33
2.2 Classification des données génomiques . . . 35
2.3 Le dispositif immersif *MUSE* . . . 40
2.4 Lunettes pour la stéreovision . . . 41
2.5 Système de tracking ArtTrack©, utilisé dans *MUSE* . . . 43
2.6 Souris 3D . . . 44
2.7 Gants de données . . . 44

2.8 Virtuose™de chez Haption© 45

3.1 Graphe multivarié appliqué à des données biologiques, où les gènes 1 et 2 dont on connaît le nom, la taille en bp, et la co-localisation cellulaire, sont liés par un critère similarité de séquence, et de distance linéaire sur le chromosome 49
3.2 Éléments graphiques atomiques d'un graphe 3D 51
3.3 Espace d'arrivée de la fonction H de transformation d'un graphe multivarié en un graphe 3D 52
3.4 Exemple de codage des données génomiques en graphe multivarié en format XML 53
3.5 Exemple de définition de la fonction H en format XML 54
3.6 Illustration de la traduction du graphe multivarié vers un graphe 3D . 54
3.7 Un exemple de contraintes de distance insolubles dans le plan 58
3.8 Visualisation des itérations d'un algorithme de dessin de graphe de type *FDP* (Force Directed Placement) 60
3.9 Optimisation de l'algorithme de dessin par extraction d'un *noyau* de $\sqrt{N}$ sommets 66
3.10 Processus de navigation globale vers un sommet éloigné ou caché . . . 73
3.11 Affichage avec la légende à droite de la sémantique des attributs graphiques 74
3.12 Répartition spatiale de séquences nucléiques d'introns-exons selon un critère de courbure 75
3.13 Architecture de Genome3DExplorer 79
3.14 Description de la nature des données à analyser 81
3.15 interopérabilité entre des fournisseurs de données génomiques et *Genome3DExplorer* 82

4.1 Visualisation usuelle des blocs dupliqués dans 5 des 16 chromosomes de la levure 86
4.2 Visualisation synthétique des duplications entre les 32 séquences . . . 87
4.3 Processus d'obtention d'une protéine à partir de la séquence d'ADN . 89
4.4 Étapes de traitements des données brutes à la visualisation 91
4.5 Mesure du degré de courbure à partir d'une trajectoire tridimentionnelle 92
4.6 Modélisation et visualisation de trajectoire tridimensionnelle d'ADN avec d'*ADN-viewer* 93
4.7 Répartition spatiale des séquences selon leur profil de courbure 94
4.8 Les trajectoires tridimensionnelles annotées des chromosomes III et IX de *S. cerevisiae*. Les phases de lecture ouvertes (ORF) apparaissent en blanc et les zones intergéniques en noir 97
4.9 Visualisation de données de co-régulation intra-chromosomique 98

4.10 Requête et résultat du *Decrypthon* . . . 101
4.11 Exploration d'un réseau d'alignements de séquences protéiques : le *Decrypthon* . . . 102
4.12 Processus de fusion de données de puce à ADN et de données externe issues de la plate-forme GenoMEDIA[] . . . 104
4.13 Exploration de réseau de co-expression chez la levure . . . 106
4.14 Organisation des concepts d'un corpus de publication concernant la maladie de Crohn, organisé par une mesure de co-occurrence (échelle : phrase) . . . 108
4.15 Organisation des termes d'un corpus de publications concernant la maladie de Crohn, selon un critère de co-occurrence (échelle : phrase) 108
4.16 Organisation des termes d'un corpus de publications concernant la maladie de Crohn, selon des critères de relation sémantiques verbales 109

A.1 Du gène à la protéine . . . 135
A.2 Code de traduction des codons triplet de l'ARN en acides aminés constituant les protéines . . . 136

B.1 Visualisation du cytochrome p450 sur la *Protein Data Bank* à partir de son fichier PDB . . . 138

C.1 Visualisation des données de duplication I . . . 139
C.2 Visualisation des données de duplication II . . . 140

D.1 Degré d'accessibilité d'une trajectoire 3D d'ADN . . . 142

E.1 Principe de l'octree . . . 143

Liste des algorithmes

1 Algorithme de dessin de graphe en $\theta(N^3)$ 59
2 Algorithme de dessin de graphe en $\theta(N^2)$ 63
3 Algorithme avec gestion des répulsions par octree en $\theta(N^2)$ 65
4 Algorithme optimisé avec pré-positionnement d'un *noyau* en $\theta(N\sqrt{N})$ 67
5 Algorithme avec gestion par octree et pré-positionnement de *noyau* . 68
6 Algorithme ChercherDansSousArbre 69
7 Algorithme DessinerSousArbre 69
8 Algorithme de contruction d'un octree ContruireOctree 144

Bibliographie

[1] European Ribosomal RNA Database, http://www.psb.ugent.be/rRNA/.

[2] GeneBank, http://www.ncbi.nlm.nih.gov/Genbank/.

[3] GeneOntology, http://www.geneontology.org/.

[4] GLF OpenGL Font, http://www.forexseek.com/glf/index.html.

[5] Graphviz File Format, http://www.graphviz.org/.

[6] Human Genome Project, http://www.ornl.gov/sci/techresources/Human_Genome/project/about.shtml.

[7] ImMunoGeneTics, http://imgt.cines.fr/.

[8] KEGG : Kyoto Encyclopedia of Genes and Genomes, http://www.genome.jp/kegg//.

