وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR-ANNABA UNIVERSITY

UNIVERSITE BADJI MOKHTAR-ANNABA

جامعة باجي مختار- عنابة

Faculté des sciences de l'ingénieur Année 2009/2010

Département d'informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme de MAGISTER

Un modèle chimio-informatique pour une synthèse virtuelle

Option

Génie logiciel

Par

Debba Fatima Zohra

DIRECTEUR DE MEMOIRE : D[r] Mohamed Tahar Kimour, MC en informatique, Univ. Annaba

DEVANT LE JURY

PRESIDENT : D[r] Rachid Boudour, MC en informatique, Univ. Annaba

EXAMINATEURS : D[r] Tahar Bensebaa MC en informatique, Univ. Annaba

D[r] Yamina Tlili, MC en informatique, Univ. Annaba

ملخص

الإعلام الآلي ا لمطبق على الكيمياء و البيولوجيا يتطلب قدرات حسابية كبيرة ، وسائل معالجة ، تخزين و برمجيات.

مجال الكيميومعلوماتية هو مجال متعدد الأنظمة في أوج التطور. يهدف إلى إيجاد حلول معلوماتية لمشاكل مرتبطة بمعالجة المعلومة الكيميائية (تخزين ، بحث ، اكتساب و استخدام المعلومات).

في حال ما إذا كانت مداخل النظام متمثلة في هيئة بنية ،كما هو الحال في العديد من التطبيقات في مجال الكيمياء، أين المداخل هي عبارة عن مركبات كيميائية، يستحسن استعمال هذه البنية مباشرة لتشكيل أجوبة النظام، و التي يمكن أن تكون خصائص فيزيوكيميائية لهذه المركبات أو أنشطتها.

هذا العمل يقدم لهذا الهدف، طريقة تعتمد على التمثيل بطريقة التعلم الإحصائي و المسماة *آلة الخط البياني* ، والمرتكزة على تمثيل المداخل بالخط البياني الموجه اللاحلقي .

النموذج المشكل بطريقة *آلة الخط البياني* يتمثل في مجموعة دوال أساسية معلمة من نوع شبكة العصبونات و التي تتقاسم نفس المعلمات.

الدوال المرتبطة بمختلف الملاحظات مختلفة عن بعضها البعض لأنها تعكس البنية الملازمة لكل ملاحظة، هذا يعني أنه لكل ملاحظة مختلفة نموذج مختلف.

ولقد بينا أن التقنيات التقليدية لإختيار النموذج يمكن إستعمالها في إطار *آلة الخط البياني* و التي تسمح بتقييم قدرات التعميم للنموذج المقترح و لكن أيضا لاكتشاف أصناف المركبات التحت ممثلة في قاعدة التعلم ، و تقريب مجالات الثقة للتوقعات.

هذه الطريقة تتموقع هكذا بتقطع مع الطرق الكلاسيكية للتمثيل أين متغيرات المداخل ممثلة بأشعة و النموذج المشكل هو نفسه لكل الملاحظات.

Abstract

The information technology applied to chemistry and to biology requires enormous capacities of calculation, means of treatment, of storage and software. The field of chemoinformatics is an interdisciplinary field in booming. It aims at bringing the information technology solutions to problems linked to the chemical information processing (storage, research, acquisition and exploitation of knowledge).

When the inputs of system can be described as structured data (e.g. certain applications in the field of chemistry where inputs are molecules) it is more efficient to use directly this structure to model the output(s) of the system, which can be the physicochemical properties related to these molecules or their activity. We present, for this purpose, a statistical learning modeling method – called *graph machines* – where molecules, considered as structured data, are represented by graphs. For each individual of the data set, a mathematical function (*graph machine*) is built, whose structure reflects the structure of the molecule under consideration. It is the combination of identical parameterized functions (e.g. neural networks). Functions associated with the various observations to each other are different because they reflect the structure inherent in each observation, i.e. with distinct observations is associated distinct models.

We showed that the traditional techniques of model selection can be used within the framework of graph machines; they make it possible to evaluate the capacities in generalization of the models suggested, but also to detect the categories of molecules under-represented in the base of training, and to estimate the confidence intervals of the predictions.

This approach positions, so in rupture with the traditional methods of modeling where the variables in entry are represented by vectors and the built model is the same one for all the observations.

Résumé

L'informatique appliquée à la chimie et à la biologie requiert d'énormes capacités de calcul, de moyens de traitement, de stockage et de logiciels. Le domaine de la chimio-informatique est un domaine interdisciplinaire en plein essor. Il vise à apporter des solutions informatiques à des problèmes liés au traitement de l'information chimique (stockage, recherche, acquisition et exploitation de connaissances).

Dans le cas où les entrées d'un système se présentent sous la forme d'une structure, comme pour certaines applications dans le domaine de la chimie où les entrées sont des molécules, il est plus avantageux d'utiliser directement cette structure pour modéliser les réponses du système, qui peuvent être les propriétés physico-chimiques de ces molécules ou leur activité. Ce travail présente à cette fin, une méthodologie à base de la modélisation par apprentissage statistique appelé graph machines, et basée sur le codage des entrées en graphes acycliques orientés. Le modèle construit par la méthode des graph machines se présente sous forme d'une composition de fonctions paramétrées élémentaires (de type réseaux de neurones) qui partagent les mêmes paramètres. Les fonctions associées aux différentes observations sont différentes les unes des autres parce qu'elles reflètent la structure inhérente à chaque observation, i.e. à des observations distinctes sont associés des modelés distincts. Nous avons montré que les techniques traditionnelles de sélection de modèle peuvent être utilisées dans le cadre des graph machines ; elles permettent d'évaluer les capacités en généralisation des modèles proposés, mais aussi de détecter les catégories de molécules sous-représentées dans la base d'apprentissage, et d'estimer les intervalles de confiance des prédictions.

Cette approche se positionne, de ce fait, en rupture avec les méthodes classiques de modélisation où les variables en entrée sont représentées par des vecteurs et le modèle construit est le même pour toutes les observations.

DEDICACE

A mes très chers parents qui ont été toujours là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance. J'espère qu'ils trouveront dans ce travail toute ma connaissance et tout mon amour.

A mes chers frères et sœurs Samia, Rabah, Sara et le petit Cheroufa.

A mon marie khaled et sa famille.

A mes tantes et mes oncles.

A chaque cousin et cousine.

A tous mes meilleurs amis Abir, Souheila, Houda, Wafa, Jouda,…

A tous mes collègues du département d'informatique.

A tous les gens qui m'ont aidé et soutenue.

Je dédie ce travail.

Debba Fatima Zohra

Remerciement

Je tiens à exprimer mes remerciements les plus vifs à M. KIMOUR Mohamed Taher, Maître de conférences à l'université de Badji Mokhtar – Annaba, qui m'a fait l'honneur d'être le directeur de mon mémoire de magister.

J'ai appris de lui toute une philosophie de travail dans le domaine de la recherche scientifique. La liberté et la confiance qu'il m'a accordées ont beaucoup contribué au développement de mon autonomie dans le travail. Malgré que je me fusse longtemps loin de lui, il m'a apporté lors de nos rencontres des conseils judicieux, des encouragements, un soutien moral, beaucoup de sympathie et de bonne humeur.

Je tiens à remercier, également, les membres de Jury, Dr Rachid Boudour, Dr Taher bensebaa, et Dr Yamina Tlili. Je les remercie infiniment pour le temps qu'ils ont consacré à l'évaluation de mon travail.

Sans doute je ne vais pas trouver les mots pour remercier les personnes qui me sont les plus chères: mes parents, pour leur sacrifices, leur patience, et tous ce qu'ils ont fait pour m'apporter le bonheur, mes frères et sœurs (Samia, Rabah, Sara et en particulier le petit Mohamed chérif).

Merci individuellement à **mon mari Khaled** qui ma beaucoup aider tout au long de ce mémoire, ainsi que mon beau père Abdallah et ma belle mère Bia qui m'ont accompagné par la prière tout au long de ce mémoire, sans oublier ma belle sœur Houda et mon beau frère Amar.

En fin, je tiens à remercier tous mes cousines Nawel et Wided ainsi que mes amies (Souheila, Abir, Houda, Wafa, Jouda) et tous ceux qui m'ont soutenu surtout pendant les moments délicats, en particulier les membres de ma grande famille (surtout ma tante Malika, Hada , Mounira, Zhayra, ma grand mère Fatma et mon grand père Hssén) qui m'ont accompagné par la pensée tout au long de ce mémoire.

Je ne peux nommer ici toutes les personnes qui de prés ou de loin m'ont aidé et m'ont encouragé mais je les en remerciant vivement.

Debba Fatima Zohra

Liste des tableaux

Liste des figures

Table des matières

Introduction générale

L'informatique joue un rôle croissant dans la recherche en Chimie. Des secteurs très variés de la recherche fondamentale ou appliquée nécessitent des spécialités du traitement informatique, de l'information chimique, de la modélisation moléculaire ou de la chimie théorique.

La chimie se prête à un traitement informatique car elle est complexe et nécessite des capacités d'acquisition, de traitement et d'archivage considérables. D'importantes bases de données se constituent à travers le monde pour permettre aux chercheurs de suivre quasiment en temps réel l'avancement de la chimie. Le but est de montrer l'implication de l'informatique dans différentes applications de la chimie. Ce domaine est appelé "*Chemoinformatique*".

Le terme "*chemoinformatique* " est apparu il y a quelques années et a rapidement gagné l'utilisation répandue, Greg Paris a proposé une définition beaucoup plus large **[1]**

Chemoinformatique est un terme générique qui entoure la conception, création, organisation, gestion, récupération, analyse, diffusion, visualisation, et utilisation d'information chimique.

Clairement, la transformation des données en information et d'information en connaissance est un effort requis dans n'importe quelle branche de la chimie non seulement dans la conception de médicament. Nous partageons donc l'opinion que des méthodes de chemoinformatique sont nécessaires dans tous les secteurs de la chimie et adhérer à une définition beaucoup plus large :

Chemoinformatique est l'application des méthodes d'informatique pour résoudre des problèmes chimiques.

Pourquoi nous devons employer des méthodes d'informatique dans la chimie ? Beaucoup de problèmes en chimie sont trop complexes pour être résolus par des méthodes basées sur les premiers principes par des calculs théoriques. C'est vrai, premièrement, pour la relation entre la structure d'un composé et son activité biologique. Cependant, il s'applique également à beaucoup d'aspects de la réactivité chimique comme l'influence d'un catalyseur ou d'une température.

Dans cette situation, nous devons analyser des données expérimentales connues et établir un modèle pour les relations qui nous intéresse, comme entre la structure moléculaire et l'activité biologique ou la réactivité chimique.

Il y a une autre raison pour laquelle nous avons besoin des méthodes d'informatique dans la chimie : la chimie produit une quantité énorme de données, c'est particulièrement vrai pour des méthodes présentées dans la dernière décennie comme la chimie combinatoire et le criblage de haut débit. Juste le fait que plus de 45 millions de composés sont déjà connus et plusieurs million de composés sont découverts tous les ans souligne le besoin de traitement d'information électronique pour gagner une vue d'ensemble de chimie connue. Ainsi, le stockage de l'information chimique dans les bases de données a une longue histoire. Nous avons une grande variété de bases de données dans la chimie, par exemple, bases de données sur la littérature, structures, réactions, spectres, et sur des données physiques, chimiques, ou biologiques.

Par exemple, toutes les structures de rayon X des composés organiques et organométalliques sont stockées dans la structure de fichier de données de Cambridge **[2]** qui contiennent actuellement plus de 300.000 structures. De même, des spectres sont stockés dans des bases de données de spectres avec le plus grand contenant, par exemple, 200.000 spectres (IRS) infrarouges. Grand en tant que ce nombre pourrait sembler, il est petit en effet par rapport au nombre de composés connus. En effet, nous connaissons les structures 3D moins de 1% de tous les composés, et nous avons moins de 1% des spectres IRS des composés connus en forme électronique. Ainsi, d'une part, nous avons beaucoup d'information mais, d'autre part, pour beaucoup de problèmes nous manquons de l'information nécessaire.
La question est donc, pouvons nous apprendre assez des données que nous avons et pouvons nous gagnons la connaissance de ces données et d'information pour faire des prévisions pour le cas où l'information nécessaire n'est pas directement disponible ? C'est exactement l'un des secteurs où les méthodes de chemoinformatique peuvent entrer. Une grande partie des données a eu besoin, comme l'activité biologique d'un composé ou le taux ou le rendement d'une réaction chimique dans des conditions spécifiques, ne peut pas encore être directement calculé à partir des premiers principes en employant des méthodes théoriques dans un processus de l'étude déductive.
Dans cette situation, nous devons voir si nous pouvons apprendre assez des données déjà disponibles pour obtenir la connaissance qui permet la prévision de nouvelles données.

Dans cet effort nous devons mettre des données dans le contexte, nous devons rapporter différents types de données l'un avec l'autre pour créer l'information. Comme exemple, la seule valeur de l'activité biologique d'un composé ne nous aide pas beaucoup ; seulement quand nous connaissons également la structure chimique du composé à ce moment nous avons l'information valable. Avec beaucoup de morceaux d'information, par exemple, avec un ensemble de paires de structures chimiques et de leurs activités biologiques associées, nous pouvons essayer de généraliser, développer un modèle des relations entre la structure chimique et l'activité biologique, et obtenir ainsi la connaissance (la connaissance des relations entre la structure d'un composé et son activité biologique).

Ce processus de la connaissance dérivé des données et des observations s'appelle étude inductive. Les méthodes informatiques sont maintenant devenues disponibles pour l'étude inductive, comme des méthodes d'identification de modèle, des réseaux de neurones artificiels, ou des méthodes d'extraction de données. L'étude inductive a une longue histoire dans la chimie.

Problématique

L'informatique appliquée à la chimie et à la biologie requiert d'énormes capacités de calcul, de moyens de traitement, de stockage et de logiciels. Le domaine de la chimio-informatique est un domaine interdisciplinaire en plein essor. Il vise à apporter des solutions informatiques à des problèmes liés au traitement de l'information chimique (stockage, recherche, acquisition et exploitation de connaissances).

Par exemple, c'est en mêlant biologie, chimie, technologie et informatique que l'on cherche à mettre au point les médicaments de demain. La chemoinformatique est un secteur prometteur qui ouvre des horizons très larges dans la conception de nouveaux traitements. Il trouve aujourd'hui de nombreuses applications, en particulier dans les stratégies de recherche de nouvelles molécules bio-actives, susceptibles de devenir les médicaments de demain.

Développer un médicament coûte très cher. De plus, l'opération dure longtemps et se révèle parfois inefficace. Pour réduire la facture, les industriels préfèrent avant de lancer la synthèse chimique très onéreuse, réaliser une synthèse virtuelle (c'est-à-dire de concevoir et évaluer sur ordinateur) des molécules qui deviendront les médicaments de demain.

Cette simulation in silico permet de réduire le temps et le coût de développement des médicaments en décelant très tôt les plus prometteurs tout en limitant plus précocement les molécules qui échoueraient en développement.

C'est pourquoi la chimio-informatique occupe désormais une place de choix. Par exemple, dans le processus de fabrication de médicaments, elle s'intègre dans les diverses étapes du processus de découverte des futurs médicaments : conception et synthèse des molécules, identification des cibles thérapeutiques, évaluation des effets secondaires, etc.

Objectifs de la thèse

L'objectif de ce travail est de montrer comment des problèmes complexes, posés par un monde réel comme la chimie, peuvent se traduire en problématiques de recherche en informatique, qui fourniront à leur tour des réponses pertinentes du point vue chimique.

Principalement, il s'agit de montrer comment les problématiques de la chemoinformatique peuvent conduire au développement de thèmes de recherche fondamentale en informatique dont les résultats fournissent des réponses pertinentes du point vu chimique. On soulignera le caractère interdisciplinaire de ce travail qui nécessite le dialogue entre informaticiens, chimistes et biologistes, tant au stade de la modélisation du domaine qu'à celui de la validation des résultats. L'accent sera mis sur les possibilités offertes par la théorie des graphes pour représenter, comparer et classifier les objets chimiques que sont les molécules et leurs réactions.

Contribution

Le travail effectué au cours de ce mémoire s'inscrit dans le cadre de *la prédiction de propriété physico-chimique*, pour cela nous proposons une méthodologie pour la construction d'un modèle prédictif à base de la méthode de modélisation par apprentissage statistique appelée *graph machines*. En partant du principe très général suivant : plutôt que la modélisation des molécules a partir des descripteurs qui conduit a une perte d'information et qui génère pour toutes les observations un seul modèle lors de l'apprentissage, il serait

envisageable de représenter les molécules directement a partir de leurs structures (par des graphes acycliques orientés) afin de s'affranchir des problèmes décrits précédemment.

Organisation du mémoire

Ce mémoire est structuré de la manière suivante :

- Chapitre 1 : l'objectif de ce chapitre est d'apporter une présentation du domaine de la chemoinformatique, ses spécificités ainsi que ses particularités. Pour cela, après avoir donné la définition de la chemoinformatique et les concepts de base pour comprendre le monde chimique, nous présentons brièvement quelques méthodes et outils ainsi que les logiciels utilisés dans ce domaine. Nous proposons ensuite un panorama des applications de la chemoinformatique existantes en mettant particulièrement l'accent sur les applications de prédiction de propriétés et d'activités des molécules .En montrant l'importance des méthodes informatiques pour la résolution des problèmes de la chimie.

- Chapitre 2 : Présente les méthodes traditionnelles de QSAR et de QSPR. Nous rappelons les principaux types de descripteurs, les problèmes liés à leur calcul et à leur sélection, ainsi que les principales techniques conventionnelles d'apprentissage et de sélection de modèle.

- Chapitre 3 : Introduit la notion de graphe, et décrit la genèse des *graph machines*, fonctions de même structure mathématique que les graphes auxquels elles sont associées.

Nous montrons alors comment ces fonctions permettent d'établir une relation entre des données structurées et des nombres.

- Le chapitre 4 : Présente l'intérêt de la méthode des *graph machines* par rapport aux méthodes traditionnelles, pour cela nous proposons une méthodologie de modélisation à l'aide des *graph machines* basée sur un codage qui tient compte directement de la structure des molécules que nous désignons par QSAR-GM (GM pour graph

machines). Dans ce codage, chaque molécule est représentée par un graphe acyclique orienté dont les nœuds sont associes aux atomes et les arêtes aux liaisons.

À la fin de ce document, une conclusion générale fait le bilan sur l'ensemble de ces travaux de recherche et indique des perspectives et des défis de la prédiction de propriétés chimiques dans le domaine de la chemoinformatique.

La chemoinformatique est une discipline scientifique qui a évolué dans les 10 dernières années à l'interface entre la chimie et l'informatique. Il a été constaté que, dans de nombreux domaines de la chimie, l'énorme quantité de données et d'informations produites par la recherche en chimie ne peut être traitée et analysée que par les méthodes assistées par ordinateur. Ainsi, les méthodes ont été développées pour la construction des bases de données sur les composés chimiques et leurs réactions, pour la prédiction des propriétés physiques, chimiques et biologiques des composés et des matériaux à l'échelle de l'atome et de la molécule, dans tous les secteurs de l'activité humaine, la conception des médicaments, pour élucider la structure, pour la prévision des réactions chimiques et de la conception de synthèse organique. La recherche et le développement sont essentiels en chemoinformatique d'une part pour accroître notre compréhension des phénomènes chimiques et d'autre part pour que l'industrie reste compétitive dans une économie mondiale.

C'est une branche de la chimie et/ou de la physico-chimie qui utilise les lois de la chimie théorique exploitées dans des codes informatiques spécifiques afin de calculer structures et propriétés d'objets chimiques (molécules, solides, clusters, surfaces ou autres), en appliquant autant que possible ces programmes à des problèmes chimiques réels.

Nous commençons par la définition de la chemoinformatique et les concepts de base pour comprendre le monde chimique, par la suite nous présentons brièvement quelques méthodes et outils ainsi que les logiciels utilisés dans ce domaine. En fin nous présentons un panorama des applications de la chemoinformatique existantes en mettant particulièrement l'accent sur les applications de prédiction de propriétés et d'activités des molécules.

1.1 Définition

La **chemoinformatique**, également dénommée chemoinformatique (anglicisme) ou chimio-informatique, est le domaine de la science qui consiste en l'application de l'informatique aux problèmes relatifs à la chimie. Elle a pour objectif de fournir des outils et des méthodes pour l'analyse et le traitement des données issues des différents domaines de la chimie.

Elle est notamment utilisée en **pharmacologie** pour la découverte de nouvelles molécules actives et la prédiction de propriétés à partir de structures moléculaires.
Certaines applications de la chemoinformatique reposent sur les équations de la **physique quantique**. Elles permettent ainsi de modéliser les conformations des **molécules** [3].

Cette très belle image qui représente des atomes liés par des liaisons chimiques n'est qu'une des innombrables possibilités de la chemoinformatique.

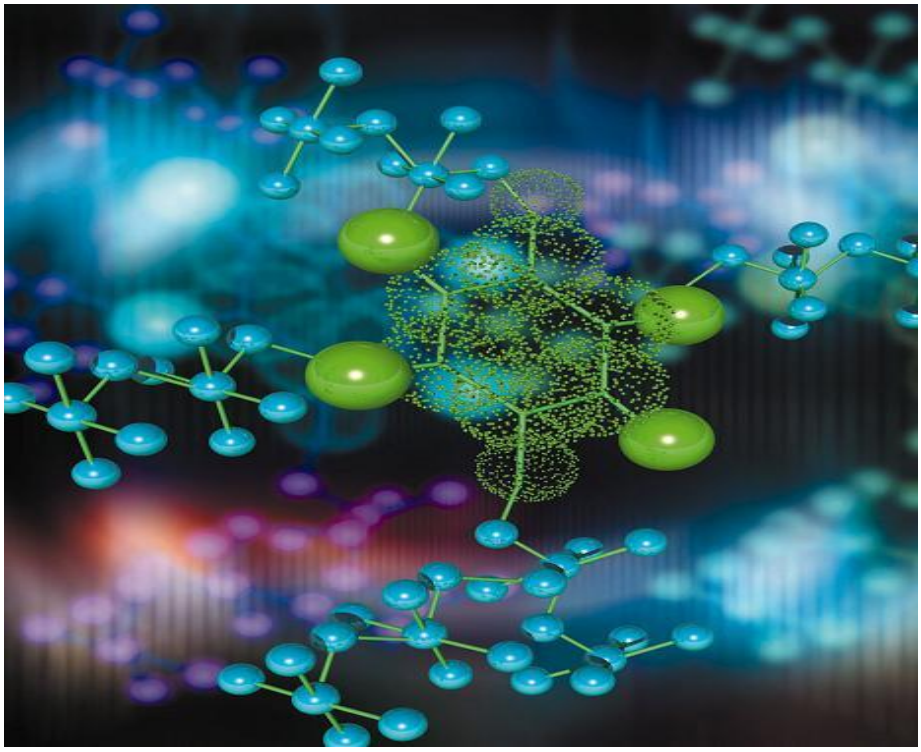


Fig.1.1 : Atomes liés par des liaisons chimiques.

La recherche et le développement en chemoinformatique sont essentiels pour :

- L'amélioration de la compréhension des phénomènes chimiques.
- Permettre à l'industrie chimique de rester compétitive dans le marché mondial [4].

1.2 Historique

La chemoinformatique est une nouvelle discipline apparue il y a environ 40 ans. Au début des années soixante, dans le but d'élucider la structure de composés chimiques inconnus, les données provenant des méthodes existantes (spectroscopie) ont été mises en commun sur informatique, C'était la naissance de la chemoinformatique. Le projet DENDRAL initié en 1964 à l'université de Stanford a été le premier à développer des générateurs de structures chimiques à partir de spectres de masses.

Ce n'est qu'à la fin des années 60 que Sasaki à l'université de technologie de Toyohashi et Munk à l'université d'Arizona ont utilisé plusieurs méthodes de spectroscopie afin d'élucider la structure chimique de leurs composés.

En 1969, Corey et Wipke ont présenté un travail similaire concernant les systèmes de représentation des molécules. Peu après d'autres groupes comme Ugi, Hendrickson et Gelernter ont développés des systèmes pour représenter des molécules organiques. La chemoinformatique (qui n'avait pas encore de nom) a été reconnue à la fin des années 60 comme une méthode utile d'analyse des données. Dès lors, beaucoup d'articles concernant cette discipline ont commencé à apparaître dans les journaux scientifiques. C'est seulement en 1998 que F.K Brown a défini pour la première fois cette discipline comme étant la chemoinformatique **[5]**.

1.3 Concepts de base

1.3.1 La chimie

La **chimie** est une **science** de la nature divisée en plusieurs spécialités, à l'instar de la **physique** et de la **biologie** avec lesquelles elle partage des espaces d'investigations communs ou proches.

Pour reprendre un canevas de présentation proposée par la plus grande association de chimistes au monde, l'*American Chemical Society*, la chimie étudie [6] :

1. les éléments chimiques à l'état libre, **atomes** ou **ions** atomiques, et les innombrables et diverses associations par liaisons chimiques qui engendrent notamment des composés moléculaires stables ou des intermédiaires plus ou moins instables. Ces entités de **matière** peuvent être caractérisées par une identité reliée à des caractéristiques quantiques et des propriétés précises.
2. les processus qui changent ou modifient l'identité de ces particules ou molécules de matière, dénommés **réaction, transformation, interaction...**
3. les mécanismes intervenant dans les processus chimiques ou les équilibres physique entre deux formes. Leurs définitions précises permettent de comprendre ou d'interpréter avec des hypothèses l'évolution matérielle avec en vue une exploitation des résultats de façon directe ou induite.
4. les phénomènes fondamentaux observables en rapport avec les forces de la nature qui jouent un rôle chimique, favorisant les réactions ou synthèse, addition, combinaison ou décomposition, séparation de phases ou extraction. L'analyse permet de découvrir les compositions, le marquage sélectif ouvre la voie à un schéma réactionnel cohérent dans des mélanges complexes.

