



**HAL**  
open science

# Recherche combinatoire guidée par apprentissage artificiel en chimie moléculaire

Jules Leguy

► **To cite this version:**

Jules Leguy. Recherche combinatoire guidée par apprentissage artificiel en chimie moléculaire. Informatique et langage [cs.CL]. Université d'Angers, 2022. Français. NNT : 2022ANGE0061 . tel-04053697

**HAL Id: tel-04053697**

**<https://theses.hal.science/tel-04053697>**

Submitted on 31 Mar 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THÈSE DE DOCTORAT DE

L'UNIVERSITÉ D'ANGERS

ÉCOLE DOCTORALE N° 601  
*Mathématiques et Sciences et Technologies  
de l'Information et de la Communication*  
Spécialité : *Informatique*

Par

**Jules LEGUY**

**Recherche combinatoire guidée par apprentissage artificiel  
en chimie moléculaire**

Thèse présentée et soutenue à Angers, le 9 décembre 2022

Unité de recherche : LERIA (EA 2645)

Thèse N° : 225149

## Rapporteurs avant soutenance :

Laetitia JOURDAN      Professeure des Universités, CRISTAL, Université de Lille  
Jean-Claude CRIVELLO      Chargé de recherche (HDR), ICMPE, CNRS

## Composition du Jury :

Rapporteurs :	Laetitia JOURDAN	Professeure des universités, CRISTAL, Université de Lille
	Jean-Claude CRIVELLO	Chargé de recherche (HDR), ICMPE, CNRS
Examineur :	Gaël VAROQUAUX	Directeur de recherche, INRIA, INRIA Saclay
Dir. de thèse :	Béatrice DUVAL	Professeure des universités, LERIA, Université d'Angers
Co-enc. de thèse :	Benoit DA MOTA	Maître de conférences, LERIA, Université d'Angers
Co-enc. de thèse :	Thomas CAUCHY	Maître de conférences, MOLTECH-Anjou, Université d'Angers



# REMERCIEMENTS

---

D'abord, je tiens à remercier chaleureusement ma directrice de thèse Béatrice Duval et mes deux encadrants Benoit Da Mota et Thomas Cauchy pour leur encadrement ainsi que leur soutien tout au long de ces trois années de thèse. Je mesure ma chance d'avoir eu trois encadrants aussi impliqués dans mon travail. Je leur suis reconnaissant pour leur accompagnement et leurs conseils, mais également pour leur accessibilité et les bons moments passés au laboratoire et en télétravail.

J'exprime ma gratitude envers Laetitia Jourdan et Jean-Claude Crivello pour avoir accepté de rapporter cette thèse. Je remercie les deux membres de mon comité de suivi individuel de thèse Adrien Goëffon et Gaël Varoquaux pour leurs précieux conseils. Je remercie Gaël Varoquaux une seconde fois pour avoir également accepté de faire partie de mon jury de soutenance.

Je remercie également mes camarades doctorants qui ont contribué à rendre ces trois années plaisantes. Je pense en particulier à Adrian Robert, Vincent Hénaux, Florian Delavernhe, Jintong Ren, Cyril Grelier, Thomas Saout, Minjie Li, Bryan Garreau et Corentin Béhuët. Je remercie également les membres du laboratoire avec qui j'ai eu plaisir à échanger régulièrement et notamment durant les pauses méridiennes. Sans pouvoir être exhaustif, je pense à Marc Legeay, Olivier Goudet, Antoine Jamin, Jean-Mathieu Chantrein, David Lesaint, Fabien Garreau, Éric Monfroy, Jérôme Chalain, Benjamin Jeanneau, Frédéric Saubion, Vincent Barichard ou encore Gilles Hunault avant son départ en retraite. J'ai eu le plaisir de jouer à des parties (glaciales) de Diplomatie avec certains d'entre eux ; nous nous souviendrons surtout de l'alliance fructueuse entre l'Autriche-Hongrie et la Turquie. Impossible également de ne pas remercier Catherine Pawlonski et Christine Bardaine, qui manqueront autant au reste du laboratoire qu'à moi puisque nous le quittons en même temps.