[9] Maladie de Crohn, http://fr.wikipedia.org/wiki/Crohn.

[10] Mouse Genome Informatics, http://www.informatics.jax.org/.

[11] NCBI Map Viewer http://www.ncbi.nlm.nih.gov/mapview/.

[12] Protein Data Bank (RSCB) http://www.rcsb.org/pdb/.

[13] PubMed http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?CMD=search&DB=pubmed.

[14] Saccharomyces Genome Database, http://www.yeastgenome.org.

[15] SwissProt, http://us.expasy.org/sprot/.

[16] The GML File Format, http://www.infosun.fmi.uni-passau.de/Graphlet/GML/.

[17] Thesaurus MeSH, http://ist.inserm.fr/basismesh/mesh.html.

[18] Virtual ENvironment for Immersive Simulation and Experiments (VENISE),http://www.limsi.fr/Recherche/ActionVenise/.

[19] YASARA, http://www.yasara.org//.

[20] Arabidopsis Genome Initiative. *Nature*, 408 :796–815, 2000.

[21] S. Altschul, W. Gish, W.Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, (215) :403–410, 1990.

[22] A. Amrani, Y. Kodratoff, and O. Matte-Tailliez. Un système dédié à l'étiquetage morpho-syntaxique des corpus de spécialité. *CIFT 2004*, pages 21–32, 2004.

[23] D. Archambault, T. Munzner, and D. Auber. Topolayout : Multi-level graph layout by topological features. *IEEE Transactions on Visualization and Computer Graphics : sous presse.*, 2006.

[24] D. Auber. Tulip. *Graph Drawing*, pages 335–337, 2001.

[25] D. Auber. Tulip : A huge graph visualisation framework. *Graph Drawing Softwares*, pages 105–126, 2003.

[26] D. Auber, Y. Chiricota, F. Jourdan, and G. Melançon. Multiscale navigation of small world networks. *Information Visualisation*, pages 75–81, 2003.

[27] D. Auber, M. Delest, and Y. Chiricota. Strahler based Graph Clustering using Convolution. *Information Visualisation*, pages 44–51, 2004.

[28] S. Auroux and L. Mayet. Entretien avec sylvain auroux : Le mystère des racines. *Sciences et Avenir*, 125 :12–15, 2001.

[29] J. Azé, M. Roche, Y. Kodratoff, and M. Sebag. Learning to order terms : supervised interestingness measures in terminology extraction. *International Journal on Computational Intelligence*, 1(2) :104–107, 2004.

[30] G. Battista, P. Eades, R. Tamassia, and I. Tollis. Algorithms for drawing graphs : An annotated bibliography. *Computational Geometry : Theory and Applications*, 4(5) :235–282, 1993.

[31] A. Bolshoy, P. McNamara, R. E. Harrington, and E. N. Trifonov. Curved dna without a-a : Experimental estimation of all 16 dna wedge angles. *Natl. Acad. Sci. USA*, 88 :2312–2316, 1991.

[32] I. Borg and P. Groenen. *Modern Multidimentionnal Scaling : Theory and Applications*. 1997.

[33] D. A. V. L. Bourqui, R. and F. Jourdan. Metabolic network visualization using constraint planar graph drawing algorithm. *Information Visualisation*, 2006.

[34] D. A. Bowman, C. J. Rhoton, and M. S. Pinho. Text input techniques for immersive virtual environments : An empirical comparaison. *Human Factors and Ergonomics Society - 46th Annual Meeting*, pages 2154–2158, 2002.

[35] R. Breitling, A. Amtmann, and P. Herzyk. Graph-based iterative group analysis enhances microarray interpretation. *BMC Bioinformatics*, 5(100), 2004.

[36] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. *Information Visualization*, pages 127–132, 1996.

[37] D. Cimbora and M. Groudine. The control of mammalian dna replication : A brief history of space and timing. *Cell*, 104 :643–646, 2001.

[38] C. Coupé. *De l'origine du langage à l'origine des langues : Modélisations de l'émergence et de l'évolution des systèmes linguistiques*. PhD thesis, Université Louis Lumière - Lyon 2, 2003.

[39] T. Cremer and C. Cremer. Chromosome territories, nuclear architecture and gene regulation in mamalian cells. *Nature Reviews Genetics*, 2 :292–301, 2001.

[40] S. Cruveiller, K. Jabbari, O. Clay, and G. Bernardi. Incorrectly predicted genes in rice ? *Gene*, 333 :187–188, 2004.

[41] C. L. N. dos Santos. 3d visualization resources into bioinformatics applications. Research note, limos rr-05-07, 2005.

[42] Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42 :149–160, 1984.

[43] P. Eades. Drawing free trees. *Bulletin of the Institute for Combinatorics and its Applications*, 5 :10–36, 1992.

[44] P. Eades and N. C. Wormald. Fixed edge-length graph drawing is np-hard. *Discrete Appl. Math.*, 28(2) :111–134, 1990.

[45] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Natl Acad Sci USA*, 95(25) :14863–14868, 1998.

[46] T. Eloranta and E. Mäkinen. Timga : A genetic algorithm for drawing undirected graphs. *Divulgaciones Matematicas*, 9(2) :155–171, 2001.

[47] A. J. Enright, S. Van Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30(7) :1575–1584, 2002.

[48] J.-D. Fekete and C. Plaisant. Excentric labeling : dynamic neighborhood labeling for data visualization. *SIGCHI conference on Human factors in computing systems : the CHI is the limit.*, pages 512–519, 1999.