1.3.1.1 Une science segmentée en de multiples disciplines

La recherche et l'enseignement en chimie sont organisés en disciplines qui, souvent en absence de services, de coopération ou d'aides réciproques, s'ignorent et se développent en toute autonomie [7] :

- la **biochimie** qui étudie les réactions chimiques dans des milieux biologiques (cellules…) et/ou avec des objets biologiques (protéines…).

- la **chimie analytique** est l'étude des méthodes d'analyses qualitatives et/ou quantitatives qui permettent de connaître la composition d'un échantillon donné ; ses principaux domaines sont : la chromatographie et la spectroscopie;
- la **chimie des matériaux** est la préparation et l'étude de substances avec une application en tant que matériau. Ce domaine intègre des éléments des autres domaines classiques de la chimie avec un intérêt particulier pour les problèmes fondamentaux concernant les matériaux.
- la **chimie inorganique** ou chimie minérale, concerne la description et l'étude des éléments chimiques et des composés sans squelette carboné.
- la **chimie organique** est la description et l'étude des composés comportant un squelette d'atomes de carbone (composés organiques) ;
- la **chimie physique** dont l'objet est l'étude des lois physiques des systèmes et procédés chimiques ; ses principaux domaines d'étude comprennent : la thermochimie, la cinétique chimique, l'électrochimie, la radiochimie, la sonochimie et les spectroscopies.
- la **chimie théorique** est l'étude de la chimie à travers un raisonnement théorique fondamental (habituellement à l'aide des mathématiques et de la physique). En particulier, l'application de la mécanique quantique à chimie a donné naissance à la chimie quantique. Depuis la fin de la seconde guerre mondiale, le progrès des ordinateurs a permis le développement de la chimie numérique (ou computationnelle).

1.3.2 L'atome : est formé d'un noyau atomique contenant des nucléons qui maintient autour de lui un nombre d'électrons équilibrant la charge positive du noyau.

1.3.3 L'élément chimique : est l'ensemble des atomes qui ont un nombre donné de protons dans leur noyau. Ce nombre est son numéro atomique. Par exemple, tous les atomes avec 6 protons dans leurs noyaux sont des atomes de l'élément carbone. Ces éléments sont

représentés dans le tableau périodique (Figure 1.2), qui rassemble les éléments de propriétés similaires.

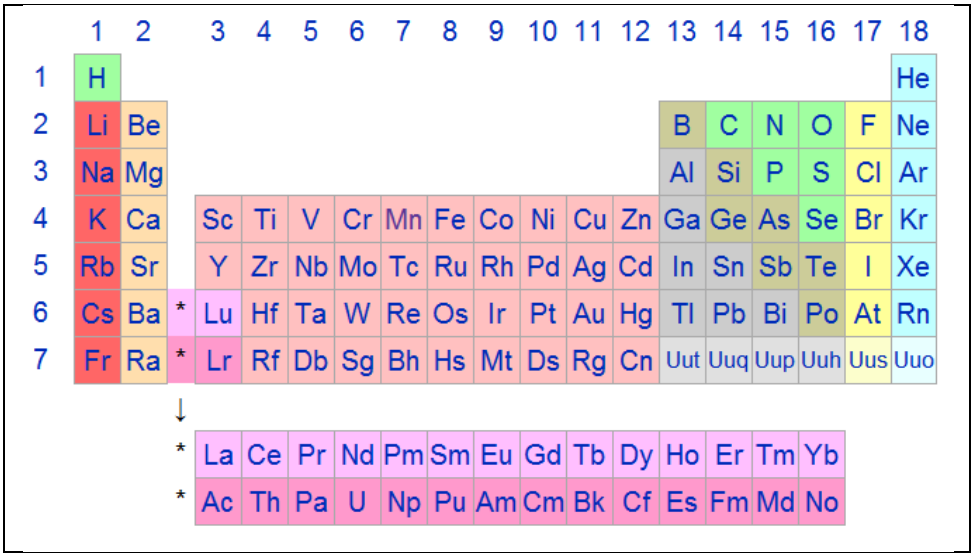


Fig.1.2 : Tableau périodique des éléments chimiques.

1.3.4 La liaison chimique:

est le phénomène qui lie les atomes entre eux en échangeant ou partageant un ou plusieurs **électrons** ou par des **forces électrostatiques** .

1.3.5 La molécule :

est un ensemble électriquement neutre d'atomes associés par des **liaisons covalentes.**

Exemple : L'**eau** : trois atomes, deux éléments, deux **liaisons**, une molécule. Un atome d'oxygène (ici en rouge), se lie à deux atomes d'hydrogène (ici en blanc).

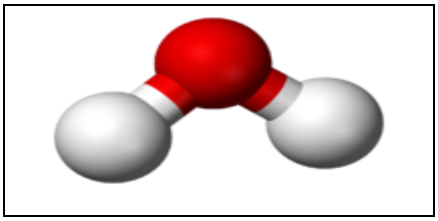


Fig.1.3 : Schéma de la molécule d'eau.

1.3.6 L'ion : est une **espèce chimique** (un atome ou une molécule) qui a perdu ou qui a gagné un ou plusieurs électrons. Il est appelé **cation** lorsqu'il est chargé positivement et **anion** lorsqu'il est chargé négativement.

1.3.7 Le composé chimique : est une substance issue de l'assemblage de plusieurs types d'atomes issus d'éléments chimiques différents dans des proportions définies. Il est caractérisé par sa **formule chimique**.

1.3.8 La réaction chimique : Une **réaction chimique** est une transformation de la matière au cours de laquelle les espèces chimiques (atomiques, ioniques ou moléculaires) qui constituent la matière sont modifiées : les espèces qui sont consommées sont appelées réactifs. Les espèces formées au cours de la réaction sont appelées produits (de réaction). Depuis les travaux de Lavoisier (1777), les scientifiques savent que la réaction chimique se fait sans variation mesurable de la masse : « Rien ne se perd, rien ne se crée, tout se transforme » qui traduit la conservation de la masse.

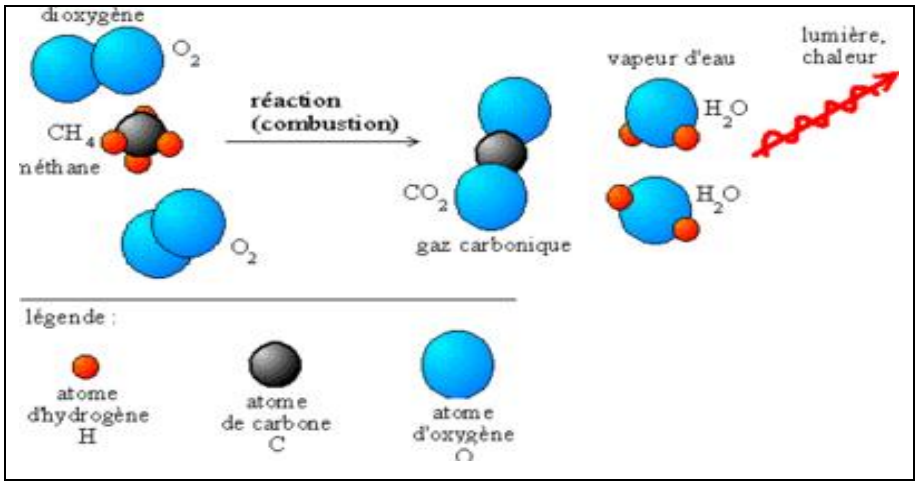


Fig.1.4 : Réaction chimique (échange d'atomes entre les composés, exemple de la combustion du méthane dans le dioxygène).

1.3.9 Propriété physico-chimique de molécule (des protéines)

1.3.9.1 Dénaturation

Une protéine est dénaturée lorsque sa conformation tridimensionnelle spécifique est changée par rupture de certaines liaisons sans atteinte de sa structure primaire. Il peut s'agir, par exemple, de la désorganisation de zones en hélice α. La dénaturation peut être réversible ou irréversible. Elle entraîne une perte totale ou partielle de l'activité biologique. Elle produit très souvent un changement de solubilité de la protéine **[8]**.

Les agents de dénaturation sont nombreux :

- agents physiques : chaleur, radiations, pH ;
- agents chimiques: solution d'urée qui forme de nouvelles liaisons hydrogène dans la protéine, solvants organiques, détergents...

1.3.10 Prédiction de propriété ou d'activité (Relation quantitative structure à activité)

Une relation quantitative structure à activité (en anglais : *Quantitative structure-activity relationship* ou QSAR, parfois désignée sous le nom de relation quantitative structure à propriété - en anglais : *quantitative structure-property relationship* ou QSPR) est le procédé par lequel une structure chimique est corrélée avec un effet bien déterminé comme l'activité biologique ou la réactivité chimique.

Ainsi par exemple l'activité biologique peut être exprimée de manière quantitative, comme pour la concentration de substance nécessaire pour obtenir une certaine réponse biologique. De plus lorsque les propriétés ou structures physico-chimiques sont exprimées par des chiffres, on peut proposer une relation mathématique, ou *relation quantitative structure à activité*, entre les deux.

L'expression mathématique obtenue peut alors être utilisée comme moyen prédictif de la réponse biologique pour des structures similaires.

La QSAR la plus commune est de la forme : activité = f (propriétés physico-chimiques et/ou structurales) **[9]**.

1.4 Vue d'ensemble

La section suivante donne une vue d'ensemble de chemoinformatique, soulignant les problèmes et les solutions communs aux divers sous-domaines plus spécialisés. Ces matières constituent également les divers chapitres du manuel de Chemoinformatique **[10]** et du manuel de Chemoinformatique **[11]**.

1.4.1 Représentation des composés chimiques

Une gamme entière des méthodes pour la représentation sur l'ordinateur des composés et des structures de produit chimique a été développée comprenant des codes linéaires, des tables de raccordement, et des matrices.
Des méthodes spéciales ont dû être conçues pour représenter uniquement une structure chimique, pour percevoir des dispositifs tels que l'aromaticité et pour traiter la stéréochimie, les structures 3D, ou les surfaces moléculaires.

1.4.2 Représentation des réactions chimiques

En manipulant des réactions chimiques il ne suffit pas d'indiquer seulement les produits de départ et les produits d'une réaction mais on doit également indiquer l'emplacement de réaction et les liens cassés et faits dans une réaction. En outre la stéréochimie des réactions doit être manipulée.

1.4.3 Données en chimie

Beaucoup de **connaissances chimiques ont été dérivées des données**. La chimie doit offrir une gamme riche des données sur les propriétés physiques, chimiques, et biologiques, par exemple, données binaires pour la classification, vraies données pour la modélisation, et données spectrales ayant une densité élevée de l'information. Ces données doivent être introduites dans une forme favorable à l'échange d'information facile et analyse de données.

1.4.4 Les sources de données et les bases de données

L'énorme quantité de données en chimie a mené au développement des bases de données pour stocker et disséminer les données en forme électronique. Par exemple, des bases de données ont été développées pour la littérature chimique, composés chimiques, structures 3D, réactions, et spectres. L'Internet est de plus en plus employé pour distribuer des données et l'information en chimie.

1.4.5 Méthodes pour calculer des données physiques et chimiques

Une variété de données physiques et chimiques des composés peut directement être calculée par une gamme des méthodes. Les premiers sont les calculs mécaniques de quantum de divers degrés de sophistication. Cependant, des méthodes simples telles que des arrangements d'additivité peuvent également être employées pour estimer une variété de données avec l'exactitude raisonnable.

1.4.6 Calcul des descripteurs de structure

Dans la plupart des cas, cependant, physique, chimique, ou les propriétés biologiques ne peuvent pas être directement calculés à partir de la structure d'un composé. Dans cette situation, une approche indirecte doit être adoptée : la structure du composé est représentée par des descripteurs de structure par la suite un rapport entre les descripteurs de structure et la propriété est établie en analysant une série de paires de descripteurs de structure et les propriétés associées par les méthodes d'étude inductives (Figure 1.5).

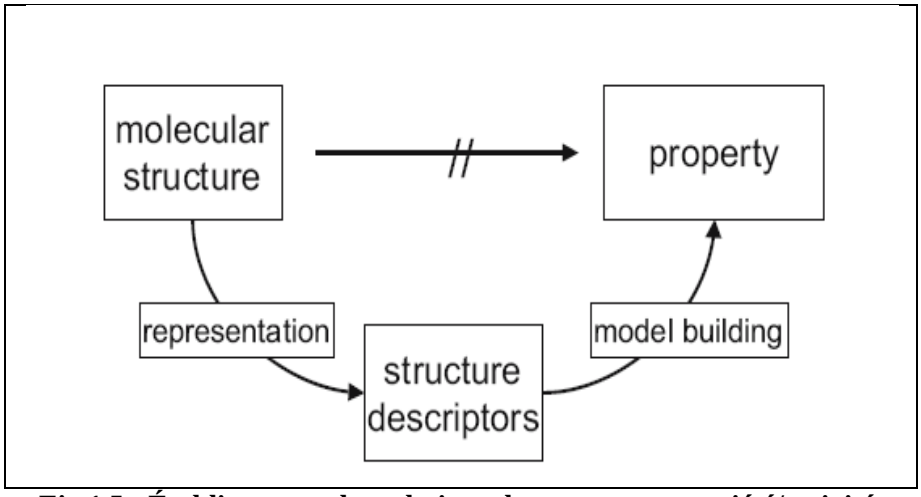


Fig.1.5 : Établissement des relations de structure-propriété/activité.

Une variété de descripteurs de structure ont été développées 1D, 2D, ou information de la structure 3D ou les propriétés de la surface moléculaires.

1.4.7 Méthodes d'analyse de données

Une variété de méthodes pour apprendre des données par des méthodes d'étude inductives sont employées dans la chimie, par exemple, statistiques, méthodes d'identification de modèle, réseaux neurones artificiels, et algorithmes génétiques. Ces méthodes peuvent être classifiées dans des méthodes d'apprentissage supervisé et non supervisé et sont employés pour la classification ou modélisation quantitatif.

1.5 Méthodes et outils

1.5.1 Méthodes de la modélisation moléculaire

La modélisation moléculaire a pour but de prévoir la structure et la réactivité des molécules ou des systèmes de molécules.

Les méthodes de la modélisation moléculaire peuvent être rangées en trois catégories **[12]** :

1.5.1.1 Méthodes quantiques

Ces méthodes sont basées sur le calcul des orbitales moléculaires (OM). Les principales variantes sont :

1. **La méthode de Hückel**
   C'est la plus simple de toutes. Elle ne prend en compte que les électrons p et utilise des approximations assez draconiennes.
2. **Les méthodes de champ auto-cohérent (SCF, Self Consistent Field)**
   Ces méthodes prennent en compte les électrons s et reposent sur des calculs plus élaborés que la méthode de Hückel.
3. **Les méthodes basées sur la fonctionnelle de la densité (DFT, Density Functional Theory)**
   Ces méthodes utilisent une expression de l'énergie électronique E en fonction de la densité électronique r, elle-même fonction de la position r de l'électron : E = f [r(r)]

1.5.1.2 Mécanique moléculaire

Cette technique calcule l'énergie des atomes au moyen d'approximations. La simplification des calculs permet de travailler sur des molécules de grande taille, ou sur des systèmes comportant un grand nombre de molécules.

1.5.1.3 Dynamique moléculaire

Cette technique a pour but de calculer les mouvements des molécules, le plus souvent à partir des énergies de la mécanique moléculaire. Elle permet de simuler l'évolution des systèmes dans le temps.

1.5.2 Méthodes de recherche de médicaments

L'apparition de nouvelles technologies a bouleversé la recherche de nouveaux médicaments dans sa phase initiale. Celle-ci inclut tout d'abord la synthèse et l'isolement de nouvelles molécules puis leur essai sur des systèmes biologiques permettant de présupposer d'un intérêt thérapeutique éventuel. Cette phase était classiquement longue et pénible. La synthèse chimique relevait d'un art difficile ; au départ, le choix d'une structure de base se faisait sans guide. Les essais sur les animaux entiers ou les organes isolés étaient longs et complexes. Au total, malgré des progrès au fil des années, le processus relevait plus de la " pêche à la ligne " que de la démarche rationnelle.

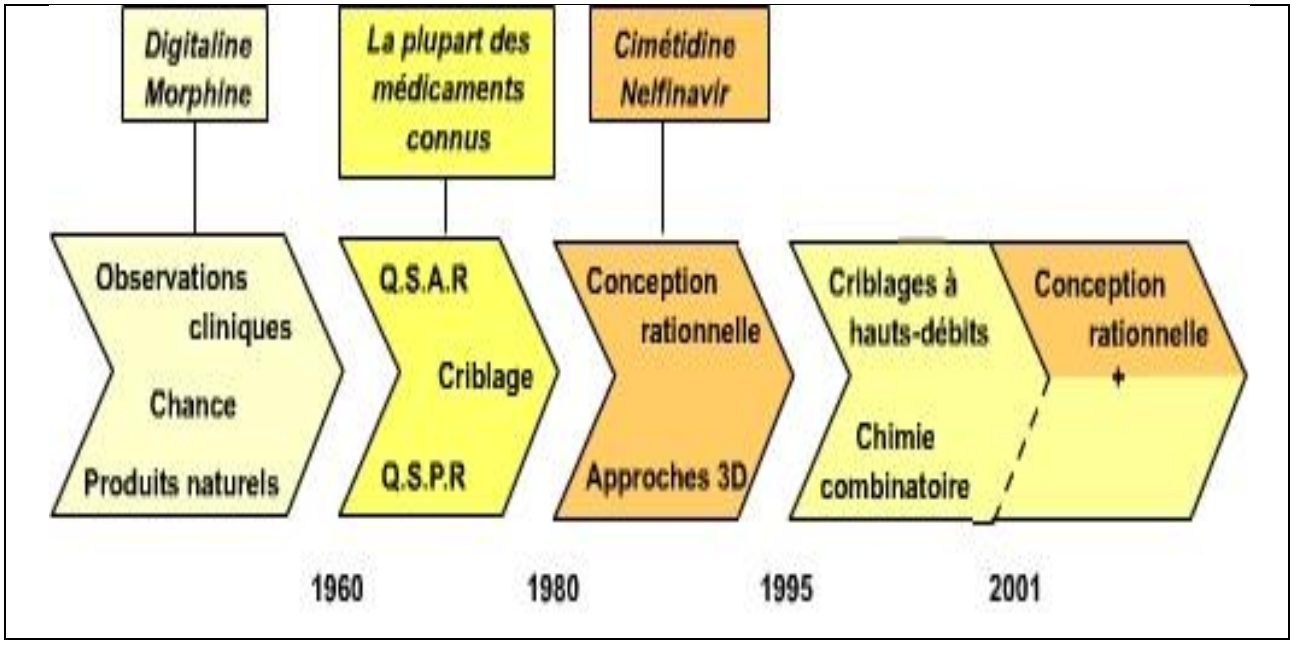


Fig.1.6 : Les approches de recherche des médicaments.

Trois approches ont profondément transformé cette recherche [13]:

1.5.2.1 Les techniques conformationnelles

La théorie des récepteurs postule que c'est l'union de la molécule de médicament avec une macromolécule qui est à l'origine de l'effet pharmacodynamique et plus généralement de la réponse thérapeutique. Cette union est fortement spécifique : seules quelques molécules privilégiées en sont capables. On dit que le médicament est comme " la clé dans la serrure ". On sait maintenant déterminer la conformation dans l'espace des protéines, notamment grâce à la radiocristallographie aux rayons X, donc celle des récepteurs. On peut donc prévoir quelles structures devront présenter les molécules pour pouvoir s'unir à eux. Cette recherche est aidée par les programmes informatiques qui permettent de visualiser les molécules et de les faire tourner dans l'espace (conception assistée par ordinateur).

Bien que hautement sophistiquée et évidemment plus ardue que ces quelques lignes pourraient le laisser croire, cette approche permet de ne plus s'en remettre au hasard dans la recherche des séries chimiques intéressantes. On voit, cependant, qu'il est indispensable de connaître au départ le récepteur pertinent, c'est-à-dire d'avoir une hypothèse physio-pathologique et d'avoir été capable d'identifier et d'isoler la protéine qui le porte. Là aussi, des progrès décisifs ont été faits dans l'isolement des protéines et, mieux encore, dans le repérage et le clonage des gènes qui commandent leur synthèse.

1.5.2.2 La chimie combinatoire

Il est désormais possible de synthétiser en une seule opération plusieurs centaines de molécules c'est ce que l'on appelle la chimie combinatoire. On part de la structure de base déterminée a priori comme il vient d'être dit et on génère systématiquement toutes les variations possibles en greffant des radicaux chimiques, des chaînes latérales, en modifiant le squelette, etc. Ceci se fait non plus étape par étape, mais en mettant en présence les réactifs nécessaires. On obtient ainsi d'un seul coup plusieurs centaines de molécules. Toutes les opérations, synthèse, isolement et identification, sont miniaturisées et robotisées. Le gain de temps et l'abaissement des

coûts sont considérables. On peut ainsi constituer une bibliothèque de plusieurs milliers de dérivés en quelques mois.

1.5.2.3 Le criblage à haut débit

Le problème est alors d'identifier parmi toutes ces molécules celles qui sont pourvues des propriétés biologiques les plus intéressantes. Le gain de temps et l'augmentation de la productivité apportés par la chimie combinatoire l'auraient été en vain si la productivité de cette phase de repérage appelée "criblage" n'avait pas été aussi améliorée. Aux essais longs et limités de la pharmacologie expérimentale classique, a succédé une technique qui permet d'essayer dans le minimum de temps des milliers de molécules.

Le test consiste à mettre en présence la substance à tester et un système biochimique (une enzyme par exemple) et de mesurer l'importance de la réaction éventuelle. L'essai peut être fait simultanément avec un grand nombre de systèmes, de significations très diverses. Tout dépend de ce que l'on met dans les tubes et, une fois de plus, on ne trouvera que ce que l'on cherche. Les systèmes biologiques testés ne sont pas indifférents : ce sont ceux dont on pense qu'ils interviennent de manière cruciale dans le déterminisme de la maladie. L'opération finale est celle du choix. La plupart du temps, toutes les molécules intéressantes ne peuvent pas passer en développement. Il faut donc sélectionner les plus prometteuses, compte tenu de leurs résultats aux tests (Figure 1.7).

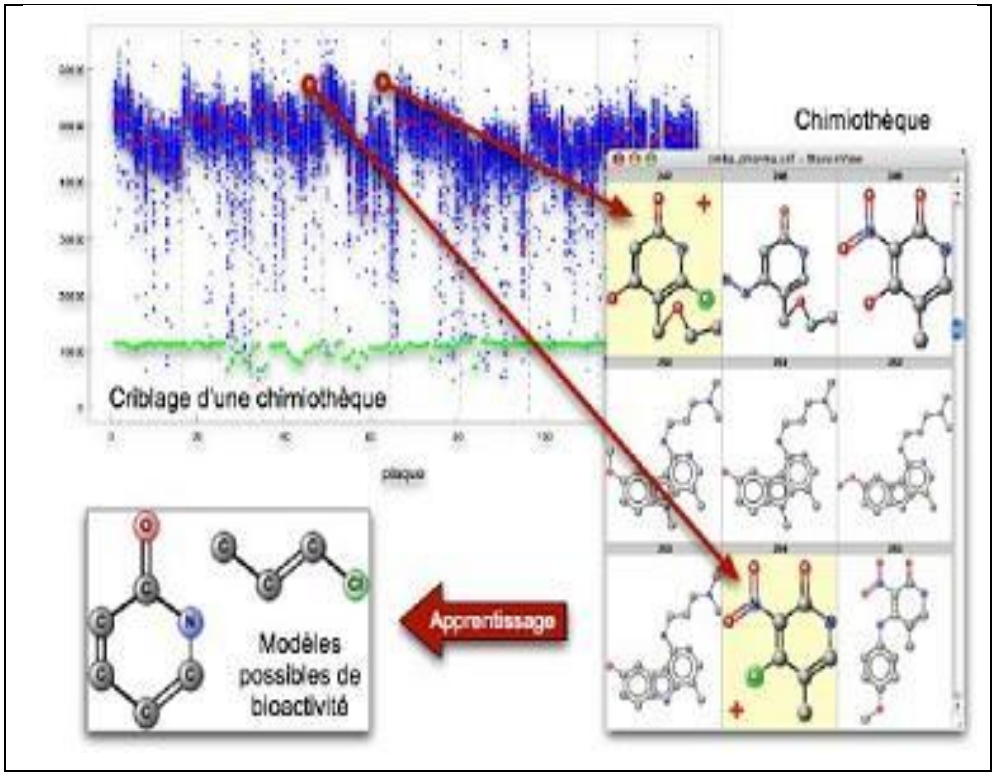


Fig.1.7 : Criblage à haut débit.

1.6 Quelques logiciels

Voici une petite sélection de logiciels pour DOS, Windows et Linux [14]:

1.6.1 Logiciels utilisés pour la représentation moléculaire

- **ChemSketch:** Editeur de formules en 2D et 3D. Optimisation géométrique par mécanique moléculaire.