Cette thèse n'aurait pas pu voir le jour sans le soutien moral et financier inconditionnel de mes parents et beaux-parents, depuis le début de mes études supérieures et bien avant. Je les en remercie très sincèrement et je remercie également mes frères et sœurs ainsi que l'ensemble de ma famille pour son soutien. J'adresse également une pensée à ceux qui ne sont plus là.

Je remercie également mes amis, que je n'ai pas pu voir autant que je l'aurais souhaité ces dernières années mais avec qui j'ai tout de même passé d'excellents moments. Je pense en particulier à Sarah, Adel, Norman, Fablet, Gwladys et Léa. Je remercie également ma belle-famille pour son soutien ainsi que pour l'exercice physique qui a été indispensable à la rédaction de ce mémoire. Je me souviens en particulier de ma glorieuse victoire sur abandon adverse à 1-6, 0-1 au tennis contre Thomas.

Bien-sûr, je remercie aussi très sincèrement Valentine. Elle a vécu avec moi ces années de thèse et de confinements, qui n'ont pas toujours été les plus simples. Malgré tout, elle n'a cessé de me soutenir et de m'encourager.

Pour finir, j'adresse un mot de remerciement aux personnes que j'ai nécessairement mais injustement oubliées dans ces quelques paragraphes.





# NOTATIONS

---

$\mathbb{R}$	l'ensemble des nombres réels.
$\mathbb{Z}$	l'ensemble des nombres entiers.
$\mathbb{B}$	l'ensemble des valeurs booléennes.
$\{a, b, c\}$	l'ensemble contenant les éléments $a$ , $b$ et $c$ .
$[a, b]$	l'ens. des valeurs réelles comprises entre $a$ et $b$ (bornes incluses).
$X \times X'$	le produit cartésien des ensembles $X$ et $X'$ .
$X^+$	la partie positive de l'ensemble $X$ .
$X_*^+$	la partie strictement positive de l'ensemble $X$ .
$\overline{B}$	la négation de la matrice booléenne $B$ .
$ X $	la cardinalité de l'ens. $X$ (nombre d'éléments contenus dans $X$ ).
$\min_{x \in X} f(x)$	la plus petite valeur de $f(x)$ , pour tout $x$ de l'ensemble $X$ .
$\max_{x \in X} f(x)$	la plus grande valeur de $f(x)$ , pour tout $x$ de l'ensemble $X$ .
$\operatorname{argmin}_{x \in X} f(x)$	la valeur de $x$ telle que $f(x)$ est minimal, pour tout $x$ de l'ens. $X$ .
$\operatorname{argmax}_{x \in X} f(x)$	la valeur de $x$ telle que $f(x)$ est maximal, pour tout $x$ de l'ens. $X$ .
$\min(a, b)$	la valeur minimale entre les éléments $a$ et $b$ .
$\max(a, b)$	la valeur maximale entre les éléments $a$ et $b$ .
$\mathbb{E}(Y)$	l'espérance de la variable aléatoire $Y$ .
$\mathbb{P}(E)$	la probabilité de l'événement $E$ .
$\mathcal{N}(\mu, \sigma^2)$	la loi normale de moyenne $\mu$ et de variance $\sigma^2$ .
$Y \sim Z$	la variable aléatoire $Y$ suivant la loi de probabilité $Z$ .