[49] A. Fricke, A. Ludwig, and H. Mehldau. A fast adaptative layout algorithm for undirected graphs. *Graph Drawing*, pages 389–403, 1994.

[50] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software-Practice and Experience*, 21(11) :1129–1164, 1991.

[51] G. W. Furnas. Generalized fisheye views. *Human Factors in Computing Systems CHI*, 1986 :16–23.

[52] A. Gasch and M. Eisen. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology*, 3(11), 2002.

[53] S. M. Gasser. Positions of potential : Nuclear organization and gene expression. *Cell*, 104 :639–642, 2001.

[54] D. Gerlich, J. Beaudouin, B. Kalbfuss, N. Daigle, R. Eils, and J. Ellenberg. Global chromosome positions are transmitted through mitosis in mammalian cells. *Cell*, 112 :751–764, 2003.

[55] R. Gherbi, P. Bourdot, J.-M. Vézien, and J. Hérisson. *Volume 4 - Chapitre 4 Exploration de données scientifiques et expérimentations virtuelles*. Presses de l'Ecole des Mines, www.ensmp.fr/Presses, 2006.

[56] R. Gherbi and J. Hérisson. Representation and processing of complex dna spatial architecture and its annotated content. *Proceedings of the International Pacific Symposium on Biocomputing (PSB 2002)*, 2002.

[57] P.-E. Gros, N. Férey, J. Hérisson, and R. Gherbi. Genomedia, a midlleware platform for distributed genomic information. *IEEE International Conference on Information & Communication Technologies : from Theory to Applications*, 2004.

[58] N. Guelzim, S. Bottani, P. Bourgine, and F. Képès. Topological and causal structure of the yeast transcriptional regulatory network. *Nature Genetics*, 31(1) :60–63, 2002.

[59] A. Guénoche. Essai de classification des langues d'après la description de M. Ruhlen http://iml.univ-mrs.fr/~lafont/rencontres/OHLL.pdf.

[60] S. Hebsgaard, P. Korning, N. Tolstrup, J. Engelbrecht, P. Rouze, and S. Brunak. Splice site prediction in arabidopsis thaliana pre-mrna by combining local and global sequence information. *Nucleic Acids Research*, 24 :3439–3452, 1996.

[61] I. Herman, M. Marshall, and G.Melançon. Density functions for visual attributes and effective partitioning in graph visualization. *Information Visualization*, 2000.

[62] I. Herman and M. S. Marshall. GraphXML - an XML-based graph description format. *Graph Drawing*, pages 52–62, 2000.

[63] I. Herman, M. S. Marshall, and G. Melançon. Graph visualisation and navigation in information visualisation : A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1) :24–43, 2000.

[64] J. Hérisson, P.-E. Gros, N. Férey, O. Magneau, and R. Gherbi. DNA in Virtuo : Visualization and Exploration of 3D Genomic Structures. *3rd International Conference on Virtual Reality, Computer Graphics, Visualization and Interaction in Africa (Afrigraph '04)*, 2004.

[65] H. Javahery, A. Seffah, and S. Krishnan. Beyond power : Making bioinformatics tools user-centric. *Communications of the ACM – Special Issue on Bio-Informatics*, 2004.

[66] L. J. Jensen, C. Friis, and D. W. Ussery. Three views of microbial genomes. *Research in Microbiology*, 150 :773–777, 1999.

[67] F. Jourdan. *Visualisation d'information : dessin, indices structuraux et navigation. Applications aux réseaux biologiques et aux réseaux sociaux.* PhD thesis, Université Montpellier II, Montpellier, FR, 2004.

[68] F. Jourdan and G. Melançon. A tool for metabolic and regulatory pathways visual analysis. *SPIE Electronic Imaging, Data and Document Visualization*, 2003.

[69] F. Jourdan and G. Melançon. Multiscale hybrid mds. *Information Visualization*, pages 388–393, 2004.

[70] W. Kabsch, S. Sanders, and E. N. Trifonov. The ten helical twist angles of b-dna. *Nucleic Acids Research*, 10 :1097–1104, 1982.

[71] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31 :7–15, 1989.

[72] M. Kanehisa, J. Fickett, and W. Goad. A relational database system for the maintenance and verification of the los alamos sequence library. *Nucleic Acids Research*, 11 ;12(:149–58, 1984.

[73] Y. Koren and D. Harel. ACE a fast multiscale graph algorithm. *IEEE Symposium on Information Visualization*, 2002.

[74] F. Képès. On the transcription-based solenoidal model of chromosomes. *International Conference on Supercomputing in Nuclear Applications*, 2003.

[75] F. Képès. Periodic epi-organization of the yeast genome revealed by the distribution of promoter sites. *Journal of Molecular Biology*, 329(5) :859–865, 2003.

[76] N. Kyrpides and C. Ouzounis. Going wrong with confidence. *Mol. Microbiol.*, 32 :886–887, 1999.

[77] T. Lee, N. Rinaldi, F. Robert, D. Odom, Z. Bar-Joseph, G. Gerber, N. Hannett, C. Harbison, C. Thompson, I. Simon, J. Zeitlinger, E. Jennings, H. Murray, D. Gordon, B. Ren, J. Wyrick, J. Tagne, T. Volkert, E. Fraenkel, D. Gifford, and R. Young. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298 :799–804, 2002.