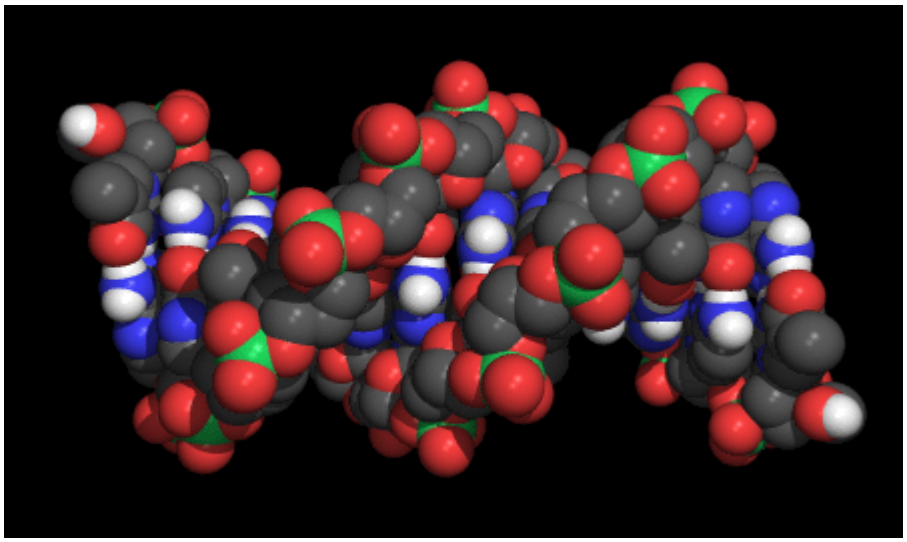


Fig.1.8 : Représentation moléculaire.

- **RasMol**: Visualiseur de molécules.
- **PovChem** : Permet de créer des images de molécules en 3D par la technique du "tracé de rayons" à l'aide du logiciel POV-Ray.
- **Logiciels pour la spectroscopie.**
- **Spartan**: Un logiciel de modélisation en trois dimensions; calcule, entre autres, des énergies, des états de transition, des conformations, offre de nombreuses possibilités de visualisation.

1.6.2 Logiciels utilisés pour les calculs de propriétés

- **MOPAC** : Calculs quantiques semi-empiriques.
- **Open Babel:** Convertisseur de fichiers de coordonnées moléculaires.
- **Tinker** : Ensemble très complet de programmes pour la mécanique et la dynamique moléculaires.
- **Vega** : Ce programme permet de calculer un grand nombre de propriétés moléculaires (volume, lipophilie...), d'analyser des trajectoires de dynamique moléculaire et de réaliser divers traitements sur les fichiers de coordonnées. (Note : la version Windows de Vega contient une copie de MOPAC, permettant de réaliser des calculs quantiques).
- **VMD**: Visual Molecular Dynamics (VMD). Visualiseur de trajectoires de dynamique moléculaire. Le site présente en outre une très belle collection d'images et d'animations.
- **Gaussian 98**: Calculateur très puissant
- **Molden**: Logiciel de visualisation de géométrie, de fréquences harmoniques, d'orbitales moléculaires, permet la représentation de résultats issus de différents logiciels de calculs de chimie.

1.7 Applications de chemoinformatique

1.7.1 Commentaires généraux

L'étendue des applications de chemoinformatique est très large ; en effet, n'importe quel champ de chimie peut profiter de ses méthodes. Nous avons dit que la chemoinformatique est l'application des méthodes d'informatique pour résoudre des problèmes chimiques.

Quel, alors, sont les problèmes principaux considérés par un chimiste ?

Il doit réaliser que la tâche principale de la chimie n'est pas tellement de produire des produits chimiques mais de produire des propriétés, les propriétés qui s'avèrent justement être attachées aux produits chimiques. La société a besoin d'une **variété de propriétés**, par exemple, pour les maladies traitantes, pour des voitures de coloration, pour construire les maisons stables, pour

coller des matériaux ensemble, pour les visages embellissant, pour des vêtements de nettoyage, etc. La première question qu'un chimiste doit répondre est :

Quelle structure j'ai besoin pour obtenir la propriété désirée ?

C'est le secteur de la **structure des propriétés ou** structure des **rapports d'activité**. Une fois qu'on a obtenu une idée sur quelle structure portera la propriété désirée, on doit répondre à la prochaine question :

Comment peux-je synthétiser cette structure ?

C'est le secteur de la **conception de synthèse**. Tout à fait une variété de problèmes doivent être résolues en concevant des synthèses efficaces, problèmes qui concerne le développement court de stratégie de synthèse et prévoir le cours des réactions chimiques.

Une fois une réaction dans un arrangement de synthèse a été exécutée, on doit répondre à la prochaine question :

Ce qui est le produit de la réaction que j'ai exécutée ?

C'est le secteur de **l'élucidation de structure**. Notre connaissance de chimie n'est pas encore assez profonde que nous pouvons toujours être sûrs que la réaction que nous exécutons prend le cours désiré. L'utilisation d'information spectroscopique doit être faite pour élucider la structure du produit de réaction. Tous ces problèmes sont trop complexes pour être résolus par des calculs basés sur les premiers principes. Ils ont tous besoin de beaucoup d'information d'être traité et profondément la connaissance chimique. À ce sens, cependant, les méthodes de chemoinformatique peuvent aider à répondre à ces trois questions fondamentales [13].

1.7.2 Applications de chemoinformatique par secteurs de chimie

Quelques applications typiques de chemoinformatique dans différents secteurs de chimie sont énumérées ci-dessous. Il doit être souligné que cette liste est loin d'être complet [15]!

1.7.2.1 L'information chimique

- Stockage et récupération des structures chimiques et des données associées pour contrôler la pléthore de données.
- Diffusion des données sur l'Internet.
- Édition absolue des données à l'information.

1.7.2.2 Tous les domaines de chimie

- Prévision des propriétés physiques, chimiques, ou biologiques des composés.

1.7.2.3 Chimie analytique

- Analyse des données de la chimie analytique pour faire des prévisions sur la qualité, l'origine, et l'âge des objets étudiés.
- Elucidation de la structure d'un composé basé sur des données spectroscopiques.

1.7.2.4 Chimie organique

- Prévision du cours et des produits des réactions organiques.
- Conception des synthèses organiques.

1.7.2.5 Conception de médicament

- Etablissement de la relation structure-activité.
- Comparaison des bibliothèques chimiques.
- Définition et analyse de diversité structurale.
- Planification des bibliothèques chimiques.
- analyse des voies biochimiques.

Le domaine de chemoinformatique est loin entièrement d'être développé. Il y a beaucoup de secteurs et problèmes qui peuvent encore tirer bénéfice de l'application des méthodes de chemoinformatique. Il y a beaucoup d'espace pour l'innovation en cherchant de nouvelles applications et en développant de nouvelles méthodes.

1.8 Applications

1.8.1 L'utilisation de QSAR et de méthodes informatiques dans la conception de médicament

Le travail de **[16]** décrit la base de QSAR moderne dans la découverte de médicament et présente quelques défis et demandes courants de découverte et d'optimisation des candidats de médicament. Les modèles de QSAR tiennent compte du calcul des propriétés physico-chimiques (par exemple, lipophilicité), de la prédiction de l'activité biologique (ou de la toxicité), aussi bien que l'évaluation de l'absorption, de la distribution, du métabolisme, et de l'excrétion (ADME). Dans la recherche pharmaceutique, QSAR a un intérêt particulier pour les étapes préclinique de la découverte de médicament pour remplacer l'expérimentation pénible et coûteuse, de filtrer de grandes bases de données chimiques, et de choisir des candidats de médicament. Cependant, pour faire partie de stratégies de découverte et de développement de médicament, le besoin de QSARs de répondre à différents critères (par exemple, predictivité suffisant). Ce travail décrit la base de QSAR moderne dans la découverte de médicament et présente quelques défis et demandes courants de découverte et d'optimisation des candidats de médicament

1.8.2 Prévisions confiantes de ségrégation des propriétés des produits chimiques pour le criblage virtuel des médicaments

Le travail de **[17]** présente une méthodologie pour évaluer la confiance en prédiction d'une propriété physico-chimique ou biologique. L'identification des prédictions incertaines de composés est cruciale pour le procédé moderne de découverte de médicament. Cette tâche est accomplie par la combinaison de la méthode de prédiction avec une carte à organisation automatique. De cette façon, la méthode peut isoler des prédictions incertaines aussi bien que des prédictions confiantes. La méthode à quatre ensembles de données différents à été appliqué, et des différences significatives dans les prévisions moyennes ont été obtenu. Cette approche constitue une nouvelle manière pour évaluer la confiance, puisqu'elle recherche non seulement des situations d'extrapolation mais également elle identifie des problèmes d'interpolation.

1.8.3 Classification de composée chimique avec les modèles de structure extraits automatiquement

Le travail de **[18]** propose de nouvelles méthodes de classification de structure chimique basée sur l'intégration de l'exploitation de la base de données de graphe et l'exploitation de données et des fonctions de noyau de graphe de la machine d'apprentissage. Dans cette méthode, ils ont d'abord identifié un ensemble de modèles généraux de graphe dans des données de structure chimique. Ces modèles sont alors employés pour augmenter une fonction de noyau de graphe qui calcule par paires la similitude entre les molécules. La matrice de similitude obtenue est employée comme entrée pour classifier les composés chimiques par l'intermédiaire des machines à noyau telles que la machine à vecteur de support (SVM). Les résultats obtenus indiquent que l'utilisation d'une approche basée-modèle pour la similarité de graphe rapporte des profils d'exécution, et parfois excédant cela des approches existantes de situation actuelle.

1.8.4 Méthode d'arbres à sortie noyau pour la prédiction de sorties structurées et l'apprentissage de noyau

Le travail présenté dans **[19]** propose une **extension** des méthodes d'arbres pour la prédiction de sorties structurées et à l'apprentissage supervisé d'un noyau. Cette extension est basée sur l'utilisation d'un noyau sur la sortie de ces méthodes qui leur permet de construire un arbre à la seule condition qu'un noyau puisse être défini sur l'espace de sortie.

Cet algorithme, appelé OK3 (pour "output kernel trees"), généralise les arbres de classification et de régression ainsi que les méthodes d'ensemble d'arbres. Il hérite de plusieurs caractéristiques de ces méthodes telles que l'interprétabilité, la robustesse aux variables non pertinentes et la résistance à l'échelle sur le nombre d'entrées. Cet algorithme donne de bons résultats sur deux problèmes de nature très différente : un problème de complétion de motif and un problème d'inférence de graphe.

1.8.5 Approche multi-classes de représentation des molécules pour la conception des produits-procédés assistées par ordinateur

La Conception de Produits Assistée par Ordinateur (CPAO) est largement utilisée dans le domaine « Process System Engineering » (PSE), comme un outil puissant pour la recherche de nouveaux produits chimiques. Les étapes cruciales de la CPAO sont la génération des molécules et **l'estimation des propriétés**, particulièrement quand les structures moléculaires complexes comme les arômes sont recherchés.

Le travail présenté dans **[20]** présente une approche multi-classes de représentation des molécules basée sur les graphes moléculaires et la connaissance chimique.

Trois catégories de groupes fonctionnels sont proposées : groupes élémentaires, groupes de base et groupes composés. Ces derniers servent à générer quatre classes de représentation qui peuvent être utiles pour la **prédiction des propriétés** et dans la conception des molécules (CAMD). La méthodologie est utile pour intégrer des méthodes de contribution de groupes dans les simulateurs où certaines molécules ne sont pas référencées.

Cette méthodologie peut être aussi utile pour le développement de méthodes de contribution de groupes se basant sur une décomposition automatique.

1.8.6 Relation structure moléculaire-Odeur (Utilisation des Réseaux de Neurones pour l'estimation de l'Odeur Balsamique)

Le travail de **[21]** présente une approche de **prédiction de l'odeur** des molécules **basée sur les descripteurs moléculaires.** Les techniques d'analyse en composantes principales (ACP) et d'analyse de colinéarité permettent d'identifier les descripteurs les plus pertinents. Un réseau de neurones supervisé à deux couches (cachée et sortie) est employé pour corréler la structure moléculaire à l'odeur. Un ensemble de paramètres est modifié jusqu'à la satisfaction de la meilleure régression. Le réseau neurologique corrèle d'une manière satisfaisante les molécules avec leur odeur assignée, basée sur des descripteurs moléculaires suffisamment nombreux et divers. Mais il ne peut pas prévoir l'odeur balsamique et ses sous-notes.

1.8.7 Une distance d'écoulement de réseau entre les graphes étiquetés

Le travail présenté dans **[22]** propose une mesure **de similarité originale** entre les graphes étiqueté **qui a des** applications à l'analyse de donnée structurée, par exemple : chemical informatics, web document clustering, etc. Les métriques exactes sur des graphes basés sur de sous-graphe isomorphisme ont été proposés plus tôt mais en raison du manque d'un efficace algorithme, ils ne peuvent pas être appliqués sur de grandes données. La métrique proposé basé sur les graphes **exploite** la similitude de contexte de sommet **et calcule** des points assortis globaux dans le temps polynômial dans la taille des graphes en utilisant une formulation d'écoulement de réseau du problème. Cette métrique est employée, dans un cadre distinctif **pour prévoir les propriétés** chimiques comme la cancérogénicité et la mutagénicité des molécules et les examiner sur des ensembles de données de PTC et de MUTAG. Les grains définis positifs construit en utilisant cette métrique présente une exécution améliorée de manière significative des grains existants finis pour des graphes sur la plupart des ensembles de données démontrant l'efficacité de la technique.

1.8.8 La fouille de graphes dans les bases de données réactionnelles au service de la synthèse en chimie organique

Le problème auquel s'intéresse le travail présenté dans **[23]** est la découverte de nouvelles familles de réactions chimiques à partir de bases de données de réactions, et montre en quoi ce problème peut se reformuler en un problème particulier de fouille de graphes. La découverte de nouvelles réactions présente un grand intérêt pour la synthèse en chimie organique, discipline dont le but est la conception de molécules complexes à partir de composants chimiques usuels et de réactions. En effet, plus un expert de la synthèse a de réactions à sa disposition, plus il peut créer de nouveaux produits à partir d'un ensemble donné de molécules et plus il peut optimiser le plan de synthèse d'une molécule cible donnée.

1.8.9 Regrouper des molécules : Influence des mesures de similitude

Le travail présenté dans [24] présente les résultats d'une étude expérimentale pour analyser l'effet de diverses mesures de similitude (ou distance) sur la qualité de regroupement d'un ensemble de molécules.il se concentre principalement sur les approches de regroupement capables de traiter directement la représentation 2D des molécules (c.-à-d., des graphes). Dans un tel contexte, il semble approprié d'employer une approche basée sur des mesures asymétriques de similitude.

Plusieurs d'autres travaux ont été énoncés dans [25].

1.9 Conclusion

Nous avons analysé le concept de chemoinformatique sous l'angle de sa modélisation. Il nous a été donné de constater que dans ce domaine, d'énorme volume d'information produite par la recherche en chimie ne peut être traitée et analysée que par les moyens informatiques.

Dans ce chapitre nous avons présenté la définition de la chemoinformatique en tant que domaine de la science qui consiste en l'application de l'informatique aux problèmes relatifs à la chimie. Par la suite nous avons présenté les concepts de bases de la chimie, ainsi qu'une vue d'ensemble de chemoinformatique, soulignant les problèmes et les solutions communs aux divers sous-domaines plus spécialisés.

Ensuite des Méthodes de la modélisation moléculaire qui ont pour but de prévoir la structure et la réactivité des molécules ou des systèmes de molécules ont été présentées, ainsi qu'une petite sélection de logiciels utilisés pour la représentation moléculaire et les calculs de propriétés.
En fin quelques applications typiques de chemoinformatique dans différents secteurs de chimie ont été énumérées.
Nous montrerons dans le chapitre suivant les techniques de modélisation par apprentissage qui ont permis la mise en place de nombreuses méthodes ; elles reposent pour la plupart sur la recherche d'une relation entre un ensemble de nombres réels, descripteurs de la molécule, et la propriété ou l'activité que l'on souhaite prédire.

La chemoinformatique est fondamentalement basée sur la modélisation, cette dernière est donc une activité essentielle en vue d'effectué automatiquement la classification, la prédiction,…etc.

Les premiers essais de modélisation d'activités de molécules datent de la fin du 19ème siècle, lorsque Crum-Brown et Frazer **[26]** postulèrent que l'activité biologique d'une molécule est une fonction de sa constitution chimique. Mais ce n'est qu'en 1964 que furent développés les modèles de "contribution de groupes", qui constituent les réels débuts de la modélisation QSAR. Depuis, l'essor de nouvelles techniques de modélisation par apprentissage, linéaires d'abord, puis non linéaires, ont permis la mise en place de nombreuses méthodes ; elles reposent pour la plupart sur la recherche d'une relation entre un ensemble de nombres réels, descripteurs de la molécule, et la propriété ou l'activité que l'on souhaite prédire. Nous montrerons tout d'abord comment les molécules peuvent être représentées par des vecteurs de réels, et comment ces descripteurs sont sélectionnés. Nous introduirons ensuite les outils de modélisation sans contrainte les plus utilisés, c'est-à-dire la régression linéaire multiple et la régression non linéaire à l'aide de réseaux de neurones, qui sont fondés sur le calcul de descripteurs. Nous présenterons le problème de la sélection de modèle, ainsi que les stratégies les plus efficaces pour le résoudre. Enfin, d'autres méthodes de modélisation, telles que la méthode CoMFA, mises au -point pour la modélisation d'activités biologiques, seront présentées.

2.1 Les descripteurs

De nombreuses recherches ont été menées, au cours des dernières décennies, pour trouver la meilleure façon de représenter l'information contenue dans la structure des molécules, et ces structures elles-mêmes, en un ensemble de nombres réels appelés descripteurs ; une fois que ces nombres sont disponibles, il est possible d'établir une relation entre ceux-ci et une propriété ou activité moléculaire, à l'aide d'outils de modélisation classiques. Ces descripteurs numériques réalisent de ce fait un codage de l'information chimique en un vecteur de réels. On en dénombre aujourd'hui plus de 3000 types, qui quantifient des caractéristiques physico-chimiques ou structurelles de molécules. Ils peuvent être obtenus de manière empirique ou non-empirique, mais les descripteurs calculés, et non

mesurés, sont à privilégier : ils permettent en effet d'effectuer des prédictions sans avoir à synthétiser les molécules, ce qui est un des objectifs de la modélisation. Il existe cependant quelques descripteurs mesurés : il s'agit généralement de données expérimentales plus faciles à mesurer que la propriété ou l'activité à prédire (coefficient de partage eau-octanol **[27]**, polarisabilité, ou potentiel d'ionisation).

Avant toute modélisation, il est nécessaire de calculer ou de mesurer un grand nombre de descripteurs différents, car les mécanismes qui déterminent l'activité d'une molécule ou une de ses propriétés sont fréquemment mal connus. Il faut ensuite sélectionner parmi ces variables elles qui sont les plus pertinentes pour la modélisation.

2.1.1 Les descripteurs moléculaires

Nous allons présenter les descripteurs moléculaires les plus courants, en commençant par les descripteurs les plus simples, qui nécessitent peu de connaissances sur la structure moléculaire, mais véhiculent peu d'informations. Nous verrons ensuite comment les progrès de la modélisation moléculaire ont permis d'accéder à la structure 3D de la molécule, et de calculer des descripteurs à partir de cette structure.

Les descripteurs 1D sont accessibles à partir de la formule brute de la molécule (par exemple C_6H_6O pour le phénol), et décrivent des propriétés globales du composé. Il s'agit par exemple de sa composition, c'est-à-dire les atomes qui le constituent, ou de sa masse molaire. On peut remarquer que ces descripteurs ne permettent pas de distinguer les isomères de constitution.

Les descripteurs 2D sont calculés à partir de la formule développée de la molécule. Ils peuvent être de plusieurs types.

- Les ***indices constitutionnels*** caractérisent les différents composants de la molécule.

Il s'agit par exemple du nombre de liaisons simples ou multiples, du nombre de cycles…

- Les ***indices topologiques*** peuvent être obtenus à partir de la structure 2D de la molécule, et donnent des informations sur sa taille, sa forme globale et ses ramifications. Les plus fréquemment utilisés sont l'indice de Wiener **[28]**, l'indice de Randié **[29]**, l'indice de connectivité de valence de Kier-Hall **[30]** et l'indice de Balaban **[31]**. L'indice de Wiener permet de caractériser le volume moléculaire et la ramification d'une molécule : si l'on appelle distance topologique entre deux atomes le plus petit nombre de liaisons séparant ces deux atomes, l'indice de Wiener est égal à la somme de toutes les distances topologiques entre les différentes paires d'atomes de la

molécule. L'indice de Randié est un des descripteurs les plus utilisés ; il peut être interprété comme une mesure de l'aire de la molécule accessible au solvant.

Ces descripteurs 2D reflètent bien les propriétés physiques dans la plupart des cas, mais sont insuffisants pour expliquer de façon satisfaisante certaines propriétés ou activités, telles que les activités biologiques. Des descripteurs, accessibles à partir de la structure 3D des molécules, ont pu être calculés grâce au développement des techniques instrumentales et de nouvelles méthodes théoriques.

Les descripteurs 3D d'une molécule sont évalués à partir des positions relatives de ses atomes dans l'espace, et décrivent des caractéristiques plus complexes; leurs calculs nécessitent donc de connaître, le plus souvent par modélisation moléculaire empirique ou *ab initio*, la géométrie 3D de la molécule. Ces descripteurs s'avèrent donc relativement coûteux en temps de calcul, mais apportent davantage d'informations, et sont nécessaires à la modélisation de propriétés ou d'activités qui dépendent de la structure 3D. On distingue plusieurs familles importantes de descripteurs 3D :

- Les ***descripteurs géométriques*** les plus importants sont le volume moléculaire, la surface accessible au solvant, le moment principal d'inertie.
- Les ***descripteurs électroniques*** permettent de quantifier différents types d'interactions inter- et intramoléculaires, de grande influence sur l'activité biologique de molécules. Le calcul de la plupart de ces descripteurs nécessite la recherche de la géométrie pour laquelle l'énergie stérique est minimale, et fait souvent appel à la chimie quantique. Par exemple, les énergies de la plus haute orbitale moléculaire occupée et de la plus basse vacante sont des descripteurs fréquemment sélectionnés.

Le moment dipolaire, le potentiel d'ionisation, et différentes énergies relatives à la molécule sont d'autres paramètres importants.

− ***Descripteurs spectroscopiques*** : les molécules peuvent être caractérisées par des mesures spectroscopiques, par exemples par leurs fonctions d'onde vibrationnelles.

En effet, les vibrations d'une molécule dépendent de la masse des atomes et des forces d'interaction entre ceux-ci ; ces vibrations fournissent donc des informations sur la structure de la molécule et sur sa conformation. Les spectres infrarouges peuvent être obtenus soit de manière expérimentale, soit par calcul théorique, après recherche de la géométrie optimale de la molécule. Ces spectres sont alors codés en vecteurs de descripteurs de taille fixe. Le

descripteur EVA [32] est ainsi obtenu à partir des fréquences de vibration de chaque molécule. Les descripteurs de type *MoRSE* [33] *(Molecule Representation of Structures based on Electron diffraction)* sont calculés à partir d'une simulation du spectre infrarouge ; ils font appel au calcul des intensités théoriques de diffraction d'électrons.

2.1.2 Réduction du nombre de variables

Un grand nombre de descripteurs différents sont collectés pour la modélisation d'une grandeur donnée, car les facteurs déterminants du processus étudié ne sont a priori pas connus. Cependant, les descripteurs envisagés n'ont pas tous une influence significative sur la grandeur modélisée, et les variables ne sont pas toujours mutuellement indépendantes. De plus, le nombre de descripteurs, c'est-à-dire la dimension du vecteur d'entrée, détermine la dimension du vecteur des paramètres à ajuster. Si cette dimension est trop importante par rapport au nombre d'exemples de la base d'apprentissage, le modèle risque d'être surajusté à ces exemples, et incapable de prédire la grandeur modélisée sur de nouvelles observations.
Il est donc nécessaire de réduire la dimension des variables d'entrée. Plusieurs approches sont possibles pour résoudre ce problème :

- réduire la dimension de l'espace des entrées ;
- remplacer les variables corrélées par de nouvelles variables synthétiques, obtenues à partir de leurs combinaisons ;
- sélectionner les variables les plus pertinentes.

Nous allons maintenant décrire les méthodes les plus fréquemment utilisées.

2.1.2.1 L'analyse en composantes principales

L'analyse en composantes principales (ou ACP) [34], est une technique d'analyse de données utilisée pour réduire la dimension de l'espace de représentation des données.
Contrairement à d'autres méthodes de sélection, celle-ci porte uniquement sur les variables, indépendamment des grandeurs que l'on cherche à modéliser. Les variables initiales sont remplacées par de nouvelles variables, appelées composantes principales, deux à deux non corrélées, et telles que les projections des données sur ces composantes soient de variance maximale. Elles peuvent être classées par ordre d'importance.
Considérons un ensemble de n observations, représentées chacune par p données. Ces observations forment un nuage de n points dans R^p.

Le principe de l'ACP est d'obtenir une représentation approchée des variables dans un sous-espace de dimension *k* plus faible, par projection sur des axes bien choisis ; ces axes principaux sont ceux qui maximisent l'inertie du nuage projeté, c'est-à-dire la moyenne pondérée des carrés des distances des points projetés à leur centre de gravité. La maximisation de l'inertie permet de préserver au mieux la répartition des points. Dès lors, les *n* composantes principales peuvent être représentés dans l'espace sous-tendu par ces axes, par une projection orthogonale des *n* vecteurs d'observations sur les *k* axes principaux. Puisque les composantes principales sont des combinaisons linéaires des variables initiales, l'interprétation du rôle de chacune de ces composantes reste possible. Il suffit en effet de déterminer quels descripteurs d'origine leur sont le plus fortement corrélés.

Les variables obtenues peuvent ensuite être utilisées en tant que nouvelles variables du modèle. Par exemple, la régression sur composantes principales **[35]** (ou PCR) est une méthode de modélisation dont la première étape est une analyse en composantes principales, suivie d'une régression linéaire multiple.