# TABLE DES MATIÈRES

---

<b>Introduction</b>	<b>11</b>
<b>1 Contexte du domaine d'application et état de l'art</b>	<b>17</b>
1.1 Chimie et représentation des molécules . . . . .	19
1.1.1 Chimie et espaces moléculaires . . . . .	19
1.1.2 Représentation des molécules . . . . .	31
1.1.3 Descripteurs moléculaires . . . . .	40
1.1.4 Jeux de données en chimie moléculaire . . . . .	46
1.2 Apprentissage artificiel pour la chimie moléculaire . . . . .	49
1.2.1 Concepts élémentaires . . . . .	49
1.2.2 Modèles . . . . .	51
1.2.3 Apprentissage de propriétés moléculaires . . . . .	55
1.3 Optimisation et génération moléculaire . . . . .	56
1.3.1 Optimisation . . . . .	57
1.3.2 Optimisation moléculaire par perturbation d'une représentation . . . . .	64
1.3.3 Optimisation d'un espace moléculaire latent continu . . . . .	66
1.3.4 Apprentissage d'un générateur moléculaire . . . . .	67
1.4 Propriétés moléculaires et réalisme des molécules . . . . .	70
<b>2 Optimisation évolutionnaire de propriétés moléculaires</b>	<b>75</b>
2.1 Introduction . . . . .	77
2.2 EvoMol : un algorithme pour l'optimisation de propriétés moléculaires . . . . .	78
2.2.1 Conception de notre approche . . . . .	78
2.2.2 Espace de recherche . . . . .	80
2.2.3 Parcours de l'espace de recherche . . . . .	83
2.2.4 Algorithme . . . . .	87
2.3 Évaluation . . . . .	92
2.3.1 Optimisation de propriétés peu coûteuses . . . . .	92
2.3.2 Test sur le <i>benchmark</i> GuacaMol . . . . .	97

2.3.3	Optimisation de propriétés électroniques coûteuses . . . . .	105
2.4	Génération de solutions réalistes . . . . .	114
2.4.1	Optimisation des métriques d'accessibilité synthétique . . . . .	114
2.4.2	Contraintes sur l'espace de recherche . . . . .	116
2.4.3	Effet des contraintes pour l'optimisation de la QED . . . . .	118
2.4.4	Effet des contraintes sur le <i>benchmark</i> GuacaMol . . . . .	126
2.5	Conclusion et perspectives . . . . .	128
<b>3</b>	<b>Optimisation de la diversité moléculaire</b>	<b>131</b>
3.1	Introduction . . . . .	133
3.2	Quantification de la diversité moléculaire . . . . .	135
3.2.1	Mesures de diversité . . . . .	136
3.2.2	Descripteurs moléculaires . . . . .	138
3.3	Méthode . . . . .	140
3.4	Génération d'un jeu de données avec une forte diversité . . . . .	146
3.5	Optimisation conjointe de la diversité et d'une propriété moléculaire . . . . .	150
3.6	Conclusion et perspectives . . . . .	156
<b>4</b>	<b>Optimisation boîte-noire guidée par un modèle d'apprentissage de propriétés moléculaires</b>	<b>159</b>
4.1	Introduction . . . . .	161
4.2	Optimisation basée sur un modèle de substitution . . . . .	163
4.2.1	Contexte . . . . .	163
4.2.2	Algorithme . . . . .	164
4.2.3	Sélection du jeu de données initial . . . . .	165
4.2.4	Modèle de substitution et fonction de mérite . . . . .	168
4.2.5	Optimisation de la fonction de mérite . . . . .	171
4.3	Optimisation de graphes moléculaires basée sur un modèle de substitution	172
4.3.1	Méthode . . . . .	172
4.3.2	Apprentissage du modèle de substitution . . . . .	175
4.3.3	Implémentation . . . . .	176
4.3.4	Travaux liés . . . . .	178
4.4	Étude de l'optimisation d'une propriété peu coûteuse . . . . .	179
4.4.1	Apprentissage et évaluation du modèle de substitution . . . . .	179
4.4.2	Évaluation de notre méthode d'optimisation . . . . .	188

---

4.4.3	Conclusion : apprentissage et optimisation de la QED . . . . .	206
4.5	Optimisation d'une propriété électronique . . . . .	207
4.5.1	Apprentissage et évaluation du modèle de substitution . . . . .	208
4.5.2	Évaluation de notre méthode d'optimisation . . . . .	217
4.5.3	Conclusion : apprentissage et optimisation de l'énergie HOMO . . .	221
4.6	Conclusion et perspectives . . . . .	221
	<b>Conclusion générale et perspectives</b>	<b>225</b>
	<b>Bibliographie</b>	<b>233</b>
	<b>Annexes</b>	<b>247</b>
<b>A</b>	<b>Résultats supplémentaires</b>	<b>249</b>
A.1	EvoMol : optimisation du benchmark GuacaMol . . . . .	249
A.2	EvoMol : maximisation de l'énergie HOMO . . . . .	252
A.3	BBOMol : optimisation des valeurs de QED . . . . .	254
<b>B</b>	<b>Article : génération d'explications contre-factuelles</b>	<b>257</b>