[78] O. Lespinet and B. Labedan. Orphan enzyme ? *Science*, 42 :307.5706.42a, 2005.

[79] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson, and R. Huebner. *Level of Detail for 3D Graphics.* The Morgan Kaufmann Series in Computer Graphics, 2002.

[80] A. Lukashin and M. Borodovsky. Genemark.hmm : new solutions for gene finding. *Nucleic Acids Research*, 24 :1107–1115, 1998.

[81] Morrison and Chalmers. A pivot based routine for improved parent finding in hybrid mds. *Information Visualization*, 3(2) :109–122, 2004.

[82] T. Munzner, F. Guimbretiere, S. Tasiran, L. Zhang, and Y. Zhou. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. *SIGGRAPH*, 22(3) :453–462, 2003.

[83] N. Pavy, S. Rombauts, P. Dehais, C. Mathe, D. Ramana, P. Leroy, and P. Rouze. Evaluation of gene prediction software using a genomic data set : application to arabidopsis thaliana sequences. *Bioinformatics*, 15 :887–899, 1999.

[84] L. J. Peck and J. C. Wang. Sequence dependence of the helical repeat of dna in solution. *Nature*, 292(5821) :375–378, 1981.

[85] A. G. Pedersen, L. J. Jensen, H.-H. Stærfeldt, S. Brunak, and D. W. Ussery. A DNA structural atlas for escherichia coli. *Journal of Molecular Biology (in press)*, 299(4) :907–930, 2000.

[86] D. Rhodes and A. Klug. Sequence-dependent helical periodicity of dna. *Nature*, 292 :378–380, 1981.

[87] I. Rojdestvenski, F. Pettersson, and D. Modjeska. Sequence world : A genetics database in virtual reality. *Information Visualisation*, page 513, 2000.

[88] J. Rougemont and P. Hingamp. DNA microarray data and contextual analysis of correlation graphs. *BMC Bioinformatics*, 4(1) :15, 2003.

[89] M. Ruhlen. *L'origine des langues : Sur les traces de la langue mère.*

[90] F. Samson, V. Brunaud, S. Duchene, Y. De Oliveira, M. Caboche, A. Lecharny, and S. Aubourg. Flagdb++, a database for the functional analysis of the arabidopsis genome. *Nucleic Acids Research*, 32, 2004.

[91] T. D. Schneider and R. M. Stephens. Sequence logos : A new way to display consensus sequences. *Nucleic Acids Res.*, 18 :6097–6100, 1990.

[92] A. Seffah and M. Donyaee. *Metrics and Measurement of Usability.* 2005.

[93] A. Seffah, J. Gulliksen, and M. Desmarais. *Human-Centered Software Engineering.* Springer, 2005.

[94] P. K. Shah, C. Perez-Iratxeta, P. Bork, and M. A. Andrade. Information extraction from full text scientific articles : Where are the keywords ? *BMC Bioinformatics*, 4(20), 2003.

[95] B. Shneiderman. The eyes have it : A task by data type taxonomy for information visualizations. *Visual Languages*, pages 336–343, 1996.

[96] E. S. Shpigelman, E. N. Trifonov, and A. Bolshoy. Curvature : software for the analysis of curved dna. *CABIOS*, 9(4) :435–440, 1993.

[97] J. Singler. Theories of creole genesis, sociohistorical considerations, and the evaluation evidence. the case of haitian creole and the relexification hypothesis. *Journal of Pidgin and Creole Languages*, 11 :185–231, 1996.

[98] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147, 1981.

[99] P. T. Spellman and G. M. Rubin. Evidence for large domains of similarly expressed genes in the drosophila genome. *Journal of Biology*, 1(1) :5, 2002.

[100] B. Stapley and G. Benoit. Biobibliometrics : information retrieval and visualization from co-occurrences of gene names in medline abstracts. *Pac Symp Biocomput*, pages 529–540, 2000.

[101] J. T. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. *Information Visualisation*, pages 57–, 2000.

[102] B. Stolk, F. Abdoelrahman, A. Koning, P. Wielinga, J.-M. Neefs, A. Stubbs, A. de Bondt, P. Leemans, and P. van der Spek. Mining the human genome using virtual reality. *Proceedings of the Fourth Eurographics Workshop on Parallel Graphics and Visualization*, pages 17–21, 2002.

[103] F. Strauss, C. Gaillard, and A. Prunell. Helical periodicity of dna, poly(da).poly(dt) and poly(da-dt).poly(dadt) in solution. *Eur. J. Biochem.*, 118 :215–222, 1981.

[104] S. A. Teichmann and M. M. Babu. Gene regulatory network growth by duplication. *Nature Genetics*, 36 :492–496, 2004.

[105] J. Thomas, D. Milward, C. Ouzounis, and S. Pulman. Automatic extraction of protein interactions from scientific abstracts. *Pacific Symposium on Biocomputing*, pages 538–549, 2000.

[106] D. Touraine and P. Bourdot. Veserver : A manager for input and haptic multi-sensorial devices. *Proceedings of the 10th IEEE International Workshop on Robot-Human Interactive Communication*, 2001.

[107] D. Touraine, P. Bourdot, Y. Bellik, and L. Bolot. A framework to manage multimodal fusion of events for advanced interactions within virtual environments. *Proceedings of International EUROGRAPHICS Workshop on Virtual Environments*, 2002.