2.1.2.2 La méthode de régression des moindres carrés partiels

La régression des moindres carrés partiels **[36,37]** (MCP, ou PLS) est également une méthode statistique utilisée pour construire des modèles prédictifs lorsque le nombre de variables est élevé et que celles-ci sont fortement corrélées. Cette méthode utilise à la fois des principes de l'analyse en composantes principales et de la régression multilinéaire. Elle consiste à remplacer l'espace initial des variables par un espace de plus faible dimension, sous-tendu par un petit nombre de variables appelées « variable latentes », construites de façon itérative. Les variables retenues sont orthogonales (non corrélées), et sont des combinaisons linéaires des variables initiales. Les variables latentes sont obtenues à partir des variables initiales, mais en tenant compte de leur corrélation avec la variable modélisée, contrairement aux variables résultant de l'analyse en composantes principales. Elles doivent ainsi expliquer le mieux possible la covariance entre les entrées et la sortie. Elles sont alors les nouvelles variables explicatives d'un modèle de régression classique, telles que la régression linéaire multiple.

2.2 Modélisation par optimisation sans contrainte

La modélisation par apprentissage consiste à trouver le jeu de paramètres qui conduit à la meilleure approximation possible de la fonction de régression, à partir des couples entrées / sorties constituant l'ensemble d'apprentissage. Le plus souvent, ces couples sont constitués d'un ensemble de vecteurs de variables (descripteurs dans le cas de molécules) $\{x^i, i = 1...N\}$, et d'un ensemble de mesures de la grandeur à modéliser $\{y(x^i), i = 1...N\}$.

La détermination des valeurs de ces paramètres nécessite la mise en œuvre de méthodes d'optimisation qui diffèrent selon le type de modèle choisi. Nous allons tout d'abord présenter les principaux types de modèles faisant appel à l'optimisation paramétrique, sans contrainte, qui consiste à déterminer les paramètres optimaux par minimisation directe d'une fonction de coût par rapport aux paramètres du modèle.

2.2.1 Réseaux de neurones

Les réseaux de neurones formels [38] étaient, à l'origine, une tentative de modélisation mathématique des systèmes nerveux, initiée dès 1943 par McCulloch et Pitts [39].

Un ***neurone formel*** est une fonction non linéaire paramétrée, à valeurs bornées, de variables réelles. Le plus souvent, les neurones formels réalisent une combinaison linéaire des entrées reçues, puis appliquent à cette valeur une « fonction d'activation » *f*, généralement non linéaire. La valeur obtenue y est la sortie du neurone. Un neurone formel est ainsi représenté sur la Figure 2.1.

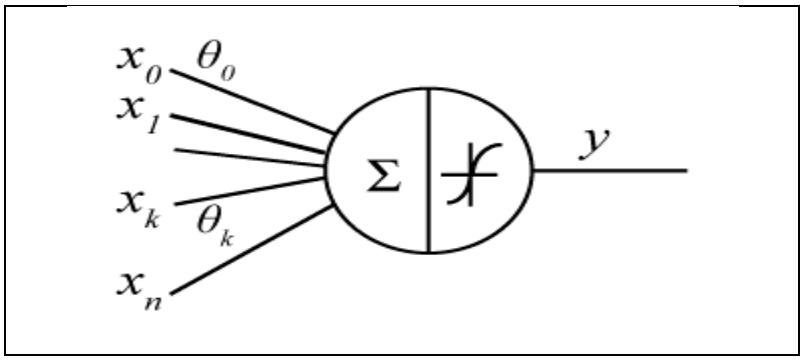


Fig.2.1 : Représentation d'un neurone formel

Les $\{x_k\}_{k=1\ldots n}$ sont les variables, ou ***entrées*** du neurone, et les $\{\theta_k\}_{k=0\ldots n}$ sont les ***paramètres***, également appelés synapses ou poids. Le paramètre θ_0 est le paramètre associé à une entrée fixée à 1, appelée biais. L'équation du neurone est donc :

$$y = f(\theta_0 + \sum_{k=1}^{n} \theta_k x_k) \qquad (1)$$

Les fonctions d'activation les plus couramment utilisées sont la fonction tangente hyperbolique, la fonction sigmoïde et la fonction identité.

Les neurones seuls réalisent des fonctions assez simples, et c'est leurs compositions qui permettent de construire des fonctions aux propriétés particulièrement intéressantes. On appelle ainsi ***réseau de neurones*** une composition de fonctions « neurones » définies ci-dessus.

La Figure 2.2 représente un réseau de neurones non bouclé, organisé en couches (perceptron multicouche), qui comporte N_e variables, une couche de N_c neurones cachés, et N_s neurones de sortie.

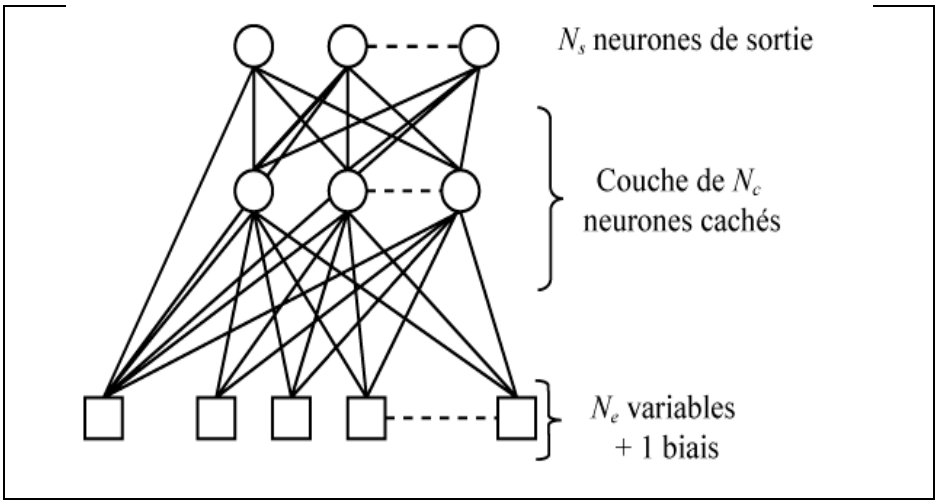


Fig.2.2 : Représentation d'un réseau de neurones

À chaque connexion est associé un paramètre. Les sorties du réseau sont donc des fonctions non-linéaires de ses variables et de ses paramètres. Le nombre de degrés de liberté, c'est-à-dire de paramètres ajustables, dépend du nombre de neurones de la couche cachée ; il est donc possible de faire varier la complexité du réseau en augmentant ou en diminuant le nombre de neurones cachés.

2.2.1.1 Propriétés des réseaux de neurones

Les réseaux de neurones ont pour but de modéliser des processus, à partir d'exemples de couples entrées / sorties. Ils ont la propriété d'***approximation universelle*** : un réseau de neurones comportant un nombre fini de neurones cachés, de même fonction d'activation, et un neurone de sortie linéaire, est capable d'approcher uniformément, avec une précision arbitraire, toute fonction bornée suffisamment régulière, sur un domaine fini de l'espace de ses variables. De plus, il s'agit d'***approximateurs parcimonieux*** : une approximation par un réseau de neurones nécessite en général moins de paramètres que les approximateurs usuels.
Le nombre de paramètres nécessaires pour obtenir une précision donnée augmente en effet linéairement avec le nombre de variables pour un réseau de neurones, alors qu'il croît exponentiellement pour un modèle linéaire par rapport aux paramètres. Cette propriété est très importante, car les réseaux de neurones demandent de ce fait moins d'exemples que d'autres approximateurs pour l'apprentissage.

2.2.1.2 Apprentissage des réseaux de neurones

Considérons un ensemble d'apprentissage, constitué de *N* couples entrées / sorties, c'est-à- dire d'un ensemble de variables $\{\mathbf{x}^i, i = 1...N\}$ et d'un ensemble de mesures de la grandeur à modéliser $\{y(\mathbf{x}^i), i = 1...N\}$. Pour une complexité donnée (le choix de cette complexité est étudié dans la section 2.2.2 de ce chapitre), l'apprentissage s'effectue par minimisation de la fonction de coût des moindres carrés, définie par :

$$j(\theta) = \frac{1}{2} \sum_{i=1}^{N} [y(x^i) - g(x^i, \theta)]^2 \qquad (2)$$

La minimisation de cette fonction s'effectue par une descente de gradient. Cet algorithme a pour but de converger, de manière itérative, vers un minimum de la fonction de coût, à partir de valeurs initiales des poids aléatoires. À chaque étape, le gradient de la fonction est calculé, à l'aide de l'algorithme de ***rétropropagation***. Puis les paramètres sont modifiés en fonction de ce gradient, dans la direction de la plus forte pente, vers un minimum local de *J*. Cette descente peut être effectuée suivant plusieurs méthodes : gradient simple ou méthodes du second ordre, dérivées de la méthode de Newton. Les méthodes du second ordre, généralement plus efficaces, sont les plus utilisées. La procédure de minimisation est arrêtée

lorsqu'un critère est satisfait : le nombre maximal d'itérations est atteint, la variation du module du vecteur des paramètres ou du gradient de la fonction de coût est trop faible…

2.2.2 Sélection du modèle

La modélisation vise à fournir un modèle qui soit non seulement ajusté aux données d'apprentissage, mais aussi capable de prédire la valeur de la sortie sur de nouveaux exemples, c'est-à-dire de généraliser.

Soit R_i l'erreur commise par le modèle considéré sur l'exemple i (également appelé résidu) :

$$R_i = y(x^i) - g(x^i, \theta) \quad (3)$$

où y (x^i) est la valeur mesurée de la grandeur à modéliser pour l'exemple i, et g(x^i ,θ) est l'estimation du modèle pour ce même exemple. L'erreur du modèle sur les données d'apprentissage, E_A , peut être évaluée par l'erreur quadratique moyenne en apprentissage, appelée également EQMA :

$$E_A = \sqrt{\frac{1}{N_A}\sum_{i=1}^{N_A}(R_i)^2} \quad (4)$$

où N_A est le nombre d'exemples servant à établir le modèle. La qualité du modèle en apprentissage est souvent visualisée sur un *diagramme de dispersion*, sur lequel sont portées les valeurs de la grandeur d'intérêt estimées par le modèle, en fonctions des valeurs mesurées de cette grandeur. La qualité de la modélisation est d'autant meilleure que les points de ce graphique sont proches de la première bissectrice. L'ajustement des points à cette droite peut être évalué par le ***coefficient de détermination*** :

$$R^2 = \frac{\sum_{i=1}^{N_A}(y^i - \overline{y})^2}{\sum_{I=1}^{N_A}(g(x^i,\theta) - \overline{y})^2} \quad (5)$$

Où $\overline{y}$ est la moyenne des valeurs mesurées. Ce coefficient est égal au rapport de la variance expliquée à la variance totale de la sortie. Plus il est proche de 1, plus la corrélation entre les

valeurs mesurées et prédites est forte. Les valeurs intermédiaires renseignent sur le degré de dépendance linéaire entre les deux variables.

L'erreur sur la base d'apprentissage seule n'est pas un bon indicateur de la qualité du modèle ; un bon modèle doit être capable de rendre compte de la relation déterministe entre les variables et la grandeur modélisée, sans s'ajuster au bruit des données d'apprentissage. Il convient donc de trouver comment estimer cette capacité.

2.2.2.1 Méthode générale de sélection

Lors de la modélisation, nous recherchons à la fois la complexité la mieux adaptée au problème étudié, et les paramètres du modèle correspondant. Le modèle doit en effet être de complexité suffisante pour rendre compte de la relation entre les variables explicatives et la grandeur modélisée, mais il ne doit pas être trop complexe, afin de ne pas être trop sensible au bruit présent dans les données. En effet, lorsque la complexité du modèle envisagé augmente, on observe généralement une diminution du coût d'apprentissage, car le modèle s'ajuste de lus en plus précisément aux données d'apprentissage. En revanche, l'erreur de généralisation diminue pour atteindre un minimum, puis augmente avec la complexité du modèle, car celui-ci est alors surajusté aux données d'apprentissage, et au bruit présent dans ces données. Ce compromis est également connu sous le nom de dilemme biais-variance : le biais caractérise l'écart entre les estimations et les mesures, et la variance reflète l'influence du choix de la base d'apprentissage sur le modèle **[40]**. De plus, lorsque les modèles envisagés ne sont pas linéaires par rapport à leurs paramètres, la sélection du modèle optimal ne consiste pas uniquement à choisir la meilleure famille de fonctions, comme c'est le cas pour des modèles linéaires (modèles polynomiaux ou combinaisons linéaires de fonctions non paramétrées). La sélection s'effectue donc souvent en plusieurs étapes.

- Pour les modèles non linéaires en leurs paramètres, la première sélection s'effectue au sein d'une famille de fonctions donnée. En effet, la fonction de coût n'étant pas quadratique en les paramètres du modèle, elle ne possède pas un minimum unique.
  L'optimisation de cette fonction peut donc conduire à des minima différents, donc à différents modèles de même complexité. Il est donc nécessaire de réaliser plusieurs apprentissages, pour une complexité donnée, et de choisir celui qui est susceptible de posséder les meilleures propriétés de généralisation.

- Il faut ensuite sélectionner, parmi les meilleurs modèles de complexités différentes, celui qui présente les meilleures capacités de généralisation.

Nous allons maintenant détailler les critères de sélection les plus fréquemment utilisés, en étudiant dans un premier temps comment l'erreur de généralisation peut être évaluée.

2.2.2.2 Sélection du modèle par estimation de l'erreur de généralisation

L'erreur de généralisation ne peut pas être évaluée exactement ; la base de données disponible est en effet de taille limitée, et la distribution de probabilité des données, dont dépend cette erreur, est généralement inconnue. Il est donc nécessaire de trouver une approximation de cette erreur de généralisation. Nous allons ainsi présenter les méthodes les plus utilisées pour effectuer ces sélections.

La méthode la plus commune, dite méthode de validation simple ou hold-out, consiste à construire, à partir de l'ensemble des données, deux ensembles : les paramètres du modèle sont ajustés sur la ***base d'apprentissage***, et l'erreur de généralisation est évaluée sur la ***base de validation.***

L'erreur en validation est définie de façon analogue au coût d'apprentissage :

$$E_V = \sqrt{\frac{1}{N}\sum_{i=1}^{N_V}(R_i)^2} \qquad (6)$$

où NV est le nombre d'exemples dans la base de validation.

Pour des modèles de complexité donnée, la méthode de sélection consiste à effectuer plusieurs apprentissages à partir de différentes initialisations des paramètres, lorsque l'on a affaire à un modèle non linéaire en ses paramètres. Il faut alors choisir, parmi les modèles obtenus, celui qui fournit la plus faible erreur en validation. Il s'agit d'une méthode très rapide, mais d'efficacité limitée, surtout lorsque le nombre d'exemples disponible est faible. En effet, si le nombre d'exemples est petit, la variance de l'estimation de l'erreur de généralisation est grande ; le coût de validation dépend alors fortement du choix de la base de validation [41].

De plus, les exemples de la base de validation ne sont pas utilisés pour l'estimation des paramètres du modèle, et peuvent faire défaut : le modèle n'étant pas ajusté à certains types d'exemples, ses capacités de généralisation sont limitées. Par conséquent, l'erreur de

généralisation est plus souvent estimée à l'aide d'autres méthodes, telles que la validation croisée ou le leave-one-out.

2.2.2.2.1 La validation croisée

L'ensemble des N exemples disponibles est cette fois-ci divisé en K sous-ensembles disjoints. La valeur $K = 10$ est souvent retenue, mais il n'y a pas de méthode de choix optimal de K. On construit alors une base d'apprentissage à partir de $K - 1$ de ces sous ensembles.
Le modèle est ajusté sur cette base, puis l'erreur de prédiction R_i est calculée pour chaque exemple du sous-ensemble restant. Cette étape est réalisée K fois, en choisissant un sous-ensemble de validation différent à chaque itération. Ainsi, chaque exemple se trouve une fois et une seule dans une base de validation. Cette procédure est illustrée sur la Figure 2.3, où $K = 5$. Les sous-ensembles $E_1 \ldots E_5$ constituent successivement la base de validation.

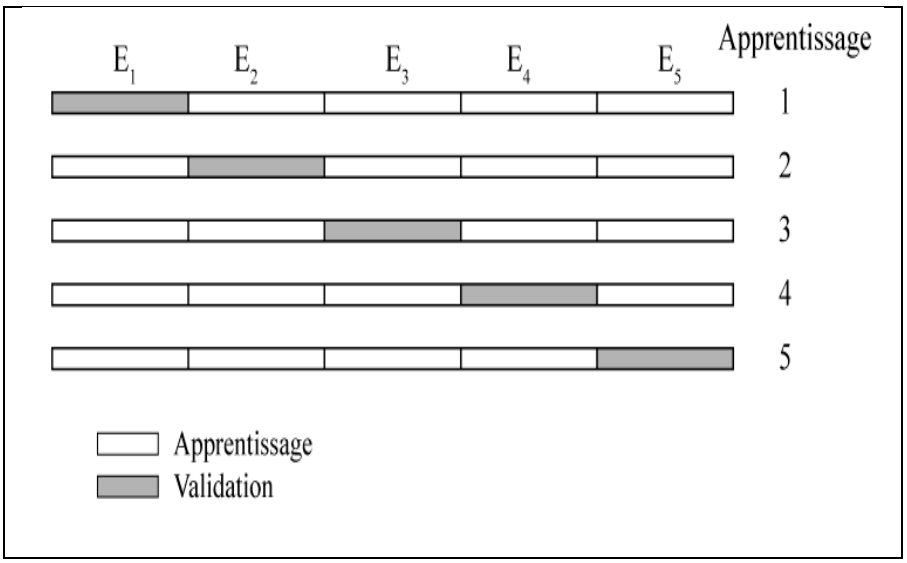


Fig.2.3 : Principe de la validation croisée.

La performance en généralisation de la famille de modèles est ensuite estimée en calculant le score de validation croisée :

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(R_i)^2} \qquad (7)$$

Cette technique a l'avantage d'être moins sensible au choix des bases de validation (la variance de l'erreur est plus faible), et permet d'utiliser toutes les données en apprentissage.

Elle nécessite cependant d'effectuer K apprentissages, qui produisent ainsi K modèles différents. Une fois la meilleure complexité déterminée, on effectue un apprentissage avec l'ensemble des données disponibles.

2.2.2.2.2 Méthode du leave-one-out

Le cas particulier où $K = N$ (N est le nombre d'exemples disponibles) est appelé ***leave-one-out*** : à chaque itération, un exemple i est extrait de l'ensemble d'apprentissage. Une série d'apprentissage est réalisée à l'aide des $N - 1$ exemples restants, et l'erreur de prédiction sur l'exemple i est calculée pour chacun des modèles obtenus. On retient l'erreur la plus faible, notée R_i^{-i} . Le score de leave-one-out est alors défini par :

$$E_t = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(R_i^{-i})^2} \qquad (8)$$

Ce score E_t est un estimateur non-biaisé de l'erreur de généralisation [42]. Cette méthode présente ainsi l'avantage de tirer pleinement profit des données disponibles, et d'être indépendante du choix de bases de validation. Cependant, le temps de calcul peut devenir très grand, car il est nécessaire de procéder à N séries d'apprentissage.

2.2.2.3 Le leave-one-out virtuel

La méthode du leave-one-out virtuel [43] consiste à estimer les erreurs R_i^{-i} à partir d'un seul apprentissage réalisé sur l'ensemble des N exemples, ce qui permet d'estimer le score de leave-one-out sans avoir à réaliser N apprentissages. Cette estimation est appelée score de ***leave-one-out virtuel***, et repose sur le calcul de la matrice jacobienne du modèle g (x,θ*), obtenu à partir des N exemples. Un développement de ce modèle au premier ordre par rapport aux paramètres, au voisinage de θ*, est :

$$g(x,\theta) \cong g(x,\theta^*) + Z(\theta - \theta^*) \qquad (9)$$

Ce développement fait apparaître **Z**, matrice jacobienne du modèle. Cette matrice est de taille (N, q), où q est le nombre de paramètres du modèle. Les N éléments de la colonne j sont égaux aux dérivées partielles de la sortie par rapport au paramètre j. Les éléments de ***Z*** s'écrivent donc :

$$(Z)_{ij} = \left(\frac{\partial g(x_i,\theta_{mc})}{\partial \theta_j}\right)_{\theta=\theta_{mc}} \quad (10)$$

Les modèles dont la jacobienne n'est pas de rang plein, c'est-à-dire de rang inférieur à *q*, doivent être rejetés. En effet, cela signifie qu'au moins deux paramètres ont des effets qui ne sont pas indépendants sur la sortie. Ces modèles comportent donc trop de paramètres, et ont de grandes chances d'être surajustés.

Considérons la matrice de projection orthogonale sur le sous-espace défini par les colonnes de la matrice $\mathbf{Z}$: $\boldsymbol{H} = \boldsymbol{Z}(\boldsymbol{Z^T Z})^{-1}\boldsymbol{Z^T}$ qui est une matrice carrée *(N, N)*.

Les termes diagonaux de cette matrice sont les ***leviers de plan tangent*** (appelés simplement leviers dans la suite du mémoire) des exemples. Le levier d'un exemple i est ainsi égal à :

$$h_{ii} = \boldsymbol{z}^{iT}(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{z}^i \quad (11)$$

où $\mathbf{z}^i$ est le vecteur dont les composantes sont celles de la i-ième ligne de **Z**.

Le calcul de ces leviers permet d'estimer l'erreur de prédiction sur un exemple i lorsqu'il est retiré de la base d'apprentissage (c'est-à-dire R_i^{-i}), à partir de l'erreur commise sur cet exemple lorsqu'il est dans cette base (c'est-à-dire R_i) :

$$R_i^{-i} \cong \frac{R_i}{1-h_{ii}} \quad (12)$$

Ce résultat est d'ailleurs exact si le modèle est linéaire ; il est alors appelé PRESS (Predicted REsidual Sum of Squares).

Dès lors, il est possible de calculer le score de leave-one-out virtuel, qui constitue une très bonne approximation de l'erreur de généralisation si le développement limité au premier ordre sur lequel elle est fondée est suffisamment précis :

$$E_p = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{R_i}{1-h_{ii}}\right)^2} \quad (13)$$

Ce calcul permet non seulement d'estimer rapidement l'erreur de généralisation, mais également de révéler d'éventuels surajustements du modèle à des exemples particuliers. Il

peut ainsi constituer la base d'une méthode de planification expérimentale : s'il est possible d'enrichir la base de données de mesures supplémentaires, celles-ci doivent être effectuées au voisinage des points qui ont des leviers importants.

2.2.2.4 Sélection de modèle à l'aide des leviers:

2.2.2.4.1 Interprétation des leviers:

Nous avons vu que les leviers sont les termes diagonaux de la matrice de projection orthogonale sur le sous-espace des colonnes de la matrice jacobienne. Lorsque la matrice **Z** est de rang plein, les leviers vérifient les propriétés suivantes :

$$\begin{cases} \sum_{i=1}^{N} h_{ii} = q \\ 0 < h_{ii} < 1 \quad \forall i \in [\![1, N]\!] \end{cases} \tag{14}$$

où *q* est le nombre de paramètres du modèle. Le levier h_{ii} relatif à un exemple *i* peut ainsi être interprété comme la proportion des paramètres utilisée par le modèle pour s'ajuster à l'exemple *i*. Par conséquent, si tous les exemples ont la même influence sur le modèle, les leviers sont tous égaux à $\frac{q}{N}$. Si le levier d'un exemple est proche de 1, le modèle est particulièrement ajusté sur cet exemple, et il est presque parfaitement appris. Par contre, l'erreur en prédiction sur cet exemple lorsqu'il n'appartient pas à la base d'apprentissage, qui peut être estimée par $R_i^{-i} \cong \frac{R_i}{1-h_{ii}}$, est très grande. En revanche, un exemple dont le levier est proche de 0 n'a aucune influence sur le modèle. Si un modèle consacre une grande partie de ses degrés de liberté à un ou plusieurs exemples, il doit être rejeté, car il est susceptible d'être ajusté au bruit présent dans ces mesures [44]. Les leviers constituent donc un outil supplémentaire de sélection.

2.2.2.4.2 Calcul des intervalles de confiance

Les leviers permettent également le calcul des intervalles de confiance sur les prédictions du modèle. Un intervalle de confiance au seuil de confiance $(1-\alpha)$ est un intervalle tel qu'on peut dire qu'il contient la vraie valeur de la grandeur prédite avec une probabilité $(1-\alpha)$. L'étendue de l'intervalle de confiance caractérise la confiance que l'on peut avoir dans la prédiction faite par le modèle : plus l'étendue est faible, plus cette confiance est grande.