# INTRODUCTION

---

Une part importante des travaux en chimie moléculaire consiste à rechercher de nouvelles molécules permettant de répondre à des problèmes applicatifs spécifiques. La recherche de médicaments qui seraient actifs spécifiquement sur une protéine d'un organisme en est un exemple. Pour découvrir de telles molécules, les chimistes cherchent traditionnellement à améliorer des molécules connues qui possèdent des propriétés proches de la cible, en effectuant des modifications locales. Leur expertise leur permet d'identifier a priori les transformations les plus pertinentes. À titre d'exemple, nous pouvons évoquer l'histoire de la molécule d'acide acétylsalicylique, qui correspond au principe actif de l'aspirine. L'écorce de saule blanc était déjà consommée par l'Homme de Néandertal pour ses propriétés antalgiques et antipyrétiques [WEYRICH et al. 2017]. Au 19<sup>ÈME</sup> siècle, des chimistes ont réussi à en isoler le principe actif, qui correspond à la molécule d'acide salicylique. Par l'étude et la modification locale de cette molécule, l'acide acétylsalicylique a été découvert en 1852 [FUSTER et SWEENY 2011]. L'acide acétylsalicylique possède des propriétés thérapeutiques plus efficaces que l'acide salicylique tout en limitant les effets indésirables. Il a ensuite été commercialisé sous le nom d'aspirine.

Dans les dernières décennies, de nouvelles approches sont apparues pour rechercher des molécules satisfaisant des propriétés cibles. Si une propriété peut être estimée par calcul ou par simulation informatique, il est possible de rechercher de manière systématique des molécules possédant des valeurs de propriété prometteuses parmi de grandes bases de données moléculaires. Cette approche nommée criblage a été rendue possible notamment par la création de bases de données contenant des millions voire des milliards de molécules. Ces molécules peuvent être connues à différents degrés. Elles peuvent correspondre à des médicaments ou à des molécules connues pour d'autres applications, ou simplement correspondre à des molécules dont on suppose la possibilité de l'existence. Cependant, ces jeux de données ne représentent en réalité qu'une infime fraction de l'espace moléculaire. Des chercheurs estiment en effet qu'il existe jusqu'à  $10^{60}$  molécules de taille moyenne, c'est-à-dire qui contiennent jusqu'à une trentaine d'atomes en excluant les atomes d'hydrogène [BOHACEK et al. 1996].

Une autre approche étudiée consiste à utiliser des méthodes d'intelligence artificielle



pour la génération de molécules satisfaisant des propriétés moléculaires. Cela correspond notamment à l'utilisation de modèles d'apprentissage profond génératifs ou de méthodes issues du domaine de l'optimisation. Ce champ de recherche a fait l'objet d'une quantité massive de travaux dans les cinq dernières années environ [FREEDMAN 2019; PAUL et al. 2021; SOUSA et al. 2021; YANG et al. 2019]. Ces méthodes suscitent un très grand enthousiasme et il est espéré qu'elles permettent de découvrir des molécules dans des zones pour le moment inconnues de l'espace moléculaire. Toutefois, ces approches sont la plupart du temps directement dépendantes d'un jeu de données moléculaires. Il est raisonnable d'attendre que cette dépendance crée implicitement un biais vers la génération de molécules proches du jeu de données de référence, et l'on peut craindre que cela entrave la génération de molécules réellement « nouvelles ».