[108] J. J. van Wijk and F. van Ham. Interactive visualization of small world graphs. *Information Visualisation*, 2004.

[109] F. Vernier. *La multimodalité en sortie et son application à la visualisation de grandes quantités d'information.* PhD thesis, Université Joseph Fourier - Grenoble I, 2001.

[110] C. Ware. *Information Visualisation : Perception for Design*. 2000.

[111] M. Werner-Washburne, D. E. Stone, and E. Craig. Complex interactions among members of an essential subfamily of hsp70 genes in Saccharomyces cerevisiae. *Mol Cell Biol*, 7 :2568–77, 1987.

[112] S. Wong and K. H. Wolfe. Duplication of genes and genomes in yeasts. *Comparative Genomics*, 15 :78–99, 2006.

[113] S. Zhai, M. Hunter, and B. Smith. The Metropolis Keyboard - an Exploration of Quantitative Techniques for Virtual Keyboard Design. *User Interface Software and Technology*, pages 119–128, 2000.

[114] P. Zimmermann, M. Hirsch-Hoffmann, L. Hennig, and W. Gruissem. Genevestigator. arabidopsis microarray database and analysis toolbox. *Plant Physiology*, 136 :2621–2632, 2004.

Annexe A

Le code génétique

La 1ère étape de l'expression des gènes est la *transcription* qui permet d'exprimer l'information contenue dans l'ADN en une copie ARN. Cet ARN messager transmet l'information génétique jusqu'aux ribosomes où s'effectue la *traduction*, seconde étape de l'expression des gènes.

La règle de complémentarité des bases nous permet d'imaginer facilement le mécanisme de transcription, au cours duquel l'information de l'ADN est transmise à l'ARN, une adénine en face d'un uracile ou d'une thymine, une cytosine en face d'une Guanine. Par contre pour la traduction c'est la séquence nucléotidique de l'ARN messager qui va déterminer la séquence en acides aminés de la protéine. Ce passage d'un "langage" à un autre est donc bien une traduction qui obéit à un code : le *code génétique*.

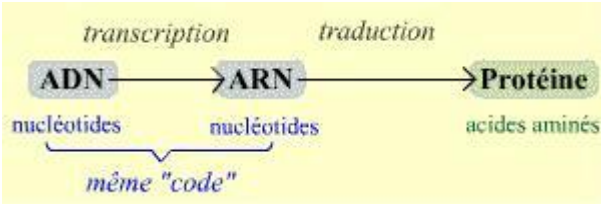


FIG. A.1 – Du gène à la protéine

Le code génétique définit la correspondance entre la séquence nucléotidique de l'ARN messager (et donc celle de l'ADN) et la séquence en acides aminés de la protéine. La séquence des acides nucléiques est une combinaison de 4 lettres (A, C, G, T dans l'ADN ; A, C, G, U dans l'ARN), la combinaison de ces 4 bases va permettre de "coder" pour les 20 acides aminés. Pour que cela soit possible, il faut qu'un acide aminé soit signifié (codé) par un groupe de 3 bases dans l'ARNm ; ce groupe de 3 bases, au niveau de l'ARNm, est appelé "codon". Comme il y a 4 bases, il y a 64 combinaisons différentes pour un triplet (4x4x4 = 64 codons).

La traduction de l'ARN messager se fait dans le sens 5' vers 3' et démarre à un point fixe (codon d'initiation : AUG), chaque codon rencontré est " décodé " en acide aminé. Chaque codon signifie un seul acide aminé. Il n'y a pas de chevauchement, cela veut dire qu'une base n'appartient qu'à un seul codon et par conséquent n'est lue qu'une seule fois. Sur les 64 codons existants, 3 sont non significatifs et correspondent, sauf exception, à des signaux de fin de traduction (ce sont les codons STOP : UAA, UAG, UGA). Il reste donc 61 codons pour 20 acides aminés, cela implique qu'un acide aminé peut être signifié par plus d'un codon. C'est en ce sens que nous parlons de code dégénéré, attention à ce terme car le code génétique n'est absolument pas ambigu, nous rappelons qu'à chaque codon correspond un seul acide aminé. Les codons signifiant un même acide aminé sont dits synonymes. Le tableau du code génétique présenté dans la figure A.2 permet d'avoir une correspondance entre la séquence du codon et la nature de l'acide aminé.

Le code génétique au niveau de l'ARNm.

1er nucléotide (en 5')	2e nucléotide U	2e nucléotide C	2e nucléotide A	2e nucléotide G	3e nucléotide (en 3')
U	Phe:F	Ser:S	Tyr:Y	Cys:C	U
	Phe:F	Ser:S	Tyr:Y	Cys:C	C
	Leu:L	Ser:S	STOP	STOP	A
	Leu:L	Ser:S	STOP	Trp:W	G
C	Leu:L	Pro:P	His:H	Arg:R	U
	Leu:L	Pro:P	His:H	Arg:R	C
	Leu:L	Pro:P	Gln:Q	Arg:R	A
	Leu:L	Pro:P	Gln:Q	Arg:R	G
A	Ile:I	Thr:T	Asn:N	Ser:S	U
	Ile:I	Thr:T	Asn:N	Ser:S	C
	Ile:I	Thr:T	Lys:K	Arg:R	A
	Met:M	Thr:T	Lys:K	Arg:R	G
G	Val:V	Ala:A	Asp:D	Gly:G	U
	Val:V	Ala:A	Asp:D	Gly:G	C
	Val:V	Ala:A	Glu:E	Gly:G	A
	Val:V	Ala:A	Glu:E	Gly:G	G