Lorsque le modèle est non-linéaire, un intervalle de confiance approché pour l'espérance mathématique de la sortie Y_p peut être calculé à l'aide des leviers.
Pour un exemple *i* de la base d'apprentissage, son expression est :

$$E\left(Y_p \middle| x^i\right) \in g\left(x^i, \theta\right) \pm t_{\propto}^{N-q} \, s\sqrt{z^{iT}(Z^T Z)^{-1} z^i} = g\left(x^i, \theta\right) \pm t_{\alpha}^{N-q} \, s\sqrt{h_{ii}} \qquad (15)$$

Où g($\mathbf{x}^i$,$\boldsymbol{\theta}$) est la prédiction du modèle pour cet exemple, $t_{\propto}^{N-q}$ est la valeur d'une variable de Student à *N*-*q* degrés de liberté et un niveau de confiance (1−α), et *s* une estimation de la variance de l'erreur de prédiction de modèle.
L'intervalle de confiance peut également être estimé pour un exemple qui n'appartient pas à la base d'apprentissage :

$$E^{(-i)}\left(Y_p \middle| x^i\right) \in \; g\left(x^i, \theta\right) \pm t_{\propto}^{N-q-1} \; s^{-i}\sqrt{\frac{h_{ii}}{1-h_{ii}}} \qquad (16)$$

Cet intervalle est donc d'autant plus large que h_{ii} est proche de 1 : dans ce cas, la prédiction sur cet exemple est très peu fiable.
Les leviers se révèlent donc être un outil efficace pour détecter le surajustement de modèles à des exemples particuliers, et pour repérer ces exemples. Ils permettent la sélection d'un modèle parmi des modèles de complexités différentes, lorsque leurs performances en généralisation sont comparables : le meilleur est généralement celui pour lequel la distribution des leviers des exemples autour de $\frac{q}{N}$ est la plus étroite. La sélection de modèle peut donc se dérouler de la façon suivante :

- Pour des modèles de complexité donnée, on réalise plusieurs apprentissages, à partir d'initialisations différentes.
- Les modèles dont la matrice jacobienne n'est pas de rang plein sont écartés ; on calcule alors les scores d'apprentissage et de leave-one-out virtuel des modèles obtenus. Le modèle présentant les meilleures capacités de généralisation est retenu.
- Cette sélection est effectuée pour des modèles de complexité croissante. On compare alors les modèles retenus pour chaque complexité, c'est-à-dire leurs scores d'apprentissage et de leave-one-out virtuel, ainsi que la répartition des leviers des exemples. Ces critères permettent de sélectionner le meilleur modèle.

Les performances de ce modèle peuvent alors être mesurées sur une base de test : elle est composée d'exemples n'ayant pas servi à établir ni à choisir le modèle. Le score de test permet ainsi d'évaluer les performances du modèle en généralisation.

2.3 Autres méthodes de QSPR/QSAR

La modélisation d'une propriété ou d'une activité moléculaire nécessite de disposer d'informations caractérisant les molécules, informations à partir desquelles la grandeur en question est prédite. Il peut s'agir de descripteurs, mais il existe des méthodes alternatives de caractérisation des molécules. Nous présenterons dans un premier temps la méthode de contribution de groupes, qui, bien que datant des débuts de la modélisation QSAR, est toujours utilisée pour des applications particulières. Nous décrirons ensuite comment les techniques de calcul et de comparaison de champs moléculaires se révèlent particulièrement efficaces pour la modélisation d'activités biologiques. Nous introduirons finalement des méthodes récentes qui, de façon analogue aux *graph machines*, considèrent que les structures des molécules contiennent des informations dont il est possible de tirer directement profit pour la modélisation.

2.3.1 Méthode de contribution de groupes

Les méthodes de contribution de groupes consistent à évaluer une propriété en décomposant la molécule en un ensemble de groupes fonctionnels, et en sommant les contributions relatives à des fragments de molécules **[45,46]**. Ces contributions sont déterminées à partir d'une base d'exemples de molécules, dont les valeurs de la propriété sont connues. Plusieurs types de groupes fonctionnels peuvent être définis. Ils sont généralement organisés en un système hiérarchique :

- Les ***groupes d'ordre 0*** sont des atomes, et le calcul d'une propriété est effectué en sommant les contributions de chacun des atomes de la molécule considérée.
- La décomposition en ***groupes d'ordre 1*** consiste à découper la molécule en groupes 'atomes (tels que -CH2-, -CH3 ou -OH). Leurs contributions à une propriété donnée sont sommées sans que l'environnement de chacun des groupes dans la molécule ne soit pris en considération. Ainsi, le groupe -CH_2- a une contribution fixe, qu'il soit relié à un carbone ou à un groupe oxygéné. Ces groupes sont assez souvent employés,

car ils permettent d'estimer rapidement la valeur d'une propriété, avec une précision parfois suffisante (par exemple pour l'enthalpie de formation). Cependant, les résultats obtenus, pour la température d'ébullition par exemple, ne sont pas toujours satisfaisants. De plus, certains isomères peuvent conduire à la même décomposition : il est alors impossible de les distinguer par cette méthode.

- ***Les groupes d'ordre 2*** **[47,48]** sont constitués des atomes centraux de la molécule (autres que H), accompagnés de leurs plus proches voisins, c'est-à-dire de tous les atomes auxquels ils sont reliés. Contrairement aux groupes d'ordre 1, ceux d'ordre 2 tiennent compte de l'environnement des atomes.

Le Tableau 1 présente une comparaison des décompositions des molécules de butan -2-ol et 2-

Méthylpropan -1-ol, représentées sur la Figure 2.4.

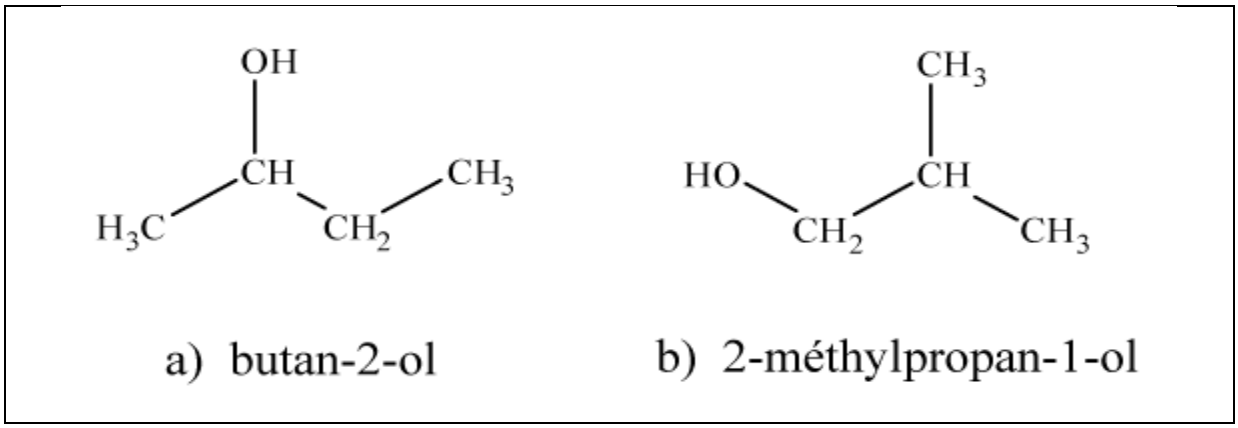


Fig.2.4 : Exemple de deux molécules, isomères de constitution

Méthode	Groupe	Nombre de groupes a)	Nombre de groupes b)
Ordre 0	C	4	4
	H	10	10
	O	1	1
Ordre 1	CH	1	1
	$-CH_2-$	1	1
	$-CH_3$	2	2
	-OH	1	1
Ordre 2	$C-(C)(H)_3$	2	2
	$C-(C)_2(H)_2$	1	0
	$C-(C)(O)(H)_2$	0	1
	$C-(C)_3(H)$	0	1
	$C-(C)_2(O)(H)$	1	0

Tableau 1 : Décomposition en groupes de deux isomères de constitution

On observe que seule la décomposition en groupes d'ordre 2 permet de distinguer ces deux molécules.

De nombreuses méthodes s'appuient donc sur des groupes des trois ordres pour améliorer la précision des prédictions et différencier les isomères. Elles sont principalement utilisées pour prédire des propriétés thermodynamiques, par exemples des propriétés critiques (température ou pression critique) et de nombreuses grandeurs énergétiques. Elles présentent également l'avantage de permettre l'estimation de propriétés de mélanges, par addition des contributions des composants du mélange.

2.3.2 Analyse comparative de champs moléculaires (CoMFA)

La méthode CoMFA (pour Comparative Molecular Field Analysis) a été développée à partir de 1988 par Cramer **[49]**. Elle est en particulier utilisée pour la modélisation d'activités biologiques, pour lesquelles les méthodes classiques se révèlent parfois peu performantes.

L'activité biologique d'une molécule dépend généralement de son interaction avec un récepteur donné. La modélisation de cette activité peut donc être réalisée en calculant les

interactions de chaque molécule (ligand) avec ce récepteur, et en établissant une relation entre ces interactions et l'activité étudiée. Cette modélisation, qui repose sur le calcul de potentiels d'interactions moléculaires, s'effectue en plusieurs étapes.

2.3.2.1 Recherche des conformations moléculaires les plus stables

Des programmes tels que Concord ou Corina **[50]** permettent d'accéder à une conformation de basse énergie, à partir de laquelle sont créées plusieurs conformations bioactives.

2.3.2.2 Alignement des ligands

Les potentiels d'interactions moléculaires dépendent des orientations des structures utilisées pour leur calcul. La comparaison des potentiels de différentes molécules nécessite donc un alignement préalable de ces structures. Il s'agit d'une étape assez délicate, en particulier lorsque les composés ont des structures diverses. Cette étape est même susceptible de limiter les possibilités d'application de la méthode. L'alignement est réalisé par rapport aux groupes fonctionnels identifiés comme potentiellement pharmacophores. Il existe plusieurs méthodes pour réaliser cet alignement. La plus simple est la méthode de superposition par sous-structures, qui consiste à superposer les molécules qui partagent un squelette commun, selon ce squelette. L'approche par superposition de pharmacophores ne nécessite pas cette supposition, mais part du fait que les pharmacophores de chaque molécule sont identifiés. Il s'agit ensuite de maximiser la superposition de ces groupements entre les molécules. Il est également possible d'aligner les molécules par rapport à leurs moments dipolaires ou à leurs champs électrostatiques.

2.3.2.3 Calcul des champs électrostatiques et stériques

Les champs électrostatiques et stériques sont ensuite évalués. Pour cela, on définit autour de chaque molécule, préalablement alignée, une grille 3D. Puis on calcule, en chaque point de la grille, l'énergie d'interaction de la molécule avec un atome sonde.

2.3.2.4 Corrélation entre les champs et les activités biologiques

Les valeurs calculées des champs peuvent alors être utilisées comme variables du modèle. Le nombre de ces variables peut être très grand (quelques milliers), lorsque la résolution de la grille est fine par rapport à la taille des molécules. Il n'est alors pas possible de corréler directement ces valeurs de champs avec l'activité étudiée. La modélisation est généralement effectuée par la méthode des moindres carrés partiels, et le nombre optimum de variables final déterminé par validation croisée.

2.3.2.5 Visualisation graphique des résultats

Une visualisation graphique des résultats permet enfin de situer autour d'une molécule les régions pour lesquelles une substitution augmente ou diminue l'affinité envers le récepteur. Cette méthode a montré son efficacité en QSAR, en particulier pour la modélisation d'interactions ligand-protéine. Elle permet en effet de décrire efficacement les interactions ligand-récepteur, car les propriétés des ligands sont calculées à partir de leurs conformations bioactives. Les problèmes sont néanmoins multiples :

- Les résultats de modélisation dépendent de la conformation bioactive choisie. Or, la recherche de cette dernière est parfois difficile, en particulier lorsque la molécule est flexible : il existe alors un nombre important de conformations possibles près du minimum d'énergie.
- Il n'existe pas de règle générale pour l'alignement des molécules. Il s'agit d'un défaut majeur, car le modèle est très sensible à cet alignement.
- De plus, le temps de calcul est généralement assez long (30 à 60 min pour une vingtaine de molécules possédant de 30 à 50 atomes).

Puisque l'alignement des molécules est une limitation de la méthode CoMFA, de nouvelles techniques ne nécessitant pas cette étape ont été développées. La **méthode CoMSA** (pour Comparative Molecular Surface Analysis), ne repose pas sur la comparaison de champs calculés en une série donnée de points, mais celle du potentiel électrostatique moyen de régions définies de la surface de la molécule. Ces régions sont nombreuses, et la méthode des cartes auto-organisatrices de Kohonen **[51]** permet de réduire la dimension des données tout en préservant leur topologie. Cette méthode est particulièrement adaptée pour transformer la surface tridimensionnelle de la molécule en une carte bidimensionnelle du potentiel électrostatique. La transformation de Kohonen permet en effet à la fois de diminuer la taille

des données, et de retrouver les données 3D à partir de leur représentation 2D. Les vecteurs obtenus sont alors analysés et corrélés à l'activité étudiée, par la méthode des moindres carrés partiels.

2.4 Conclusion

De nombreuses recherches ont été menées, au cours des dernières décennies, pour trouver la meilleure façon de représenter l'information contenue dans la structure des molécules, et ces structures elles-mêmes, en un ensemble de nombres réels appelés descripteurs ; une fois que ces nombres sont disponibles, il est possible d'établir une relation entre ceux-ci et une propriété ou activité moléculaire, à l'aide d'outils de modélisation classiques.

Dans ce chapitre nous avons commencé par la présentation des descripteurs moléculaires les plus courants, en commençant par les descripteurs les plus simples, qui nécessitent peu de connaissances sur la structure moléculaire, mais véhiculent peu d'informations. Nous avons vu ensuite comment les progrès de la modélisation moléculaire ont permis d'accéder à la structure 3D de la molécule, et de calculer des descripteurs à partir de cette structure.
Par la suite nous avons présenté les techniques de modélisation par apprentissage qui consiste à trouver le jeu de paramètres qui conduit à la meilleure approximation possible de la fonction de régression, à partir des couples entrées / sorties constituant l'ensemble d'apprentissage,

Ensuite des Méthodes de sélection de modèle qui vise à fournir un modèle qui soit non seulement ajusté aux données d'apprentissage, mais aussi capable de prédire la valeur de la sortie sur de nouveaux exemples (c.-à-d. généraliser) ont été présentées ainsi que d'autres méthodes de QSPR/QSAR tel que la méthode de contribution de groupes.

Les techniques alternatives que nous venons de décrire répondent donc généralement à des besoins précis, mais elles ont pour cette raison des domaines d'application limités. Nous montrerons dans le chapitre suivant que la méthode des *graph machines*, tout en s'affranchissant des inconvénients liés à l'utilisation de descripteurs, ne présentent pas cette limitation.

Les méthodes traditionnelles de modélisation que nous avons présentés, tels que les réseaux de neurones, réalisent pour la plupart une association entre un vecteur de nombres réels, qui constituent les variables du modèle, et un vecteur de sorties également réelles. Cependant, dans de nombreux problèmes de modélisation, les données se présentent sous la forme de structures dont il faut extraire un vecteur de réels si l'on veut avoir recours à des modèles classiques. Cette étape nécessite de sélectionner les données pertinentes relatives au problème et de les calculer à partir des structures ; elle se traduit par une perte d'information. Il semblerait donc plus pertinent de tirer directement profit de la structure des données, par l'intermédiaire d'un autre type de modèle, capable d'établir de façon directe une association entre ces structures et un vecteur de sorties. Les données structurées se représentent aisément sous la forme de graphes. Nous commencerons par présenter les graphes en tant qu'objets mathématiques, ensuite nous introduirons les mémoires récursives auto-associatives, premiers modèles à réaliser un codage de données structurées par apprentissage artificiel. Nous présenterons les *graph machines* ainsi que leur apprentissage. En fin un exemple de modélisations de propriétés physico-chimiques de molécule sera présenté.

3.1 Les graphes (définition et caractéristiques)

La théorie des graphes date du 18ème siècle, et s'est considérablement développée au 20ème siècle, car elle permet de résoudre de nombreux types de problèmes [52]. Nous allons présenter les bases de cette théorie, et les définitions essentielles relatives aux graphes.

3.1.1 Graphes simples

Un graphe simple G est un groupe {S, A} où :

- S est ensemble d'objets { s_1,s_2 ,...,s_n }, appelés sommets du graphe ;
- A est un sous-ensemble de S × S, dont les éléments { a_1,a_2 ,...,a_m } sont les arêtes du graphe.

Un graphe est ***connexe*** s'il est possible, à partir de n'importe quel sommet, de rejoindre tous les autres. L'arête $a_k = (s_i, s_j)$ est incidente aux nœuds s_i et s_j, qui sont dits adjacents. Le degré d'un nœud est défini comme le nombre d'arêtes qui lui sont incidentes. Un ***cycle*** est une chaîne $(s_1, s_2, \ldots, s_k)$ dont le premier et le dernier sommet sont identiques et tous les autres sommets distincts. Si le graphe possède *m* arêtes $\{ a_1, a_2, \ldots, a_m \}$, on peut faire correspondre à tout cycle μ un vecteur $\mu = (\mu_1, \mu_2; \ldots\ldots \mu_m)$ tel que :

$$\mu_\iota = \begin{cases} 1 \; si \; ai \in \mu \\ 0 \; si \; ai \notin \mu \end{cases} \qquad (17)$$

On dit que *p* cycles $(\mu^1, \mu^2, \ldots, \mu^p)$ sont dépendants s'il existe entre leurs vecteurs associés une relation vectorielle de la forme :

$$\sum_{i=1}^{p} \lambda_i \mu^i = 0 \qquad (18)$$

telle que $(\lambda_1, \lambda_2, \ldots, \lambda_p) \in \mathbb{R}^p$ et $(\lambda_1, \lambda_2, \ldots, \lambda_p) \neq (0,0,\ldots,0)$.

On appelle ***distance*** entre deux sommets la longueur de la plus courte chaîne reliant ces sommets ; le ***diamètre*** d'un graphe est alors la plus grande distance entre les sommets d'un graphe.

3.1.2 Graphes orientés

Un ***graphe orienté*** est un groupe G = {S, A}, où A est un ensemble d'arcs de la forme (s_i, s_j), pour lequel l'arc part de s_i et arrive en s_j. Le degré entrant d^- d'un sommet est le nombre d'arcs qui arrivent à ce sommet, et le degré sortant d^+ le nombre d'arcs qui en partent.

Un ***arbre*** est un graphe connexe sans cycle. Dans un arbre orienté, si les sommets s_i et s_j sont reliés par l'arc (s_i, s_j), s_i est ***parent*** du nœud s_j, qui est un ***enfant*** du nœud s_i.

Les nœuds qui possèdent à la fois des nœuds parents et enfants sont appelés des ***branches***. Un ***nœud racine*** est un nœud sans parent, tandis que les ***feuilles*** sont les nœuds sans enfant. Un arbre est un arbre *n*-aire si tous les nœuds de l'arbre ont au plus *n* successeurs.

Une ***arborescence*** {S, A, r} de racine r est un graphe {S, A} où r est un élément de S tel

que, pour tout sommet s, il existe un unique chemin d'origine r et d'extrémité s. On montre facilement que, dans une arborescence, la racine r n'admet pas de prédécesseur, et que tout sommet différent de r admet un seul prédécesseur.

La Figure 3.1, qui représente une arborescence, illustre les concepts que nous venons de définir.

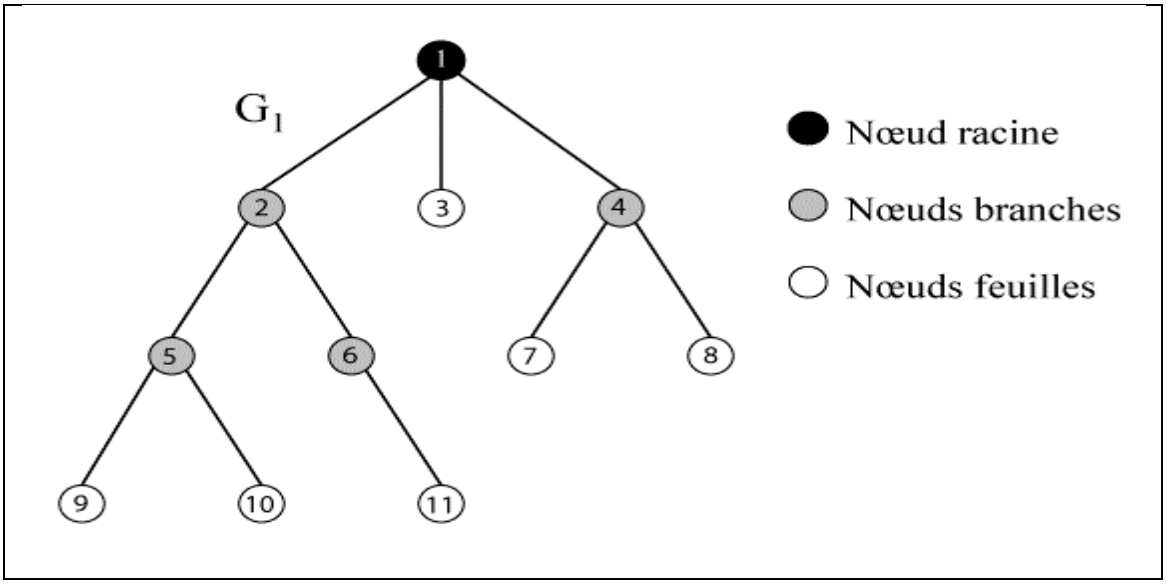


Fig.3.1 : Exemple d'arborescence

3.1.3 Graphes étiquetés

Il est possible d'affecter aux nœuds et aux arcs d'un graphe orienté un ensemble d'étiquettes : le graphe est alors ***étiqueté***. Étant donné un ensemble fini d'étiquettes de sommets L_V, et un ensemble fini d'étiquettes d'arcs L_A, un graphe étiqueté est défini par un triplet (S, r_S, r_A) où :

- S est l'ensemble des sommets ;
- $r_S \subseteq S \times L_S$ est l'ensemble des couples (s_i, l_s) tels que le sommet s_i a pour étiquette l_s ;
- $r_A \subseteq S \times S \times L_A$ est l'ensemble des triplets (s_i, s_j, l_a) tels que l'arc (si ,s j) a pour étiquette l_a.

3.1.4 Matrice d'adjacence :

Plusieurs représentations sont possibles pour les graphes. Elles ne sont pas équivalentes, et le choix d'une représentation adaptée au problème à résoudre permet d'obtenir une meilleure efficacité des algorithmes utilisés. On distingue ainsi la représentation par matrice d'adjacence, par matrice d'incidence sommets-arcs (ou sommets-arêtes), et par liste d'adjacence. Nous utilisons la matrice d'adjacence, car cette représentation permet de calculer simplement certaines caractéristiques des graphes, telles que la distance entre les sommets. La matrice d'adjacence d'un graphe G = {S, A} est la matrice M(G) dont les coefficients $m_{i,j}$ sont définis par :

$$m_{i,j} = \begin{cases} 1 \text{ si } (s_i , s_j) \in A \text{ , c'est} - \text{à} - \text{dire si les noeuds } s_i \text{ et } s_j \text{ sont agjacents} \\ 0 \text{ si } (s_i , s_j) \notin A \end{cases} \quad (19)$$

La Figure 3.2 représente un graphe ainsi que la matrice d'adjacence qui lui correspond.

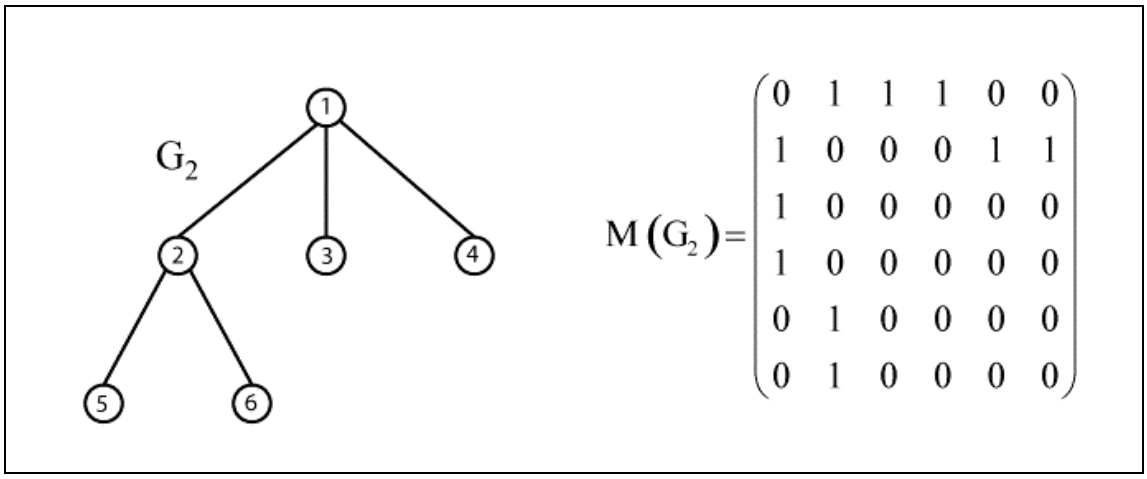


Fig.3.2 : Représentation d'un graphe par sa matrice d'adjacence

Si *n* est le nombre de nœuds du graphe, cette matrice est de taille (*n*, *n*), et chaque ligne ainsi que chaque colonne correspond à un nœud particulier. Puisque la matrice dépend de la numérotation des nœuds du graphe, la représentation n'est pas unique (il existe *n*! possibilités). Il est cependant possible de passer d'une représentation à une autre par permutation de lignes et de colonnes.

La matrice d'adjacence permet d'énumérer les chemins du graphe, et d'en déduire la connexité, la cyclicité ainsi que les distances entre les nœuds. On utilise pour cela le produit

matriciel. Par exemple, la matrice $\mathbf{M}^2$ a pour coefficients :

$$(\mathbf{M}^2)_{i,j} = \sum_{k=1}^{n} m_{i,k}, m_{k,j}$$

Chaque composant de la somme est donc non nul ssi il existe une arête de s_i à s_k et de s_k à s_j, c'est-à-dire s'il existe un chemin de longueur 2 reliant s_i à s_j et passant par s_k. Par extension, on peut démontrer que si **M** est la matrice d'adjacence d'un graphe G dont les sommets sont numérotés de 1 à *n*, le nombre de chemins de longueur exactement l allant de s_i à s_j est le coefficient (*i, j*) de la matrice M^l.
Un algorithme permet alors de calculer la distance entre deux nœuds s_i et s_j :

$$L_{i,j} = min\,(\boldsymbol{M}^l)_{i,j} \neq 0 \qquad (20)$$

On en déduit le diamètre du graphe :

$$D = \max_{(s_i s_j)\in S^2} L_{i,j} \qquad (21)$$

Il est par ailleurs possible de déterminer la connexité d'un graphe, c'est-à-dire le nombre de composantes connexes, et d'en déduire le nombre de cycles indépendants que comporte le graphe :

$$N_{cycles} = N_{arêtes} - N_{noeuds} + connexité \qquad (22)$$

Cette représentation des graphes permet ainsi d'accéder aisément aux principales caractéristiques d'un graphe, ainsi que d'effectuer des opérations telles que la suppression d'une arête ou d'un cycle.