Une difficulté importante pour la recherche automatique de molécules satisfaisant des propriétés moléculaires est que les chimistes attendent communément que les molécules proposées respectent en plus de la propriété cible un ensemble de caractéristiques. Ces objectifs annexes sont parfois sous-entendus, et sont en pratique souvent difficiles à formaliser. Les attentes les plus courantes sont la possibilité de synthétiser expérimentalement les molécules proposées ainsi que le fait qu'elles soient stables dans des conditions normales de pression et de température. Bien que ces propriétés soient difficiles à formaliser, les chimistes en ont souvent une perception intuitive qui est liée à leur expérience. Dans ce mémoire, nous parlerons de *réalisme* des molécules pour qualifier cette perception liée à une part de subjectivité. Il existe également d'autres contraintes qui sont liées à certaines applications spécifiques. Pour la chimie pharmaceutique par exemple, il existe une contrainte forte de non-toxicité des molécules proposées. Le coût de synthèse et son impact environnemental sont également des éléments d'importance.

Les travaux présentés dans cette thèse sont effectués en collaboration avec le laboratoire MOLTECH-Anjou. Il s'agit d'un laboratoire de recherche en chimie faisant partie de l'Université d'Angers et qui est également affilié au CNRS. Les chercheurs de MOLTECH-Anjou s'intéressent en particulier à la chimie des matériaux moléculaires organiques. Il s'agit d'un domaine de la chimie qui correspond aux molécules possédant des propriétés dites électroniques, qui dépendent de la répartition des électrons au sein des molécules. Parmi les applications de ce domaine de la chimie, nous pouvons citer les cellules photovoltaïques organiques ou encore les diodes électroluminescentes organiques (connues sous l'abréviation anglaise OLED). L'une des spécialités de MOLTECH-Anjou est l'analyse des relations entre structures et propriétés moléculaires. Cela revient à chercher à identifier

l'effet de groupes d'atomes spécifiques sur les propriétés observées.

Dans ce mémoire, nous cherchons à développer une méthode d'optimisation moléculaire adaptée au domaine de la chimie des matériaux moléculaires organiques. Une caractéristique importante des propriétés électroniques est que leur estimation dépend de simulations informatiques coûteuses. Dans un contexte de génération moléculaire dans un espace de recherche de très grande taille, le coût de l'estimation des propriétés cibles ajoute une difficulté supplémentaire puisqu'il limite le nombre de molécules qui peuvent être évaluées. Afin de réduire ce coût, de nombreux travaux de la littérature présentent des modèles d'apprentissage artificiel supervisé permettant de prédire les valeurs de propriétés électroniques à partir d'un apprentissage sur un jeu de données moléculaires [DERINGER et al. 2021 ; NOÉ et al. 2020]. Cependant, notre équipe a montré dans des travaux antérieurs à cette thèse qu'il peut exister un manque de diversité moléculaire dans les jeux de données de référence, qui peut altérer la capacité de généralisation de ces modèles [GLAVATSKIKH et al. 2019]. Dans un contexte de recherche de molécules nouvelles satisfaisant des propriétés moléculaires, il serait pourtant souhaitable qu'un tel modèle puisse prédire avec une précision suffisante les valeurs de propriétés de molécules quelconques.

Dans les travaux que nous présentons au sein de ce mémoire, nous considérons la recherche de molécules satisfaisant des propriétés moléculaires comme un problème d'optimisation combinatoire de graphes moléculaires. Les contributions principales sont les suivantes. Nous proposons un algorithme évolutionnaire générique et interprétable pour l'optimisation de propriétés moléculaires variées, et nous montrons qu'il est très efficace pour l'optimisation de nombreuses propriétés moléculaires. Nous définissons une méthode d'optimisation efficace pour la maximisation de la diversité moléculaire, basée sur notre algorithme évolutionnaire. Cela nous permet de générer un jeu de molécules très divers, et nous montrons également que l'objectif de diversité peut apporter un gain d'efficacité lors de l'optimisation d'une propriété moléculaire. Finalement, nous proposons une méthode d'optimisation basée sur un modèle de substitution pour l'optimisation efficace de propriétés moléculaires coûteuses. Le modèle de substitution est un modèle d'apprentissage artificiel supervisé permettant une estimation rapide des valeurs de la propriété cible. Nous menons une étude étendue de notre approche pour l'optimisation d'une propriété peu coûteuse, puis nous montrons qu'elle permet effectivement l'optimisation efficace d'une propriété électronique dont l'évaluation est coûteuse.