FIG. A.2 – Code de traduction des codons triplet de l'ARN en acides aminés constituant les protéines

Annexe B

Exemple de fichiers *PDB* (Cytochrome p450)

```
HEADER    OXIDOREDUCTASE                          03-JUN-04   1W0E
TITLE     CRYSTAL STRUCTURE OF HUMAN CYTOCHROME P450 3A4
...
SOURCE   2 ORGANISM_SCIENTIFIC: HOMO SAPIENS;
SOURCE   3 ORGANISM_COMMON: HUMAN;
SOURCE   4 ORGAN: LIVER;
...
REMARK   2 RESOLUTION. 2.8  ANGSTROMS.
REMARK   3   AUTHORS     : BRUNGER, ADAMS, CLORE, DELANO,
...
SEQRES   1 A  485  MET ALA TYR GLY THR HIS SER HIS GLY LEU PHE LYS LYS
SEQRES   2 A  485  LEU GLY ILE PRO GLY PRO THR PRO LEU PRO PHE LEU GLY
SEQRES   3 A  485  ASN ILE LEU SER TYR HIS LYS GLY PHE CYS MET PHE ASP
...
ATOM     13  N   GLY A  26      72.916 108.818  12.562  1.00 89.43           N
ATOM     14  CA  GLY A  26      73.386 107.623  13.253  1.00 84.98           C
ATOM     15  C   GLY A  26      72.333 107.186  14.268  1.00 82.66           C
ATOM     16  O   GLY A  26      72.176 106.005  14.572  1.00 80.58           O
...
ATOM   3608  N   ASP A 497      60.182  45.836  25.479  1.00 95.76           N
ATOM   3609  CA  ASP A 497      60.010  44.396  25.355  1.00 97.32           C
ATOM   3610  C   ASP A 497      60.132  44.014  23.873  1.00 98.09           C
ATOM   3611  O   ASP A 497      59.136  43.701  23.213  1.00 97.67           O
ATOM   3612  CB  ASP A 497      58.646  43.979  25.915  1.00 97.48           C
ATOM   3613  N   GLY A 498      61.359  44.052  23.352  1.00 98.57           N
ATOM   3614  CA  GLY A 498      61.579  43.716  21.954  1.00 99.46           C
```

```
ATOM   3615  C   GLY A 498      62.823  42.887  21.682  1.00 99.73           C
ATOM   3616  O   GLY A 498      63.694  43.353  20.908  1.00 99.36           O
ATOM   3617  N   THR A 499      62.925  41.766  22.236  1.00 99.99           N
TER    3618      THR A 499
...
CONECT 3650 3649
CONECT 3651 3619 3652 3655
CONECT 3652 3623 3651 3653
CONECT 3653 3652 3654 3656
MASTER      349    0    1   22    7    0    6    6 3689    1   45   38
END
```

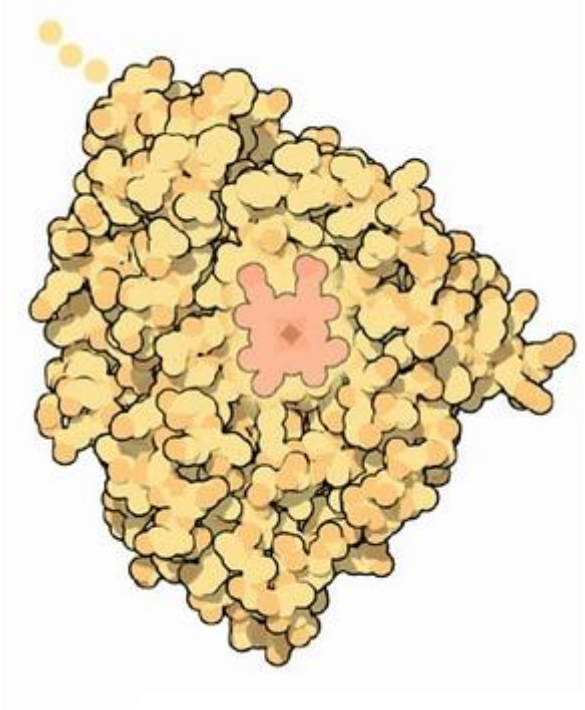


FIG. B.1 – Visualisation du cytochrome p450 sur la *Protein Data Bank* à partir de son fichier PDB