3.2 Apprentissage à partir de graphes : RAAMs et LRAAMs

Nous allons maintenant expliquer comment il est possible d'établir une relation entre un ensemble de données structurées, représentées par des graphes, et un ensemble de nombres réels, réalisant ainsi l'association structures-données réelles évoquée. Les premiers essais de modélisation de données structurées remontent aux mémoires associatives dynamiques. Ces essais ont par la suite conduit aux Mémoires Auto-Associatives Récursives (RAAMs), qui permettent de représenter de façon compacte les arbres, et aux RAAMs étiquetés (LRAAMs).

3.2.1 Les Mémoires Auto-Associatives Récursives

Afin de montrer la capacité des réseaux de neurones à manipuler des données structurées, Pollack a proposé un modèle, fondé sur l'idée d'une mémoire associative entre une structure et un vecteur de taille fixe [53]. Les RAAMs (pour *Recursive AutoAssociative Memory*) et leurs dérivées sont des modèles établis à partir de réseaux de neurones, qui apprennent un codage d'une structure en un vecteur, puis sont à même de décoder la structure d'origine à partir de ce vecteur avec une perte d'information minimale.

L'élément de base des RAAMs est un codeur-décodeur constitué par un réseau de neurones possédant autant de variables que de sorties ($N_e = N_s$), et dont le nombre de neurones cachés est inférieur au nombre de variables ($N_c < N_e$). Un exemple d'un tel codeur-décodeur est représenté sur la Figure 3.3. À l'issue de l'apprentissage, le système réalise une auto-association, car les sorties du modèle sont identiques aux variables : la base d'apprentissage est du type ($\{\mathbf{x}_k, \mathbf{x}_k\}$, $1 \leq k \leq n$).

Lorsque le problème admet une solution, les neurones cachés réalisent un codage du vecteur de variables **x** sous forme compacte **f**(**x**) (car $N_c < N_e$), où **f**(**x**) désigne le vecteur des sorties des neurones cachés. Ce vecteur **f**(**x**) peut ensuite être décodé, afin de retrouver en sorties les données d'entrée.

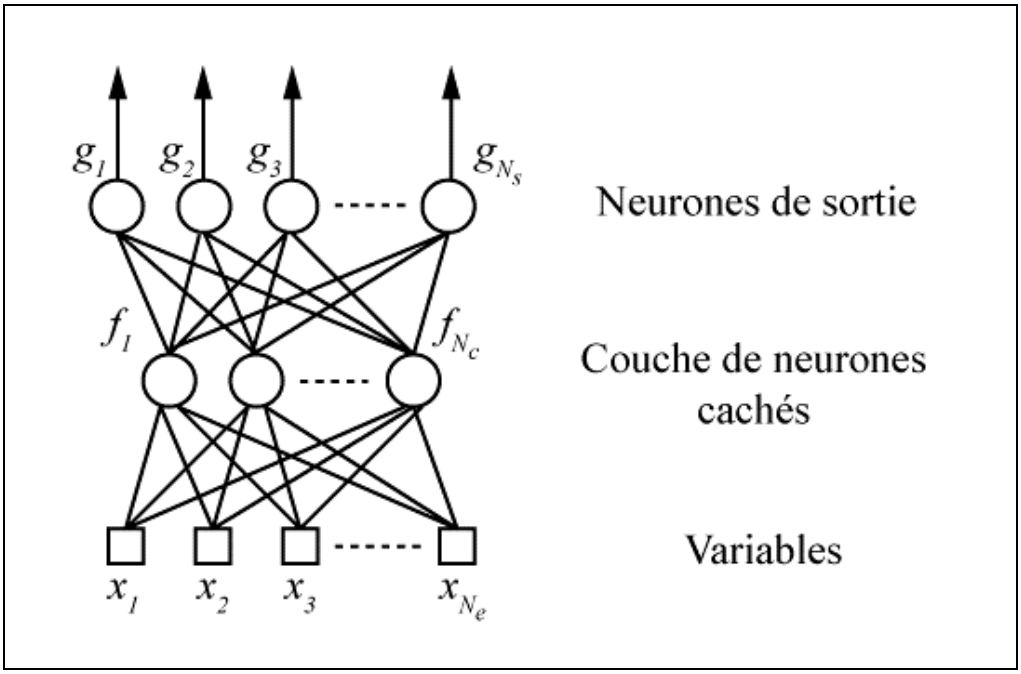


Fig.3.3 : Codeur-décodeur constitué de neurones formels

Cette unité codeur-décodeur peut être utilisée pour effectuer le codage d'un arbre, de différentes manières. Dans l'article [54], le codage s'effectue de la façon suivante : les feuilles de l'arbre, c'est-à-dire les noeuds sans enfant, sont tout d'abord codées, puis leurs parents et les noeuds suivants, de façon récursive, jusqu'au noeud racine. Le vecteur obtenu en sortie du codeur correspondant au noeud racine est alors une représentation compacte de l'ensemble de l'arbre. Le décodage s'effectue de façon symétrique et de manière récursive, fournissant successivement le décodage du noeud racine jusqu'aux feuilles de l'arbre. Il est alors possible de réaliser un apprentissage de l'ensemble codeur-décodeur obtenu, de façon à retrouver en sortie un vecteur égal au vecteur d'entrées. Considérons par exemple l'arbre binaire ((**A**,**B**),(**C**,**D**)) de la Figure 3.4 où **A**, **B**, **C** et **D** sont des vecteurs de taille identique.

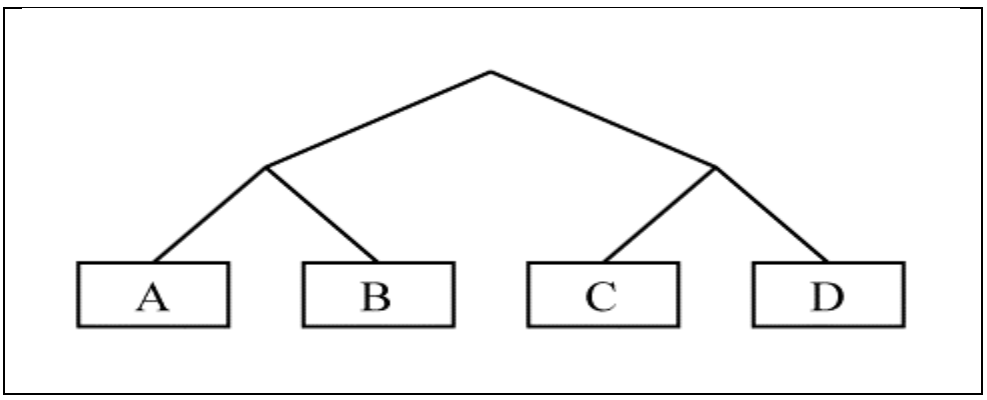


Fig.3.4 : Arbre binaire

A et **B** sont tout d'abord codés en une représentation R_1, **C** et **D** en une représentation R_2, puis R_1 et R_2 sont codés à leur tour en R_3, qui est une forme compacte de l'arbre. Pour retrouver l'information initiale, il faut alors décoder R_3 en (R'_1 , R'_2), à leur tour respectivement décodés en (**A'**,**B'**) et (**C'**,**D'**). L'apprentissage est séquentiel : il se fait tout d'abord sur le codeur – décodeur pour auto-associer (**A**,**B**) à (**A**,**B**), puis sur la paire (**C**,**D**), et enfin sur la paire (R_1,R_2). Ce type de codage pose alors le problème de la « cible mobile» : les représentations des vecteurs non-terminaux (dans le cas de l'exemple évoqué, R_1 et R_2), donc une partie de la base d'apprentissage, changent au cours de l'apprentissage. La convergence de celui-ci n'est alors plus garantie.

Pour nous affranchir de ce problème, nous proposons une méthode de codage légèrement différente. Le principe de base consiste à créer, à partir de plusieurs codeurs-décodeurs tels que ceux précédemment décrits, un modèle dont la structure est isomorphe à celle des arbres à coder, à partir de réseaux de neurones à poids partagés. Ce modèle effectue alors, avec le même jeu de paramètres, le codage des paires (**A**,**B**), (**C**,**D**) et (R_1,R_2).

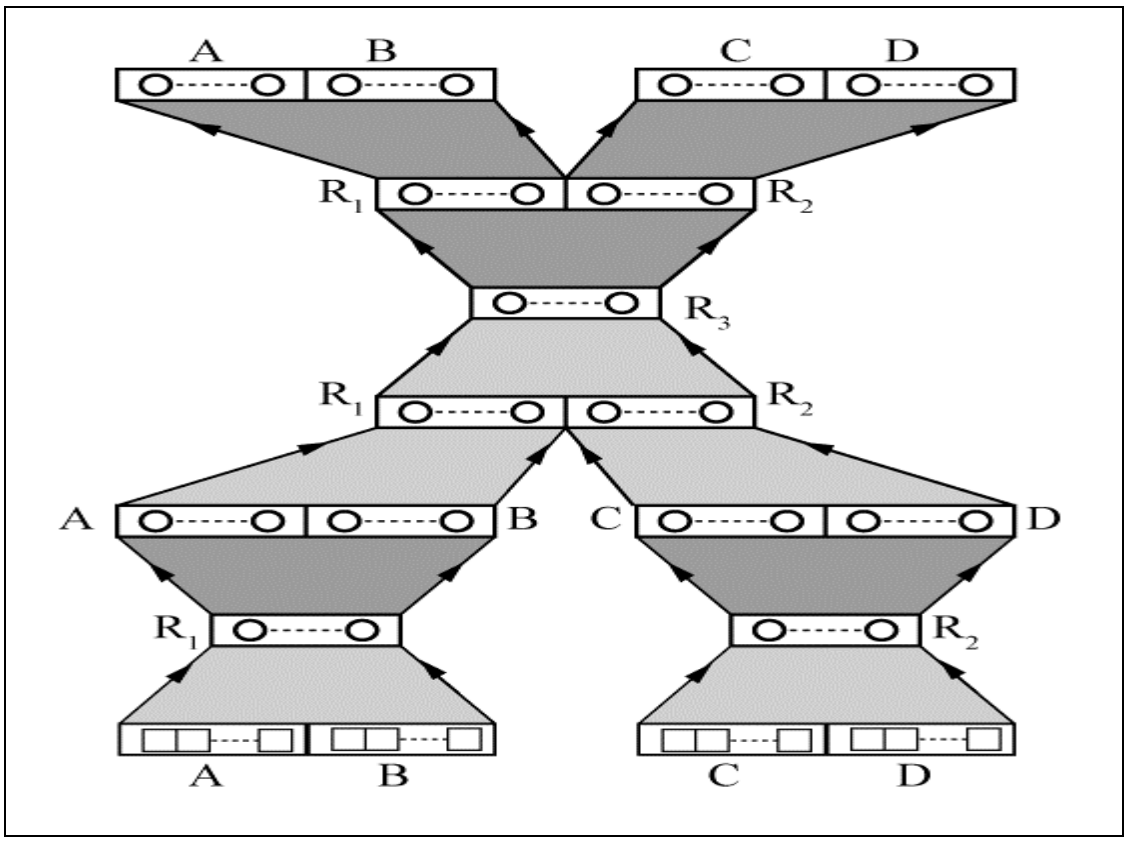


Fig.3.5: Modèle associé au graphe de la Fig.3.4

La Figure 3.5 représente l'ensemble « codeurs-décodeurs » correspondant au modèle associé à l'arbre de la Figure 3.4. Les connexions ne sont pas détaillées, mais représentées par des

zones grisées : les zones grisées de même teinte correspondent à des connexions de même vecteur de paramètres. Sur ce schéma, les carrés n'effectuent aucun calcul. On peut remarquer que la structure de ce réseau est effectivement identique à celle de l'arbre qui est codé.
Les zones grisées peuvent représenter non pas une seule couche de paramètres, mais deux couches de paramètres et une couche de neurones cachés. Supposons que les zones grisées représentent une seule couche de paramètres. Le nombre de neurones cachés doit alors être égal au nombre de neurones nécessaires pour coder une feuille de l'arbre. Or, la complexité de la relation établie entre les entrées et les sorties, qui est liée au nombre de neurones cachés, n'est en général pas liée au nombre de neurones nécessaire au codage des feuilles. Il est possible de se libérer de cette contrainte en intercalant une couche cachée supplémentaire dans le codeur et/ou le décodeur. Dans ce cas, les zones grisées ne représentent plus une seule couche de paramètres, mais deux, ainsi qu'une couche cachée. Les zones grisées de même teinte correspondent alors à des paramètres et à des neurones identiques.

Les RAAMs permettent également de coder des séquences (elles sont alors appelés SRAAMs, pour Sequential RAAMs). Considérons par exemple la séquence (x_1,x_2 ,x_3) représentée sur la Figure 3.6. Le codage est récursif : si l'on note R_x la représentation codée d'un vecteur x, les éléments d'entrée des codeurs successifs sont (0 x_1), (R_{x_1} x_2) et ($R_{x_1x_2}$ x_3).

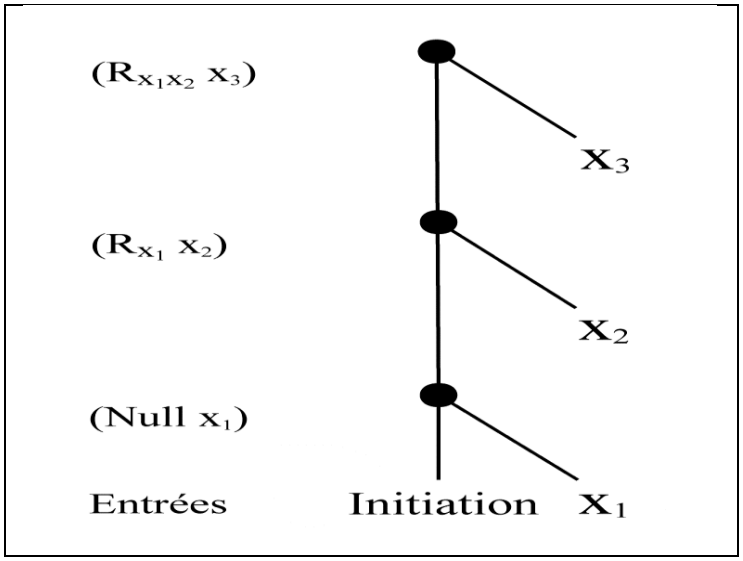


Fig.3.6 : Codage de la séquence (x_1, x_2, x_3) par une SRAAM

La couche d'entrée est ainsi composée de deux sous-éléments, l'un contenant le vecteur de variables correspondant au noeud à coder (c'est-à-dire x_1, x_2, x_3), l'autre au vecteur nul, puis,

au fur et à mesure du codage, à un pointeur vers le noeud enfant, qui est la représentation de la sous-séquence déjà codée (R_{x_1} ou $R_{x_1x_2}$).

3.2.2 Les Mémoires Récursives Auto-Associatives Étiquetées

Les Mémoires Récursives Auto-Associatives Étiquetées (LRAAMs, pour Labeling RAAMs) [54] sont des RAAMs qui permettent de tenir compte des étiquettes d'un graphe. Le principe des LRAAMs consiste à considérer deux types d'entrées pour chaque noeud : une partie des entrées correspond aux pointeurs vers les enfants du noeud dans le graphe, comme précédemment, et une seconde partie correspond aux étiquettes des noeuds.

À chaque noeud du graphe est ainsi associée une entrée, qui comporte un vecteur de variables (qui peuvent être codées de façon binaire) associé à ses étiquettes, et un ensemble de pointeurs vers ses enfants dans le graphe, qui sont des nombres réels, et dont le nombre est égal à son degré sortant. Ces données constituent donc les entrées des codeurs associés à chaque noeud. Les SRAAMs sont en ce sens des LRAAMs, pour lesquelles il n'y a qu'un seul pointeur.

Ceci soulève une nouvelle fois le problème de la cible mobile, car les pointeurs des nœuds non-terminaux sont calculés lors de l'apprentissage, donc varient. Ce problème peut être évité en utilisant la même idée que celle présentée dans le cadre des RAAMs, c'est-à-dire en associant à l'arbre étiqueté un modèle dont la structure est isomorphe à celle de l'arbre. À chaque noeud sont associés un codeur et un décodeur, et les réseaux de neurones constituant les codeurs-décodeurs des différents noeuds sont à poids partagés : ces réseaux partagent le même jeu de paramètres.

3.3 Les graph machines

Nous avons présenté, dans la section précédente, des techniques qui permettent de manipuler des données structurées par apprentissage artificiel.

Nous allons maintenant montrer comment il est possible, sur cette base, d'établir une relation entre des données structurées et des vecteurs de réels, de même que les outils de modélisation traditionnels établissent une relation entre des vecteurs de variables et les vecteurs de données à modéliser.

3.3.1 Modélisation à partir de graphes acycliques

Nous avons rappelé comment les RAAMs et les LRAAMs permettaient de trouver une représentation compacte d'un arbre, sous forme vectorielle, en sortie du codeur. Cette représentation vectorielle de l'arbre peut alors être exploitée, non plus par un décodeur pour retrouver l'information d'origine, mais comme vecteur d'entrée d'un classifieur, par exemple un réseau de neurones [55]. De plus, l'apprentissage de la représentation et celui du classifieur peuvent être réalisés simultanément : l'apprentissage du codeur est effectué dans le contexte de la « cible mobile » tandis que celui du classifieur se fait de manière conventionnelle. Nous pouvons cependant éviter le problème de la cible mobile de la manière décrite dans la section 3.2.1 : à chaque graphe est associé un réseau de structure identique, et les codeurs correspondant à chacun des noeuds sont à poids partagés.

3.3.2 Structure mathématique des graph machines

La première étape, qui constitue l'idée principale sous-jacente aux *graph machines* [56], consiste à construire, pour chacun des graphes, une fonction mathématique réalisée en combinant des fonctions élémentaires, selon une combinaison décrite par le graphe qui lui est associé.

Considérons un ensemble de graphes acycliques $G = \{G_i\}$. Pour chaque graphe G_i, on construit une fonction g^i de la façon suivante : à chaque nœud de G_i est associée la fonction paramétrée dite « fonction de nœud » f_θ, où θ est le vecteur des paramètres, qui est identique pour toutes les fonctions. La fonction associée au nœud racine peut être différente des fonctions relatives aux autres nœuds : nous la noterons F_Θ. Les fonctions f_θ sont alors composées de façon à refléter la structure du graphe : si s_j et s_k sont deux sommets du graphe G_i , tels qu'un arc part de s_j et arrive en s_k, alors le résultat de la fonction associée au nœud s_j est argument de celle associée au nœud s_k.

La fonction de nœud d'un nœud s_k est ainsi de la forme :

$$f_\theta(z_k) = f_\theta(z_0, v_k, x_k). \qquad (23)$$

Les arguments de la fonction sont de plusieurs types :

- z_0 est une constante égale à 1.

- v_k est un vecteur dont les composantes sont égales aux valeurs prises par les fonctions associées aux nœuds enfants du nœud s_k. Puisque la fonction f_θ est la même pour tous les nœuds, ce vecteur doit avoir un nombre fixe de composantes, que nous définissons ainsi :

  $M_i = \text{argmax}_k\, d_k^+$, où d_k^+ est le degré sortant du nœud s_k.

  Pour un nœud s_k tel que $d_k^+ < M_i$, les composantes superflues de v_k sont égales à 0.

- x_k est un vecteur optionnel qui apporte de l'information sur le nœud : ce sont les étiquettes du nœud.

z_k est ainsi un vecteur de taille D_i tel que :

$$D_i = 1 + M_i + |x_k| \qquad (24)$$

Sa première composante z0 vaut 1, les composantes 2 à $d_k^+ + 1$ sont les valeurs prises par les fonctions f_θ des nœuds enfants. Si $d_k^+ < M_i$, c'est-à-dire si le nombre de nœuds enfants du nœud s_k est inférieur au maximum M_i possible, les composantes $d_k^+ + 2$ à $M_i + 1$ sont nulles. Enfin, les composantes $M_i + 1$ à D_i sont celles du vecteur x_k.

La fonction paramétrée, appelée *graph machine*, relative au graphe G_i, est finalement de la forme :

$$g^i_{\theta,\Theta} = F_\Theta\,(z_r\,) \qquad (25)$$

Où $\mathbf{z}_r$ est le vecteur des arguments de la fonction associée au nœud racine.

Lorsque de telles fonctions sont construites pour l'ensemble des graphes $G = \{Gi\}$, les fonctions de nœud f_θ sont identiques pour tous les nœuds d'un même graphe, mais aussi pour tous les graphes. Il est donc nécessaire de s'assurer que les vecteurs $\mathbf{z}$ sont de dimension D fixée quel que soit le graphe considéré, c'est-à-dire $1+M_i + |\mathbf{x}| = D$ pour tout i.

Il faut donc :

- Que le vecteur $\mathbf{x}$ fournisse les mêmes types d'informations pour tous les graphes (et soit ainsi de taille fixe) ;
- Que le vecteur $\mathbf{v}$ soit de dimension M fixe. Il suffit de choisir $M = \max_i M_i$.

3.3.3 Les étiquettes

Le vecteur **x** fournit des informations sur chacun des noeuds du graphe : il correspond aux étiquettes des LRAAMs. Ces informations peuvent être codées de différentes façons. Si la propriété caractérisée par une étiquette est quantitative (si elle mesure par exemple la taille des régions d'une image), il est pertinent d'affecter une seule entrée à cette étiquette, c'est-à-dire une valeur du vecteur **x**, qui varie selon le noeud considéré et la valeur de l'étiquette associée. Au contraire, si cette propriété n'agit pas de façon quantitative sur le problème modélisé, et que l'ensemble des valeurs prises est borné et fini (par exemple les couleurs possibles des régions d'une image), on a affaire à une variable catégorielle : un codage « un parmi *n* » est mieux adapté [56].

3.4 L'apprentissage des graph machines

3.4.1 Propriété d'approximation universelle

Les réseaux de neurones sont des approximateurs universels, ce qui signifie que toute fonction bornée, suffisamment régulière, peut être approchée dans un domaine fini de l'espace de ses variables, avec une précision arbitraire, par un réseau de neurones possédant une couche de neurones cachés et un neurone de sortie linéaire.

Il a été montré que cette propriété peut être étendue aux réseaux de neurones récursifs [57-59]. Elle est également valable dans le cas des *graph machines*. Celles-ci se composent en effet de deux parties : un codeur, qui fournit une représentation compacte d'un graphe sous une forme vectorielle, et une fonction (par exemple un réseau de neurones) permettant d'effectuer une classification ou une régression à partir de cette représentation vectorielle. Cette fonction possède la propriété d'approximation universelle ; la capacité d'approximation des *graph machines* dépend donc de celle du codeur.

Considérons une application F(G) de l'ensemble des arbres *n*-aires, dont les étiquettes sont des réels, vers l'ensemble des réels. Soit $\varepsilon > 0$ une précision arbitrairement choisie et $\delta > 0$ une confiance donnée. Il est alors possible de trouver une machine *M* qui possède la propriété suivante :

$$P\left[G||M(G) - F(G) > \varepsilon|\right] < \delta \qquad (26)$$

Ce résultat, prouvé dans [57-59], signifie qu'il est possible de trouver une *graph machine M* capable d'approcher la fonction *F*, pour tout arbre *n*-aire, avec une précision arbitraire.
Les *graph machines* sont donc, en théorie, bien adaptées pour établir une relation entre des données structurées et des sorties réelles. Nous allons maintenant montrer comment leur apprentissage est réalisé.

3.4.2 Utilisation des algorithmes traditionnels

Nous avons présenté dans le chapitre 2 l'apprentissage traditionnel, pour lequel la base d'exemples permettant d'estimer les paramètres du modèle g_θ est un ensemble de N couples entrées / sortie $(\{\mathbf{x}^i, y^i\}, i = 1,...,N)$. Le modèle est le même pour tous les exemples, et, lors de l'apprentissage, la fonction de coût minimisée est :

$$J(\theta) = \sum_{i=1}^{N} (y^i - g(x^i, \theta))^2 \qquad (27)$$

Lors de l'apprentissage des *graph machines*, la base d'apprentissage est constituée de N couples structure – sortie $(\{G_i, y^i\}, i = 1,\ldots,N)$. Il n'y a plus un modèle unique pour tous les exemples : à chaque exemple correspond une fonction particulière $g^i_{\theta,\Theta}$, composée à partir des fonctions paramétrées F_Θ (pour le nœud racine) et f_θ (pour les autres nœuds), de façon à refléter la structure de l'exemple $_i$. Rappelons que les fonctions F_Θ et f_θ sont les *mêmes* pour tous les exemples.
Il est alors possible de définir une fonction de coût similaire à la fonction de coût des moindres carrés traditionnelle. Cette fonction mesure les écarts entre les observations et les valeurs prédites par le modèle, et peut comporter des termes de régularisation :

$$J(\theta,\Theta) = \sum_{i=1}^{N} (y^i - g^i_{\theta,\Theta})^2 + \lambda_1 \|\Theta\| + \lambda_2 \|\theta\| \qquad (28)$$

où λ_1 et λ_2 sont des constantes de régularisation correctement choisies. La régularisation vise à limiter l'amplitude des paramètres, pour éviter un surajustement du modèle.