Le contenu de ce mémoire est organisé de la manière suivante.

— Dans le premier chapitre, nous abordons les concepts essentiels en chimie pour la

compréhension et la motivation de nos travaux. Nous effectuons également un état de l'art de notre domaine de recherche, à la fois en informatique et en chimie-informatique. En informatique, nous nous intéressons en particulier aux domaines de l'apprentissage artificiel supervisé ainsi qu'au domaine de l'optimisation combinatoire. En chimie-informatique, nous nous intéressons notamment aux représentations moléculaires ainsi qu'aux approches de génération moléculaire.

- Dans le deuxième chapitre, nous proposons un algorithme évolutionnaire générique pour l'optimisation de propriétés moléculaires issues de différents domaines de la chimie. Nous considérons le problème de génération de molécules satisfaisant des propriétés moléculaires comme un problème d'optimisation combinatoire de graphes moléculaires. Nous limitons les biais intégrés dans la version de base de notre approche afin qu'elle ne soit pas associée à un domaine de la chimie en particulier. De plus, notre approche est conçue pour permettre une interprétabilité forte des résultats par les utilisateurs du domaine d'application. Nous proposons une visualisation de la recherche sous la forme d'un arbre d'exploration. Finalement, nous proposons une étude de différentes contraintes permettant de restreindre l'espace de recherche accessible afin d'améliorer le réalisme des molécules proposées.
- Dans le troisième chapitre, nous définissons une méthode de maximisation de la diversité moléculaire. Notre objectif initial est de générer un jeu de données moléculaires possédant une grande diversité, afin de favoriser les capacités de généralisation d'éventuels modèles d'apprentissage artificiel de propriétés moléculaires qui l'utiliseraient en tant que jeu de données d'entraînement. Notre approche est basée sur l'algorithme évolutionnaire que nous proposons au deuxième chapitre, pour lequel nous définissons une fonction objectif correspondant à une approximation efficace de la contribution à la diversité totale de la population. Nous montrons que lorsqu'il est combiné à un objectif correspondant à une propriété moléculaire que l'on souhaite optimiser, l'objectif de diversité permet une optimisation plus efficace de la propriété cible que lorsqu'elle est optimisée seule. Nous montrons ainsi que l'objectif de diversité favorise l'exploration de l'espace de recherche.
- Dans le quatrième chapitre, nous cherchons à améliorer l'efficacité de la recherche pour l'optimisation de propriétés électroniques coûteuses. Nous proposons pour cela d'utiliser un algorithme d'optimisation basée sur un modèle de substitution. Le modèle de substitution est un modèle d'apprentissage artificiel supervisé permettant une estimation rapide des valeurs de la propriété cible. Il est utilisé au

sein d'une procédure d'optimisation qui permet de limiter les appels à la fonction objectif coûteuse. Cette procédure dépend de l'algorithme évolutionnaire que nous avons proposé au deuxième chapitre afin de sélectionner des solutions candidates selon les valeurs prédites par le modèle de substitution. En tant que modèle de substitution, nous utilisons un modèle de régression par processus gaussien, dont la prédiction probabiliste est mise à profit pour guider la recherche de candidats avec un meilleur contrôle. Les travaux présentés au sein de ce mémoire font partie d'un domaine de recherche très dynamique. Dans la conclusion, nous discuterons de différents développements qui peuvent être poursuivis après cette thèse, dont certains pour lesquels des travaux préliminaires ont déjà été effectués.