Annexe C

Duplications inter-chromosomiques chez la levure

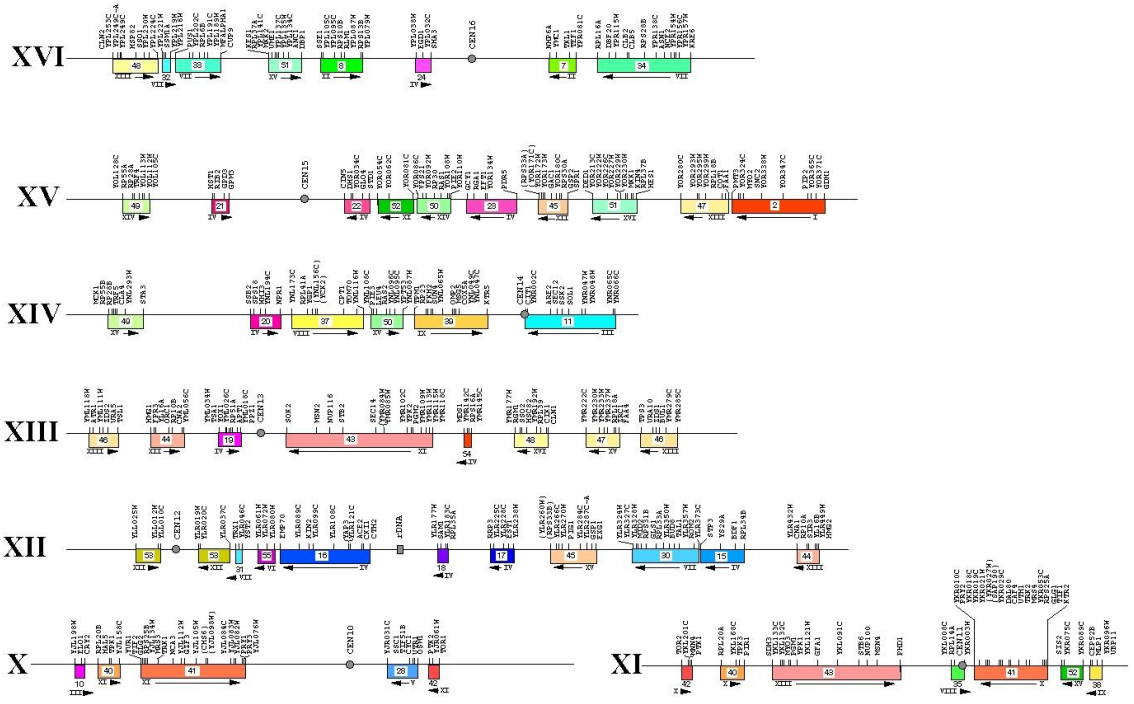


FIG. C.1 – Visualisation des données de duplication issues du travail de Wolfe [112] I

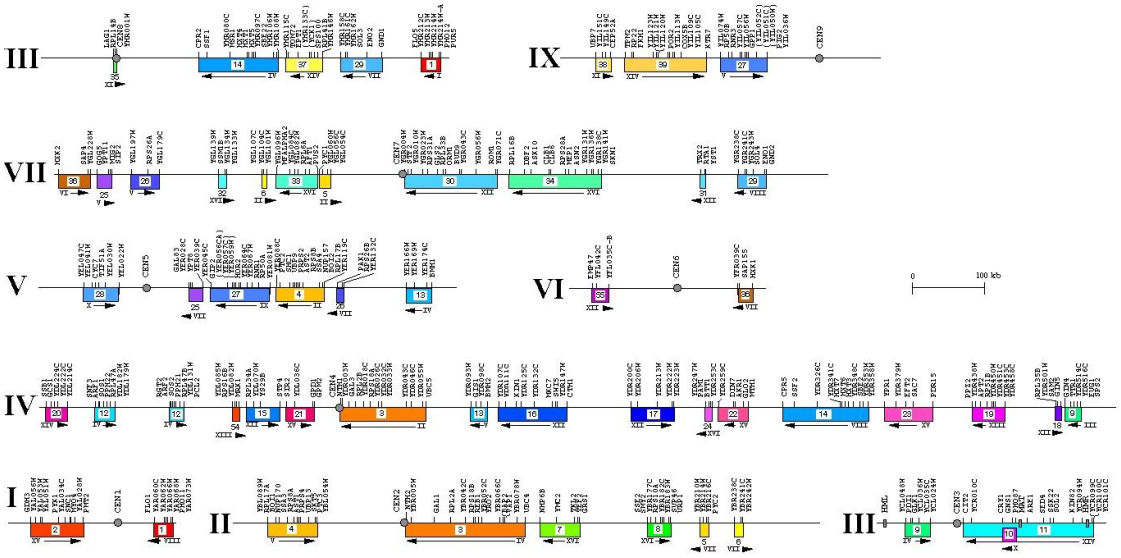


FIG. C.2 – Visualisation des données de duplication issues du travail de Wolfe [112] II

Annexe D

Degré d'accessibilité des gènes

Parce que le modèle de conformation 3d de l'ADN nu utilisé par *ADN-viewer* est statique, local et partiel, la trajectoire calculée d'un chromosome complet n'est qu'une approximation de la réalité. Il est donc important de relaxer nos critères géométriques et, pour la co-localisation des gènes, de considérer un degré d'accessibilité entre 2 gènes plutôt qu'une simple distance spatiale. En effet, nous pensons que 2 gènes en périphérie du chromosome et placés en vis-à-vis à une certaine distance pourront plus facilement se rapprocher dans l'espace que s'ils étaient plus proches l'un de l'autre mais dans une zone très dense du chromosome.