La minimisation de la fonction de coût s'effectue de la même manière que lors d'un apprentissage classique, en modifiant les paramètres de façon itérative en fonction de son gradient.

La méthode des poids partagés permet le calcul de celui-ci. Sa *k*-ième composante, c'est-à-dire la dérivée de la fonction de coût par rapport à la composante *k* du vecteur **θ**, est :

$$\frac{\partial J(\theta,\Theta)}{\partial \theta_k} = \sum_{i=1}^{N} \frac{\partial J^i}{\partial \theta_k} \quad (29)$$

où J^i est la contribution de l'exemple i à la fonction de coût.

Le terme $\frac{\partial J^i}{\partial \theta_k}$ peut être calculé grâce à la technique des poids partagés. Le jeu de paramètres **θ** est en effet le même pour tous les nœuds, donc si le graphe G_i comporte n_i nœuds, le nombre d'occurrences du paramètre θ_k dans le graphe G_i est également n_i. Ce terme peut alors s'exprimer comme la somme des contributions de chacun des nœuds :

$$\frac{\partial j^i}{\partial \theta_k} = \sum_{j=1}^{n_i} \frac{\partial J^i}{\partial \theta_k^j} \quad (30)$$

où θ_k^j désigne le paramètre θ_k qui correspond au nœud *j*.

Le gradient s'exprime finalement ainsi :

$$\frac{\partial J(\theta,\Theta)}{\partial \theta_k} = \sum_{i=1}^{N} \sum_{j=1}^{n_i} \frac{\partial J^i}{\theta_k^j} \quad (31)$$

Lorsque les fonctions F_Θ et f_θ sont des réseaux de neurones, ce gradient peut être calculé par rétropropagation, de la manière usuelle. Dans d'autres cas, il est possible d'avoir recours à des méthodes numériques pour ce calcul.

Les algorithmes traditionnels de descente du gradient, tels que Levenberg-Marquardt, BFGS ou le gradient conjugué, peuvent alors être mis en œuvre pour minimiser la fonction de coût.

3.4.3 Sélection de modèle

Les outils que nous avons passés en revue au paragraphe 2.2.2 du chapitre 2 permettent de sélectionner les modèles présentant les meilleures performances de généralisation. Si la validation croisée et le leave-one-out réel peuvent être mis en œuvre de la même manière qu'en modélisation traditionnelle, l'utilisation des leviers et du leave-one-out virtuel n'est pas aussi immédiate : les expressions de la fonction de coût et du gradient ne sont en effet pas les mêmes, et le calcul des leviers n'est pas applicable tel quel. Cette méthode peut cependant être étendue aux *graph machines*.

3.5 Modélisation à partir de graphes cycliques

Nous avons choisi une approche qui consiste à transformer les graphes cycliques en graphes acycliques tout en tenant compte de la présence des cycles du graphe initial. Cette méthode nous permet ainsi de modéliser indifféremment des structures cycliques ou acycliques.

3.5.1 Transformation de graphes quelconques en arborescences

Pour que la structure d'un graphe connexe quelconque puisse être exploitée par la méthode présentée, il est nécessaire de transformer celui-ci en arborescence, donc de rendre ce graphe acyclique et de choisir un nœud racine, ce qui oriente implicitement l'arborescence obtenue. Nous avons décidé d'effectuer cette transformation en appliquant un algorithme dont sa première étape consiste à sélectionner, parmi les nœuds du graphe, celui qui sera le nœud racine de l'arborescence, grâce à un premier algorithme qui numérote les nœuds de façon canonique et unique, selon des critères tels que leurs distances aux autres nœuds du graphe et leur degré. Le nœud choisi comme racine du futur arbre est celui qui porte le numéro 1 ; il est un nœud central du graphe. D'autres critères peuvent être pris en considération, en fonction du type de structures étudiées, et des informations fournies par les étiquettes des nœuds. Par exemple, lorsque les graphes représentent des images, et les nœuds des régions de ces images, il peut être nécessaire d'ajouter un critère pour tenir compte de la taille de ces régions, fournie par les étiquettes, lors de la numérotation canonique des nœuds. Les nœuds des graphes

associés à des molécules chimiques sont numérotés suivant les critères détaillés dans [60]. Dans une seconde étape, les graphes sont transformés en graphes acycliques si nécessaire. Il faut pour cela supprimer une arête pour chaque cycle du graphe, tout en veillant à ce que celui-ci reste connexe. Un second algorithme effectue le choix de ces arêtes à supprimer, en sélectionnant les plus distantes du nœud racine. L'arborescence alors obtenue est orientée du nœud central vers les extrémités du graphe.

Une particularité importante des graph machines est la façon de gérer les cycles : bien que certaines arêtes soient supprimées pour rendre les graphes acycliques, l'information correspondant à leur présence ne l'est pas. Elle est en effet implicitement conservée, par l'intermédiaire des étiquettes, qui fournissent le degré, dans le graphe initial, des deux nœuds reliés par l'arête supprimée. Considérons ainsi les graphes de **la Figure 3.7.**

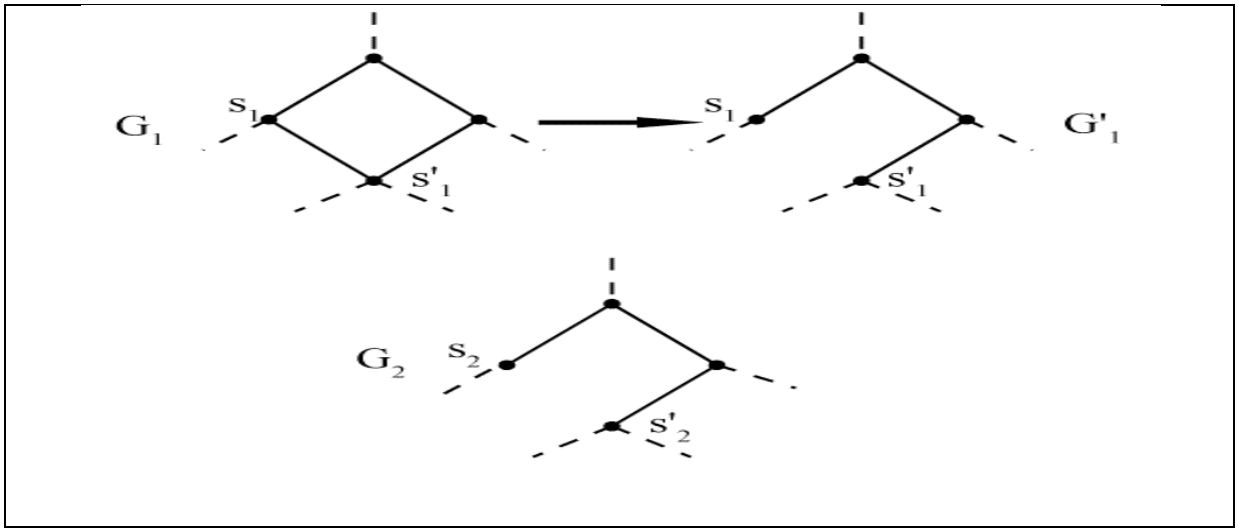


Fig.3.7 : Graphes cycliques et acycliques différenciés par leurs degrés

Le graphe G1 est un graphe cyclique, qui après ouverture devient le graphe G'_1. Le graphe G_2 est acyclique, et semble identique à G'_1 , mais il diffère de celui-ci par ses étiquettes : tandis que les nœuds s_1 et s'_1 sont respectivement de degrés d_1 et d'_1, les nœuds s_2 et s'_2 sont de degrés d_1 -1 et d'_1 -1. Ainsi, un graphe acyclique après suppression d'un ou plusieurs cycles n'est pas identique à un graphe acyclique de même structure, mais n'ayant pas subi de transformation, grâce à l'information contenue dans le degré des nœuds, et portée par les étiquettes.

La Figure 3.8 illustre comment le graphe G_3, qui comporte quatre cycles, est transformé en arborescence. Les numéros affectés aux nœuds lors de la numérotation canonique figurent près des nœuds du graphe G_3. Le nœud 1 est le centre du graphe : c'est le nœud pour lequel la

distance maximale aux autres nœuds est la plus faible, et de plus fort degré. Il est alors choisi comme nœud racine. Le graphe G_3 comporte 4 cycles : il faut donc supprimer 4 arêtes. Le cycle formé par les nœuds 1, 2 et 4 constitue un exemple simple : l'arête 2-4 étant la plus éloignée du nœud racine, c'est celle qui est supprimée. Les autres cycles sont ouverts de la même manière, et l'arborescence obtenue, orientée du nœud racine aux feuilles, est le graphe G'_3 . Les noeuds initialement reliés dans le graphe G_3 conservent leur degré, malgré la suppression des arêtes : ces degrés sont indiqués sur le graphe G'_3 par les chiffres en gras et en italique à gauche des nœuds.

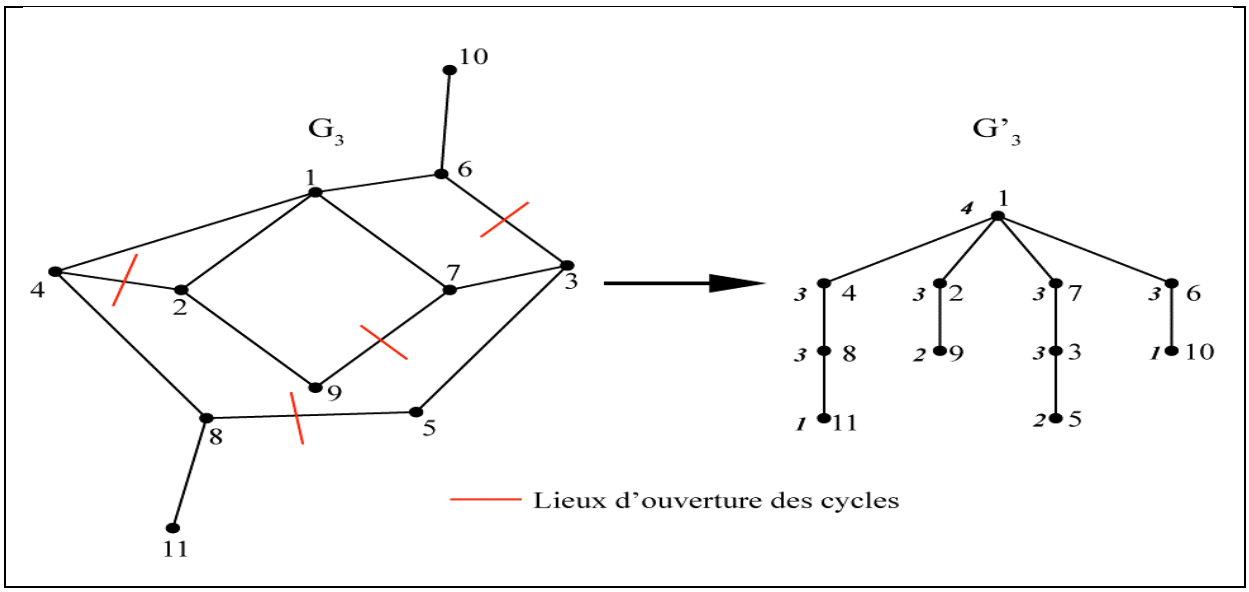


Fig.3.8 : Transformation d'un graphe comportant plusieurs cycles en graphe acyclique

Les graphes cycliques peuvent alors être codés par les *graph machines* de la même manière que les graphes acycliques.

3.5.2 Méthode alternative de modélisation à partir de graphes cycliques

Il est également possible de modéliser les graphes cycliques sans les rendre acyclique. La construction des *graph machines* associées nécessiterait, de la même façon que pour les graphes acycliques, de choisir un nœud racine et d'orienter le graphe. Considérons par exemple les graphes de la Figure 3.9, où G_4 est la structure d'origine et G'_4 le graphe cyclique orienté correspondant, dont le nœud 1 est la racine.

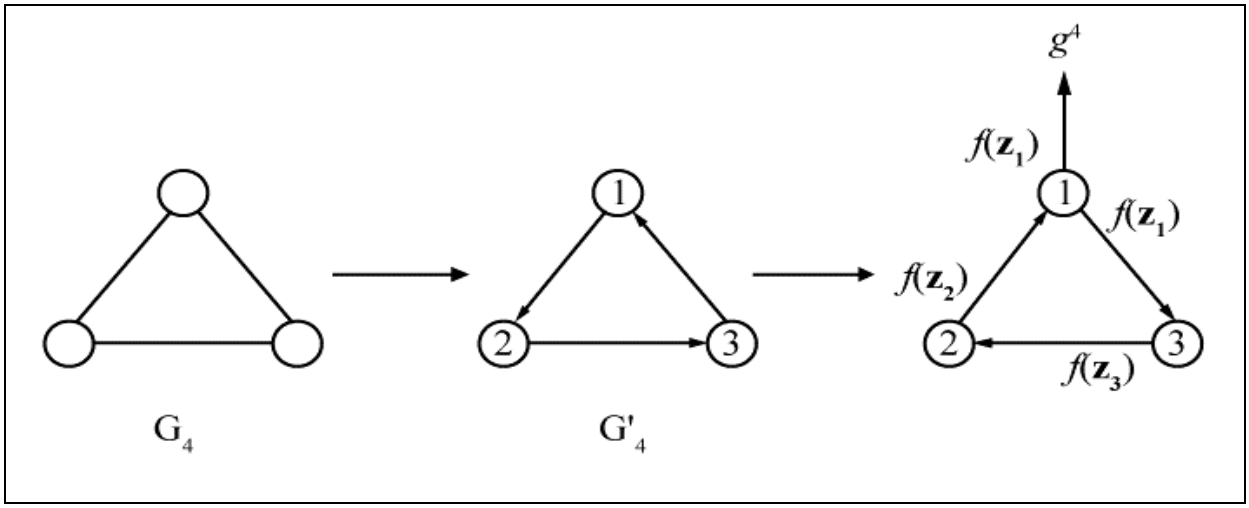


Fig.3.9 : Alternative pour modéliser un graphe cyclique - graphe cyclique simple

La méthode utilisée dans le cas de graphes acycliques, et appliquée au graphe G'_4 , donne alors la relation :

$$g^4 = \underline{f(z_1)} = f(f(z_2), x_1) = f(f(f(z_3), x_2), x_1) = f\left(f\left(f\left(\underline{f(z_1)}, x_3\right), x_2\right), x_1\right) \quad (32)$$

Cette relation ne peut être vérifiée que pour une certaine forme de fonction f donnée, qu'il est possible de déterminer en résolvant l'équation récursive (53).

Un exemple de structure plus complexe est présenté sur la Figure 3.10.

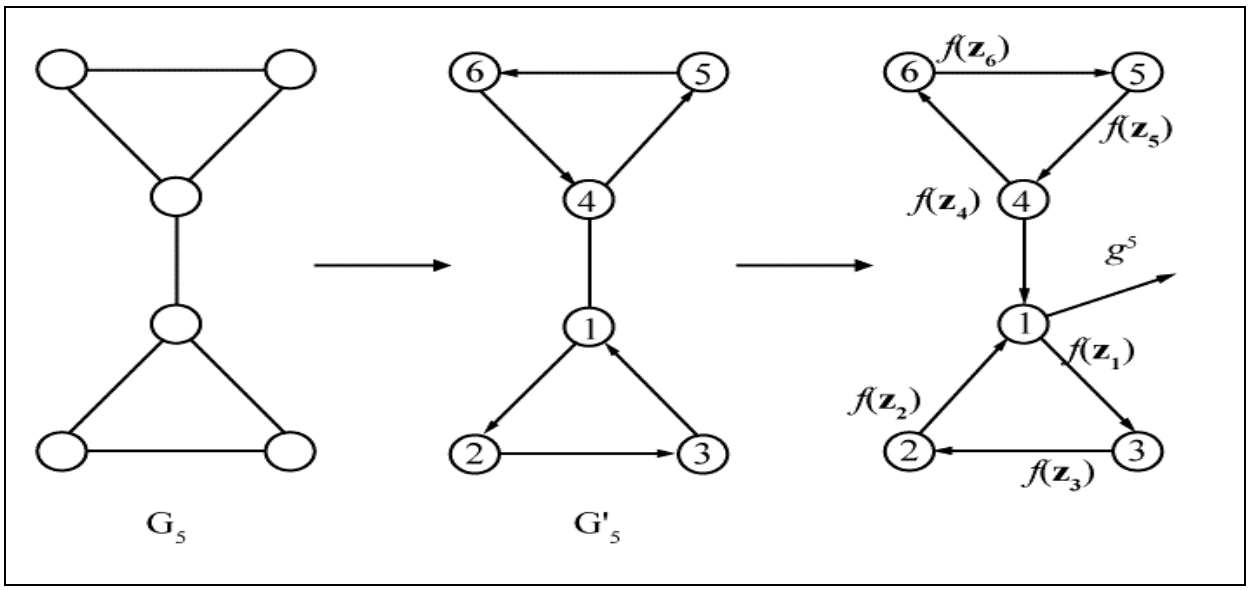


Fig.3.10 : Alternative pour modéliser un graphe comportant deux cycles

Nous voyons que dans ce cas, *f* doit satisfaire deux équations récursives différentes. Il n'est donc plus possible d'utiliser une fonction identique pour tous les nœuds.

L'étude d'une base constituée de *N* structures nécessite ainsi la résolution de *N* équations récursives, et conduit à *N* différentes fonctions de nœud racine F^i , $i \in [1, \mathrm{N}]$. Les inconvénients de cette méthode sont donc multiples :

- La résolution des équations récursives, qui peuvent devenir très complexes pour de grandes structures, augmente le temps de calcul.
- Nous ne pouvons plus utiliser les poids partagés et ne disposons pas d'un outil de calcul nous permettant, par apprentissage, de déterminer les paramètres de différentes fonctions *f*.
- La fonction de nœud n'étant plus unique, nous ne pouvons plus utiliser les leviers ni tous les outils qui s'y rapportent.

Cette approche semble donc moins efficace que celle retenue pour la modélisation des graphes cycliques.

3.6 Exemple de prédiction d'une propriété moléculaire par les graph machines (coefficient de partage eau/octanol)

Le transport, le passage à travers les membranes, la bioaccumulation ou encore l'activité pharmacologique d'une molécule peuvent être conditionnés par son partage entre une phase lipidique et une phase aqueuse, c'est-à-dire son caractère hydrophile. Celui-ci peut être quantifié par le coefficient de partage eau-octanol, noté logP, qui mesure la solubilité différentielle d'un soluté dans ces deux solvants non miscibles :

$$LogP = log\left(\frac{C_{octanol}}{C_{H_2O}}\right) \quad (33)$$

$C_{octanol}$ et C_{H2O} sont les concentrations du soluté dans l'octanol et l'eau. Le logP est ainsi utilisé dans de nombreux modèles en tant que descripteur, pour la prédiction d'effets toxiques ou biologiques ou d'interactions ligand-récepteur. Cette propriété physico-chimique peut être mesurée, mais ces mesures sont généralement longues et coûteuses. Par conséquent,

différentes méthodes de prédiction du logP ont été mises au point, et il existe un nombre important de logiciels de prédiction de cette propriété. Ceux-ci s'appuient aussi bien sur des méthodes de contribution de groupes (ACD/LogP [61], KowWin [62, 63], cLogP [64]…) que sur des régressions multilinéaires à partir de descripteurs (VLogP [65]) ou sur des réseaux de neurones (AUTOLogP [66]).

Il a été réalisé un modèle prédictif du coefficient de partage eau-octanol, par apprentissage, à l'aide d'une base de 1050 composés appartenant à diverses familles de molécules. L'erreur expérimentale sur les valeurs du logP est estimée entre 0,2 et 0,3. La modélisation et la sélection du modèle sont réalisées sur un ensemble d'apprentissage de 875 exemples, choisis de façon aléatoire. Le coût d'apprentissage et le score de leave-one-out virtuel, donnés dans le Tableau 3, sont quasiment identiques pour le modèle choisi : celui-ci ne devrait donc pas être surajusté aux données d'apprentissage.

La qualité du modèle obtenu est alors évaluée sur une base de test de 175 exemples. Les résultats d'apprentissage (EQMA) et de test (EQMT) sont comparés dans le Tableau 3 aux résultats obtenus par une méthode de régression par des réseaux de neurones, à partir de descripteurs [67]. Nous y indiquons également le nombre d'exemples dans chacune des bases d'apprentissage (N_{app}) et de test (N_{test}) qui sont également indiqué dans le tableau.

Modèle	**GM**	**Réseaux de neurones [79]**
N_{app} / N_{test}	875 / 175	980 / 105
EQMA	0,29	0,41
LOO virtuel	0,30	-
EQMT	**0,30**	0,53

Tableau 3 : Comparaison des performances de modélisation du logP des graph machines et de réseaux de neurones

Les prédictions sur la base de test sont également représentées sur la Figure 3.11.

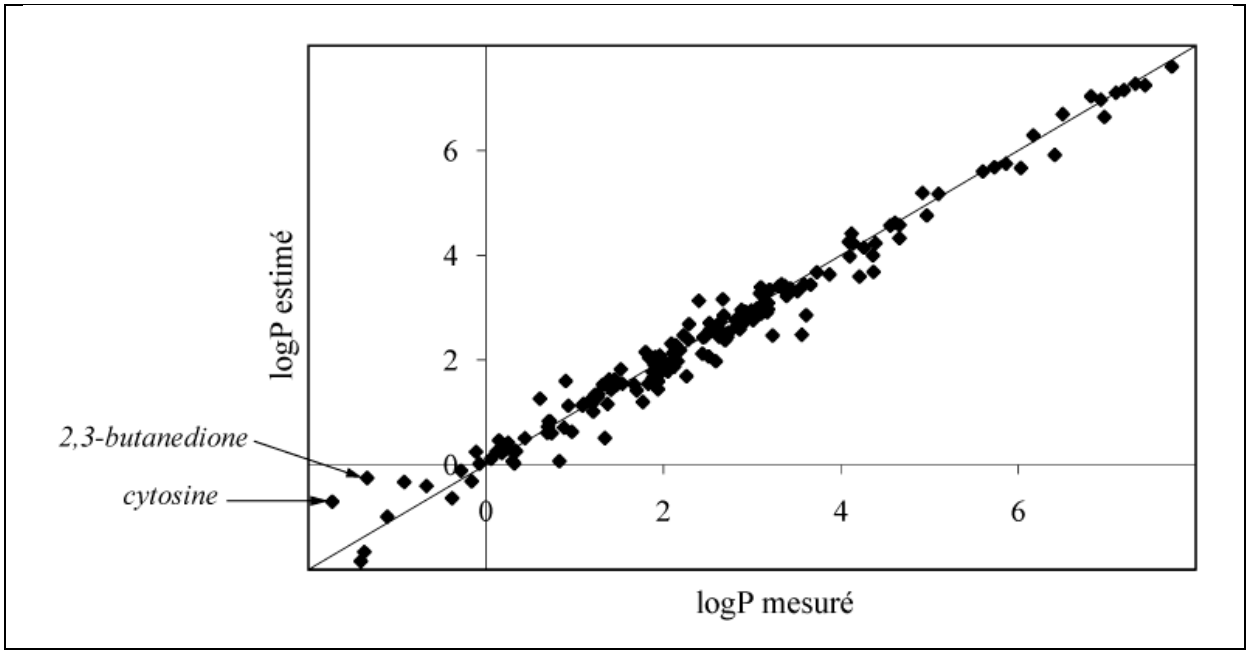


Fig.3.11 : Prédiction du coefficient de partage eau-octanol sur la base de test

Le coefficient de partage eau-octanol de ces molécules est bien prédit : les erreurs commises par le modèle sur les bases d'apprentissage et de test sont proches de l'erreur expérimentale. De plus, l'erreur de prédiction sur la base de test est quasiment identique au coût d'apprentissage, ce qui suggère à nouveau que le modèle n'est pas surajusté. Les valeurs du Log les moins bien prédites sont deux valeurs très faibles, qui se situent à la périphérie du domaine d'apprentissage [57].

3.7 Conclusion

Dans de nombreux problèmes de modélisation, les données se présentent sous la forme de structures, il est plus pertinent de tirer directement profit de la structure des données, par l'intermédiaire d'un modèle capable d'établir de façon directe une association entre ces structures et un vecteur de sorties, sachant que les données structurées se représentent aisément sous la forme de graphes.

Dans ce chapitre nous avons commencé par la présentation des graphes en tant qu'objets mathématiques. Nous avons introduit ensuite les mémoires récursives auto-associatives, premiers modèles à réaliser un codage de données structurées par apprentissage artificiel, ainsi que les *graph machines* qui permettent d'établir une relation entre des données structurées et des vecteurs de réels. Enfin un exemple de modélisations de propriétés physico-chimiques de molécule a été présenté.

La modélisation par apprentissage statistique consiste à construire, à partir d'un échantillon d'individus, des modelés mathématiques qui reproduisent le comportement d'un système, afin de pouvoir prédire-pour un ensemble plus grand d'individus-une ou plusieurs réponses du système à partir de ses variables d'entrée [**66**]. Dans de nombreux domaines, comme les sciences sociales, la chimie moléculaire ou le traitement de données textuelles, il arrive que les entrées du système se présentent sous forme de structures (réseaux sociaux, arrangements d'atomes, constructions grammaticales des phrases,…). Il serait alors avantageux d'utiliser ces structures pour la modélisation des réponses étudiées. Ceci est souvent le cas dans le domaine de la chimie où, dans de nombreuses applications, les entités en entrée d'un procédé peuvent être des molécules dont on cherche à prédire les propriétés physico-chimiques (réponses du procédé), pour des réactions particulières, à l'aide de modèles construits à partir de données expérimentales. Il existe un certain nombre de méthodes, dans le domaine de la chimiométrie, qui s'appuient sur le principe que les propriétés physico-chimiques des molécules dépendent fortement de leur structure. Regroupées sous l'acronyme *QSAR* (pour *Quantitative Structure-Activity Relationship*), ce sont principalement des méthodes de régression linéaire ou non linéaires qui ont pour objectif de modéliser les propriétés (ou activités) physico-chimique à partir de caractéristiques décrivant la structure des molécules (vue dans le chapitre 2). Ces caractéristiques, appelées descripteurs moléculaires, sont générées par des techniques de modélisation moléculaire. On pourrait reprocher aux modèles obtenus par ces méthodes de ne pas être directement construit à partir de la structure des molécules, mais de s'appuyer sur des nouvelles variables, que sont les descripteurs moléculaires, qui sont en fait des représentations vectorielles de cette structure.