Le degré d'accessibilité détermine l'ouverture 3d maximale d'un gène, d'un nucléotide ou de tout autre segment d'ADN. Au sein d'un chromosome, le degré d'accessibilité d'un quelconque point d'intérêt est défini par l'angle maximal en ce point du sommet d'un cône qui ne doit intersecter aucun autre fragment de ce chromosome. Pour cela, nous avons construit une sphère centrée sur le point d'intérêt et sur laquelle on effectue la projection de la trajectoire 3d du chromosome (voir figure D.1). Puis l'algorithme effectue une phase de propagation de cette trajectoire projetée sur le reste de la surface de la sphère. On s'arrête quand la surface de la sphère est entièrement remplie. Pour la projection de la trajectoire 3d, l'algorithme prend la position 3d de chaque centre de plateau (P_i) et calcule les coordonnées polaires $(\rho_i, \theta_i, \psi_i)$ centrées sur le point d'intérêt i. Pour chaque plateau, l'algorithme remplit la surface de la sphère de projection S de telle manière que

$$\begin{aligned} S(\theta_i, \psi_i) &= 1 \text{ pour tout } i \\ &= 0 \text{ sinon} \end{aligned}$$

La propagation de la projection est ici gérée par une opération de dilatation (morphomath) qui consiste à transformer un 0 en 1 s'il est voisin d'un 1. Ainsi, on est sûr de remplir la surface de la sphère en un nombre fini d'itérations. Afin de propager la projection de manière homogène dans toutes les directions, il est nécessaire

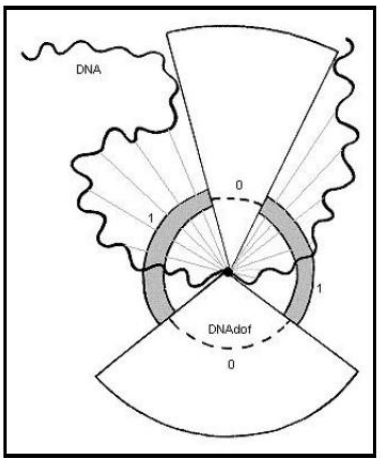


FIG. D.1 – Degree d'accessibilité d'une trajectoire 3D d'ADN

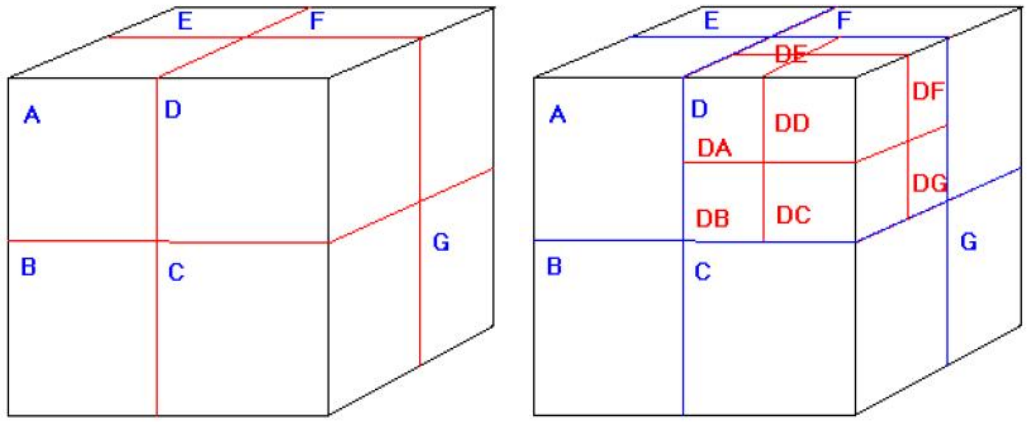
de choisir un pas identique d'échantillonnage pour les 2 angles θ_i et ψ_i. Le degré d'accessibilité d est défini comme le double du produit entre le nombre d'itérations N nécessaires pour remplir la surface de la sphère et le pas d'échantillonnage q de l'angle : $d = 2 \times N \times q$.

Annexe E

Les octrees, une méthode dichotomique de gestion de l'espace

Un octree O est un arbre contenant 8 octree fils qui subdivisent en 8 zones l'espace, selon la nomenclature illustrée dans la figure E.1.

- O_A, O_B, O_C, O_D, O_E, O_F, O_G, O_H les 8 octrees fils.
- O_{zone} ensemble qui contient la position des objets situés à l'intérieur de cet zone, $\emptyset$ si n'est pas une feuille
- un nom, selon la nomenclature de la figure E.1.



(a) Subdivision de l'espace en huit parties (b) Nomenclature des zones de l'octree

FIG. E.1 – Principe de l'octree

Algorithme 8 Algorithme de contruction d'un octree ContruireOctree

Entrées : P la position du centre de l'octree, L la longueur d'une arête de l'octree
Entrées : K la profondeur de l'octree, N le nom de l'octree
Sorties : un Octree

$O \leftarrow OctreeVide(K, N)$
$O_zone \leftarrow ListeDePositionVide$
Si $K > 0$ **Alors**
$O_A \leftarrow ContruireOctree(P + (-L/4, L/4, -L/4), K - 1, L/2, N + A)$
$O_B \leftarrow ContruireOctree(P + (-L/4, -L/4, -L/4), K - 1, L/2, N + B)$
$O_C \leftarrow ContruireOctree(P + (L/4, -L/4, -L/4), K - 1, L/2, N + C)$
$O_D \leftarrow ContruireOctree(P + (L/4, L/4, -L/4), K - 1, L/2, N + D)$
$O_E \leftarrow ContruireOctree(P + (-L/4, L/4, L/4), K - 1, L/2, N + E)$
$O_F \leftarrow ContruireOctree(P + (L/4, L/4, L/4), K - 1, L/2, N + F)$
$O_G \leftarrow ContruireOctree(P + (L/4, -L/4, L/4), K - 1, L/2, N + G)$
$O_H \leftarrow ContruireOctree(P + (-L/4, -L/4, L/4), K - 1, L/2, N + H)$
Fin Si
retourne O