Dans ce chapitre nous proposons une méthodologie de modélisation à l'aide des *graph machines* basée sur un codage qui tient compte directement de la structure des molécules que nous désignons par QSAR-GM (GM pour graph machines). Dans ce codage, chaque molécule est représentée par un graphe acyclique orienté dont les nœuds sont associes aux atomes et les arêtes aux liaisons (détaillé dans le chapitre 3).

Pour distinguer QSAR-GM de la méthode QSAR classique, nous désignons cette dernière par *QSAR-DM* (DM pour descripteurs moléculaires).

4.1 Méthodologie de la modélisation à base des graph machines

La prédiction de propriétés et d'activités physico-chimiques de molécules présente un enjeu industriel important, car elle permet de réduire les délais et les coûts de développement. Deux disciplines de la chimiométrie se sont développées en réponse à ce besoin : la modélisation des relations structures-activité désignées par QSAR (pour Quantitative Structure-Activity Relationships), et la modélisation des relations structure-propriété désignées par QSPR (pour Quantitative Structure Property Relationships). Elles consistent essentiellement en la recherche de similitudes entre molécules dans de grandes bases de données de molécules existantes dont les propriétés sont connues. La découverte de telles relations permettent de prédire les propriétés physiques et chimiques et l'activité biologique de composés, de développer de nouvelles théories ou d'expliquer les phénomènes observés. Elle permet également de guider la synthèse de nouvelles molécules, sans avoir à les réaliser, ou à analyser des familles entières de composés.

Nous proposons, de façon distincte mais complémentaire à l'approche QSAR-DM, une méthode que nous désignons par QSAR-GM qui permet de modéliser la propriété étudier directement à partir de la structure des molécules codées par des graphes qui s'appelle *graph machines* (détaillé dans le chapitre 3). Les molécules sont représentées par des graphes acycliques qui tiennent compte des liaisons chimiques, de la nature des atomes ou encore de la stéréochimie du composé initial : à chaque atome non-hydrogènes est associé un nœud, et à chaque liaison entre deux atomes une arête entre les deux nœuds correspondants. Les nœuds peuvent de plus être caractérisés par des étiquettes, qui fournissent des informations sur la nature, le degré ou l'isomérie de l'atome en question. Il est également possible d'utiliser des descripteurs au sein même des *graph machines* par l'intermédiaire des étiquettes. Enfin, le graphe est orienté, par le choix d'un nœud central. Un exemple de représentation de molécule par un graphe est donné à la figure 4.1. Le nœud central est l'atome de carbone de degré 3.

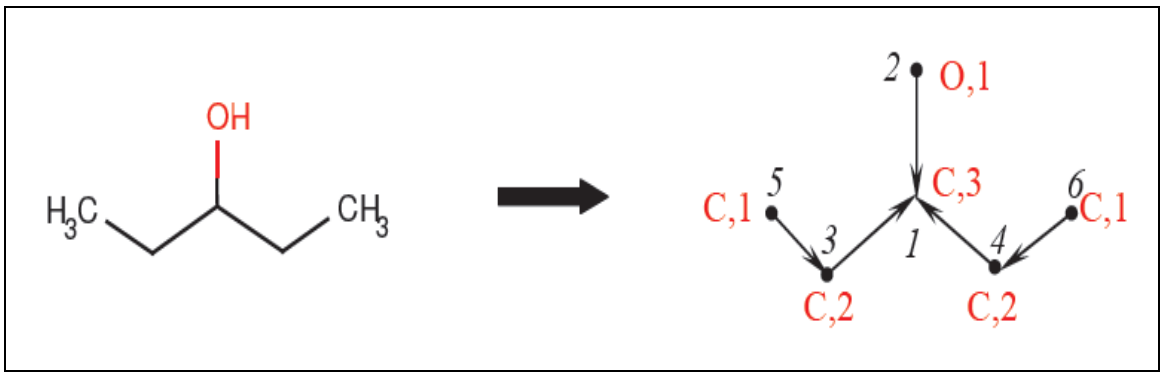


Fig.4.1 : Représentation d'une molécule par un graphe étiqueté. Les étiquettes du graphe (police rouge), indiquent la nature de l'atome (C ou O) ainsi que son degré (1,2 ou 3) i.e. le nombre de liaisons avec les atomes voisins. Les numéros en gras italique sont les indices de chacun des nœuds.

La méthode QSAR-GM consiste alors à faire correspondre à chaque graphe de la base de données une fonction de même structure mathématique que le graphe associé, de la façon suivante :

-A chaque nœud du graphe est associée une fonction paramétrée f_θ , appelée pour cette raison *fonction de nœud*, où θ est le vecteur des paramètres, identique pour tous les nœuds. Les fonctions paramétrées sont, par exemple, des réseaux de neurones.

- Pour chaque graphe G_i , on construit une fonction g_θ^i par composition des fonctions f_θ , de façon à refléter la structure du graphe : si s_a et s_b sont deux sommets du graphes, tels que a est parent de b (i .e. un arc part de s_a et arrive ens_b), alors le résultat de la fonction associée au nœud s_b est argument de celle associée au nœud s_a .

La fonction de nœud paramétrée f_θ associée au nœud z est donc de la forme :

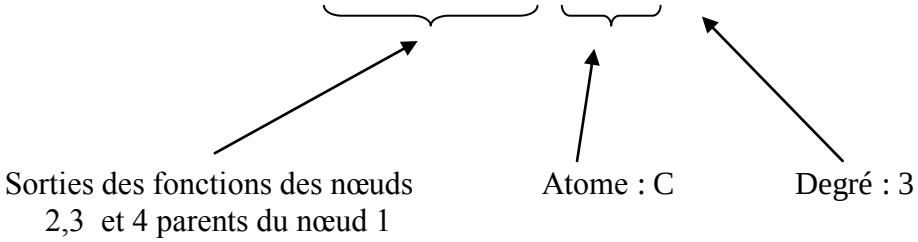
$$f_{\boldsymbol{\theta}}(z) = f(\boldsymbol{u}, \boldsymbol{v}) \qquad (34)$$

Où :

- $\boldsymbol{u}$ est un vecteur dont les composantes sont égales aux arguments de sorties des fonctions associées aux nœuds parents du nœud z en question.
- $\boldsymbol{v}$ est un vecteur optionnel dont les composantes fournissent l'information localisée au nœud : ce sont les étiquettes du nœud pouvant être une valeur qualitative (comme la nature du nœud, exemple le type d'atome associé au nœud, codée en disjonctif complet) ou quantitative (comme le nombre total d'arêtes qui sont reliées au nœud).

Ainsi, la fonction *graph machines* associée à la molécule représentée sur la figure 4.1 est :

$$g_\theta = f_\theta(f(z_2), f(z_3), f(z_4), (0,1,0,0), 3) \quad (35)$$



A *N* molécules correspondent ainsi *N* fonction composées, appelées *graph machines*, partageant le même jeu de paramètres. La modélisation d'une propriété consiste à estimer ces paramètres par apprentissage statistiques (section 3.4 du chapitre 3). Cet apprentissage diffère de l'apprentissage traditionnel, pour lequel le modèle est unique, et la base d'apprentissage constituée de *N* couples entrées /sorties. Lors de l'apprentissage des *graph machines*, la base d'apprentissage est constituée de *N* couples structures/sorties, et le modèle n'est plus unique. Cependant, puisque ces modèles partagent le même jeu de paramètres, il est possible d'utiliser les techniques traditionnelles d'apprentissage pour estimer ces paramètres.

La modélisation par apprentissage statistique consiste à estimer les paramètres qui conduisent à la meilleure approximation de la fonction de régression, à partir des couples entrées/sortie constituant l'*ensemble d'apprentissage*. Dans le cadre des méthodes classiques d'apprentissage, les paramètres d'un modèle g_θ sont estimés à l'aide d'un ensemble de *N* couples $\{(x^i,y^i), i=1,\ldots,N\}$ où les vecteurs $\boldsymbol{x^i}$ sont les entrées du modèle, et y^i les valeurs mesurées de la réponse à modéliser. Le modèle est le même pour toutes les observations, et la fonction de coût minimisée peut se mettre sous la forme :

$$J(\theta) = \sum_{i=1}^{N} (y^i - g(x^i, \theta))^2 \quad (36)$$

Lors de l'apprentissage des *graph machines*, l'ensemble d'apprentissage est constitué de *N* couples structures/sorties $(\{G_i, y^i\}, i=1,\ldots,N)$, où G_i est la fonction mathématique paramétrée associée au graphe *i*, et y^i la valeur de la réponse modélisée pour ce même graphe. Il n'y a plus un modèle unique pour toutes les observations : à chaque exemple *i* correspond une fonction particulière g_θ^i , composée de la fonction paramétrée f_θ , associée la structure de l'individu *i*. Une fonction de coût similaire à la fonction de coût des moindres

carrés traditionnelle peut être définie. Cette fonction mesure les écarts entre les observations et les valeurs prédites par le modèle :

$$J(\theta) = \sum_{i=1}^{N} (y^i - g_\theta^i)^2 \quad (37)$$

La minimisation de cette fonction de coût s'effectue de la même manière que lors d'un apprentissage classique, en modifiant les paramètres de façon itérative en fonction de son gradient. Lorsque la fonction f_θ est un réseau de neurones, ce gradient peut être calculé par rétropropagation, de la manière usuelle.

Les techniques habituelles de sélection de modèle, par validation croisée par exemple, peuvent également être appliquées aux *graph machines* (section 2.2.2.2.1 du chapitre 2). En effet, la modélisation vise à fournir un modèle qui soit non seulement ajusté aux données d'apprentissage, mais aussi capable de prédire la valeur de la sortie sue des molécules n'appartenant pas à l'ensemble d'apprentissage, c'est-à-dire généraliser.

4.2 Conclusion

Dans ce chapitre, nous avons mis l'accent principalement sur deux problèmes

- La perte d'information au niveau de la représentation de la structure moléculaire par des descripteurs
- L'unicité de modèle généré pour toutes les observations lors de l'apprentissage classique basé sur *N* couples entrées /sorties.

Face à ces problèmes nous avons proposé une approche de construction d'un modèle virtuel pour la prédiction des propriétés chimiques à l'aide de la méthode *graph machines* qui s'affranchie de ces problèmes, d'une part par la représentation de la structure moléculaires par des graphes acycliques orientés dont les nœuds sont associes aux atomes et les arêtes aux liaisons et d'autre part par la génération de plusieurs modèles (un modèle pour chaque observation) lors de son apprentissage basé sur *N* couples structure /sorties .

Conclusion générale

Dans ce mémoire, nous avons procédé à une analyse du domaine de la chemoinformatique dans le contexte de sa modélisation. Cette analyse a fait ressortir quelques limites de cette modélisation, en particulier, la perte d'information et l'unicité du modèle obtenu lors de la phase d'apprentissage.

Nous avons proposé une méthodologie de modélisation à base du *graph machines* qui permet de faire la régression et la classification à partir de données structurées, ainsi que les applications de cette méthode à la prédiction de propriétés et d'activités moléculaires.

Les techniques traditionnelles de modélisation établissent une relation entre la grandeur modélisée et un vecteur de variables qui la détermine. Les principaux inconvénients de ces méthodes résident dans la difficulté du choix des variables pertinentes, et dans leur calcul ou leur mesure préalable. Nous avons proposé une approche basée sur les *graph machines* qui s'affranchit de ces problèmes lorsque les données sont structurées, car elle établit une relation directe entre la structure de ces données et la grandeur modélisée. L'apprentissage s'effectue donc non plus à partir de vecteurs de données, mais à partir de graphes. Notre approche consiste à faire correspondre à chaque graphe de la base de données d'apprentissage une fonction de même structure mathématique que le graphe associé. Cette fonction est la combinaison de fonctions paramétrées identiques, obtenues en associant à chaque nœud du graphe une fonction paramétrée. La modélisation d'une propriété consiste ensuite à déterminer ces paramètres par apprentissage, à partir de la base d'apprentissage constituée de couples structures/sorties.

Les fonctions partagent toutes le même jeu de paramètres, ce qui permet d'utiliser les techniques traditionnelles d'apprentissage. Nous avons montré que les techniques habituelles de sélection de modèle peuvent également être appliquées, notamment le calcul des leviers des exemples, qui sont une mesure de l'influence de chacun d'eux sur le modèle. Ce calcul permet de détecter les catégories de molécules sous-représentées dans la base d'apprentissage, et éventuellement de modifier celles-ci pour améliorer le modèle. Les leviers permettent également d'évaluer les capacités de généralisation des modèles obtenus par apprentissage.

Ainsi, cette nouvelle approche présente l'avantage d'éviter le calcul des descripteurs moléculaires habituellement utilisés pour ce type de problème, puisqu'il suffit de connaître la structure en

graphe de la molécule. De plus, contrairement aux méthodes traditionnelles de modélisation, les *graph machines* ne sont pas spécifiques à une propriété ou une activité donnée. Un aspect remarquable des *graph machines* est que la même machine peut permettre de prédire différentes propriétés, au prix seulement d'un apprentissage, et sans qu'il soit nécessaire de la reconstruire.

En terme de perspective nous envisageons de développer une approche basée sur les *graph machines* afin de prédire l'odeur balsamique des molécules odorantes. Notre approche sert a bien représenter les données à partir de leurs structures et de montrer sa capacité à prédire cette propriété chimique.

La méthodologie est utile aussi pour toutes les applications de conception d'un nouveau médicament en se basant sur le principe de représentation des molécules directement à partir de leurs structures, et ce, à l'aide des graphes acycliques orientés. De nombreux développements peuvent donc être envisagés pour cette nouvelle approche de l'apprentissage de données structurées, tant du point de vue méthodologique que du point de vue de la diversité des applications potentielles.

Bibliographie

[1]. Paris G, Meeting of the American Chemical Society, quoted by W. Warr at http://www. warr .com/warrzone.htm , August 1999.

[2]. FH Allen, Acta Crystallogr B58:380–388 http://www.ccdc.cam.ac.uk/ ,2002.

[3]. fr.wikipedia.org/wiki/Chemoinformatique.

[4]. http://www.greyc.ensicaen.fr/~mbrun/1A_MCF_PROJETS/HEMELAERE_CASTRO/con t eneur.php? page=pagea&ban=1.

[5]. http://www.greyc.ensicaen.fr/~mbrun/1A_MCF_PROJETS/HEMELAERE_CASTRO/con t eneur.php? page=pageb&ban=1.

[6]. http://fr.wikipedia.org/wiki/Chimie.

[7]. http://fr.wikipedia.org/wiki/Chimie#Disciplines_de_la_chimie.

[8]. http://fr.wikipedia.org/wiki/Propri%C3%A9t%C3%A9s_physico-chimiques_des_prot%C3 %A9ines.

[9]. http://fr.wikipedia.org/wiki/Relation_quantitative_structure_%C3%A0_activit%C3%A9.

[10]. J Gasteiger (ed), Handbook of chemoinformatics-from data to knowledge, Wiley-VCH, Weinheim, 2003.

[11]. J Gasteiger, T Engel (Eds) Chemoinformatics-a textbook. Wiley-VCH, Weinheim , 2003.

[12]. http://www.greyc.ensicaen.fr/~mbrun/1A_MCF_PROJETS/HEMELAERE_CASTRO /conteneur.php?page=pagee&ban=1.

[13]. http://www.greyc.ensicaen.fr/~mbrun/1A_MCF_PROJETS/HEMELAERE_CASTRO /conteneur.php?page=paged&ban=1.

[14]. http://www.greyc.ensicaen.fr/~mbrun/1A_MCF_PROJETS/HEMELAERE_CASTRO /conteneur.php?page=page3&ban=2.

[15]. Johann Gasteiger, Chemoinformatics: a new field with a long tradition, *Springer*-Verlag 2005.

[16]. FANIA BAJOT, The use of QSAR and computationnel methods in drug design, 2010.

[17]. Axel J. Soto[1,2] , Ignacio Ponzoni[1,2], and Gustavo E. Vazquez[1] ,Segregating Confident Predictions of Chemicals'Properties for Virtual Screening of Drugs, 2009.

[18]. [1]A. M. SMALTER AND [1]J. HUAN AND [2]G. H. LUSHINGTON, Chemical compound classification with automatically mined structure patterns , 2007.

[19]. Pierre Geurts[1,2] , Louis Wehenkel[2], Florence d'Alché-Buc[1] , OK3: Méthode d'arbres à sortie noyau pour la prédiction de sorties structurées et l'apprentissage de noyau, 2006.

[20]. Mourad KORICHI[1,2] , Vincent GERBAUD[1], Xavier JOULIA[1], Pascal FLOQUET[1] , Approche multi-classe de représentation des molécules pour la conception des produits-procédés assistée par ordinateur, 2006.

[21]. Mourad KORICHI[1,2] , Vincent GERBAUD[1], Thierry TALOU[3], Christine RAYNAUD[3], Pascal FLOQUET[1] , Relation Structure moléculaire – Odeur : Utilisation des Réseaux de Neurones pour l'estimation de l'Odeur Balsamique, 2006.

[22]. Subhransu Maji, Shashank Mehta, A netflow distance between labeled graphs: applications in chemoinformatics, 2005.

[23]. Frédéric Pennerath[*,**], Amedeo Napoli[**] , La fouille de graphes dans les bases de données réactionnelles au service de la synthèse en chimie organique, 2004.

[24]. Samia Aci[1], Gilles Bisson[2], Sylvaine Roy[3], and Samuel Wieczorek[3] ,Clustering of Molecules: Influence of the Similarity Measures, 2006.

[25]. PATRA VOLARATH, Application of term-rewriting grammar in chemical reaction prediction, 2008.

[26]. Crum-Brown, A., et Frazer, T., on the connection between chemical constitution and physiological action. *Transactions of the Royal Society of Edinburgh* 1868-69, 25, p. 151-203.

[27]. Hansch, C., Leo, A., et Hoekmann, D. Exploring QSAR: hydrophobic, electronic and steric constants. Washington, DC: American Chemical Society, 1995.
[28]. Wiener, H., Structural determination of paraffin boiling points. *Journal of Chemical Information and Computer Sciences*, 1947, 69, p. 17-20.
[29]. Randié, M., on characterization of molecular branching. *Journal of the American Chemical Society*, 1975, 97, p. 6609-6614.
[30]. Kier, L.B., et Hall, L.H, Molecular connectivity in chemistry and drug research. New-York : Academic Press, 1976.
[31]. Balaban, A.T, Highly discriminating distance-based topological index. *Chemical Physics Letters*, 1982, 89, p. 399-404.
[32]. Heritage, T.W., *et al*, EVA: A novel theoretical descriptor for QSAR studies, *Perspectives in Drug Discovery and Design*, 1998, 9-11 (0), p. 381-398.
[33]. Schuur, J.H., Selzer, P., et Gasteiger, J , The coding of the three-dimensional structure of molecules by molecular transforms and its application to structure-spectra correlations and studies of biological activity. *Journal of Chemical Information and Computer Sciences*, 1996, 36 (2), p. 334-344.
[34]. Jolliffe, I.T., Principal Component Analysis. New-York, NY: Springer, 2ème édition, 2002.
[35]. Martens, H., et Næs, T., Multivariate calibration. Chichester: Wiley, 1989.
[36]. Wold, H., Estimation of principal components and related models by iterative least squares, in *Multivariate Analysis*, Krishnaiaah, P.R., Editor. 1966, New York : Academic Press. p. 391-420.
[37]. Höskuldson, A., PLS regression methods. *Journal of Chemometrics*, 1988, 2, p. 211-228.
[38]. Dreyfus, G., *et al.* , Réseaux de neurones, méthodologie et applications. Paris: Eyrolles, 2ème édition, 2004.
[39]. McCulloch, W.S., et Pitts, W., A logical calculus of ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 1943, 5, p. 115-133.
[40]. German, S., Bienenstock, E., et Doursat, R., Neural networks and the bias/variance dilemma. *Neural Computation*, 1992, 4 (1), p. 1-58.
[41]. Bengio, Y., et Grandvalet, Y., No unbiased estimator of the variance of K-fold crossvalidation. *Journal of Machine Learning Research*, 2003, 5, p. 1089-1105.
[42]. Vapnik, V.N., The nature of statistical learning theory. Springer ed, 1995.
[43]. Monari, G., Sélection de modèles non linéaires par leave-one-out : étude théorique et application des réseaux de neurones au procédé de soudage par points thèse en ligne]. Paris : Université Pierre et Marie Curie (Paris 6), 1999. Disponible sur : http://www.neurones.espci.fr/Theses_PS/MONARI_G/THESE.pdf.
[44]. Monari, G., et Dreyfus, G., Local overfitting control via leverages. *Neural Computation*, 2002, 14, p. 1481-1506.
[45]. Leo, A., *et al.*, Calculation of hydrophobic constant (logP) from π and f constants. *Journal of Medicinal Chemistry*, 1975, 18, p. 865.
[46]. Klopman, G., *et al.*, Computer automated logP calculations based on an extended group approach. *Journal of Chemical Information and Computer Sciences*, 1994, 34 (4), p. 752-781.
[47]. Jalowka, J.W., et Daubert, T.E., Group contribution method to predict critical temperature and pressure of hydrocarbons. *Industrial and Engineering Chemistry Process Design and Development*, 1986, 25 (1), p. 139-142.
[48]. Daubert, T.E., et Bartakovits, R., Prediction of critical temperature and pressure of organic compounds by group contribution. *Industrial & Engineering Chemistry Research*, 1989, 28 (5), p. 638-641.
[49]. Cramer, R.D., Patterson, D.E., et Bunce, J.D., Comparative Molecular Field Analysis (CoMFA). 1. Effect of shape on binding of steroids to carrier proteins. *Journal of the American Chemical Society*, 1988, 110 (18), p. 5959.

[50]. Sadowski, J., et Gasteiger, J., From atoms and bonds to three-dimensional atomic coordinates: automatic model builders. *Chemical Reviews*, 1993, 93, p. 2567-2581.
[51]. Kohonen, T. , Self-organization and associative memory. *Springer Series in Information Sciences*. Vol. 8. Berlin: Springer Verlag, 1984.
[52]. Berge, C. Graphes. Gauthier-Villars ed. Paris : Bordas, 3ème édition, 1983.
[53]. Pollack, J. , Recursive distributed representations. *Artificial Intelligence*, 1990, 46, p. 77-106.
[54]. Sperduti, A., Labeling RAAM. *Connection Science*, 1994, 6 (4), p. 429-459.
[55]. Goller, C., et Küchler, A., Learning task-dependent distributed structure representations by backpropagation through structure. *IEEE International Conférence on Neural Networks*, 1996, p. 347-352.
[56]. Goulon A, Une nouvelle méthode d'apprentissage de données structurées : application à l'aide à la découverte de médicament, thèse, Université Pierre et Marie Curie (Paris 6) ,2008.
[57]. Hammer, B., On the approximation capability of recurrent neural networks, *Neurocomputing*, 2000, 31 (1-4), p. 107-123.
[58]. Hammer, B., Recurrent networks for structured data - A unifying approach and its properties. *Cognitive Systems Research*, 2002, 3 (2), p. 145-165.
[59]. Hammer, B., Learning with recurrent neural networks, in *Lecture Notes in Control and Information Sciences*. 2000, New York : Springer-Verlag.
[60]. Jochum, C., et Gasteiger, J., Canonical numbering and constitutional symmetry. *Journal of Chemical Information and Computer Sciences*, 1977, 17 (2), p. 113-117.
[61]. ACD/LogP, v. 10, Advanced Chemistry Development, Inc., Toronto, ON, Canada, www.acdlabs.com, 2007.
[62]. Meylan, W.M., et Howard, P.H., Atom/fragment contribution method for estimating octanol-water partition coefficients. *Journal of Pharmaceutical Sciences*, 1995, 84, p. 83-92.
[63]. SRC KOWWIN software, SRC-LOGKOW Version 1.66, Syracuse Research Corporation, Syracuse, USA, http://www.syrres.com/eSc/kowwin.htm.
[64]. Hansch, C., et Leo, A.J., Substituent constants for correlation analysis in chemistry and biology. New-York, NY: Wiley, 1979.
[65]. Gombar, V.K., Reliable assessment of logP of compounds of pharmaceutical relevance. *SAR and QSAR in Environmental Research*, 1999, 10, p. 371-380.
[66]. Devillers, J., *et al.*, Simulating lipophilicity of organic molecules with a backpropagation neural network. *Journal of Pharmaceutical Sciences*, 1998, 87 (9), p. 1086-1090.
[67]. Breindl, A., Beck, B., et Clark, T., Prediction of the n-octanol/water partition coefficient, logP, using a combination of semiempirical MO-calculations and a neural network. *Journal of Molecular Modeling*, 1997, 3, p. 142-155.
[68]. Dreyfus,G.,Martinez,J.M.,Samuelides,M.,Gordon,M.B.,Badran,F.,etThiria ,S., Apprentissage statistique, Eyrolles , 2008.