# CONTEXTE DU DOMAINE D'APPLICATION ET ÉTAT DE L'ART

---

## Sommaire

---

<b>1.1 Chimie et représentation des molécules . . . . .</b>	<b>19</b>
1.1.1 Chimie et espaces moléculaires . . . . .	19
1.1.2 Représentation des molécules . . . . .	31
1.1.3 Descripteurs moléculaires . . . . .	40
1.1.4 Jeux de données en chimie moléculaire . . . . .	46
<b>1.2 Apprentissage artificiel pour la chimie moléculaire . . . . .</b>	<b>49</b>
1.2.1 Concepts élémentaires . . . . .	49
1.2.2 Modèles . . . . .	51
1.2.3 Apprentissage de propriétés moléculaires . . . . .	55
<b>1.3 Optimisation et génération moléculaire . . . . .</b>	<b>56</b>
1.3.1 Optimisation . . . . .	57
1.3.2 Optimisation moléculaire par perturbation d'une représentation . . . . .	64
1.3.3 Optimisation d'un espace moléculaire latent continu . . . . .	66
1.3.4 Apprentissage d'un générateur moléculaire . . . . .	67
<b>1.4 Propriétés moléculaires et réalisme des molécules . . . . .</b>	<b>70</b>

---

Ce chapitre fait l'objet de la publication suivante.

[LEGUY et al. 2022a]



Dans ce premier chapitre, nous présentons le contexte et les concepts essentiels de notre domaine d'application, c'est-à-dire la chimie moléculaire organique. Nous effectuons également un état de l'art des travaux en informatique et en chimie-informatique qui forment notre domaine de recherche et sur lesquels les travaux que l'on développe au sein de ce mémoire vont s'appuyer. Nous nous intéressons notamment aux domaines de l'apprentissage artificiel et de l'optimisation ainsi qu'à leurs applications en chimie moléculaire.

Nous organisons ce chapitre de la façon suivante. D'abord, nous introduisons l'ensemble des concepts en chimie indispensables à la motivation et la compréhension de nos travaux, et nous nous intéressons en particulier à la façon dont les molécules peuvent être représentées informatiquement. Nous présentons également des jeux de données moléculaires de référence. Par la suite, nous nous intéressons au domaine de l'apprentissage artificiel et à ses applications en chimie-informatique. Finalement, nous présentons le domaine de l'optimisation et ses applications en chimie pour la recherche de molécules satisfaisant des propriétés moléculaires. Plus généralement, nous nous intéressons aux méthodes qui entrent dans le cadre de la génération moléculaire, dont un certain nombre est basé sur des modèles d'apprentissage artificiel. Nous concluons ce chapitre par la présentation de propriétés moléculaires communément manipulées dans le cadre de ces méthodes de génération.

## 1.1 Chimie et représentation des molécules


Dans cette section, nous présentons les concepts élémentaires en chimie moléculaire, à savoir notamment les atomes, les liaisons, les molécules ainsi que les différents domaines de la chimie. Par la suite, nous étudions un ensemble de représentations moléculaires formelles permettant de manipuler des molécules de manière systématique. Nous distinguons les représentations des descripteurs moléculaires, qui sont utilisés notamment dans le cadre de l'apprentissage artificiel. Finalement, nous présentons plusieurs jeux de données moléculaires de référence dans la littérature.

### 1.1.1 Chimie et espaces moléculaires

En chimie, et plus précisément en chimie moléculaire, la molécule est l'objet d'étude de base. Elle est composée d'atomes associés entre eux par des interactions fortes appelées liaisons. Dans ce mémoire, nous allons nous intéresser plus précisément à la chimie



H																	He
Li	Be											B	C	N	O	F	Ne
Na	Mg											Al	Si	P	S	Cl	Ar
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
Cs	Ba	*	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
Fr	Ra	**	Rf	Db	Sg	Bh	Hs	Mt	Ds	Rg	Uub	Uut	Uuq	Uup	Uuh	Uus	Uuo
		* lanthanides	La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu
		** actinides	Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr

FIGURE 1.1 – Tableau périodique des éléments chimiques avec mise en évidence des éléments de la chimie organique, centrée sur l'atome de carbone (C). En bleu et bleu-vert, éléments considérés au cœur de la chimie organique. En violet, éléments que l'on considère dans une définition plus étendue de la chimie organique. En rose, éléments dont la considération pourrait également être justifiée. Image adaptée de Wikimedia (Cdang) .

organique moléculaire, qui correspond à l'étude des molécules formées à partir d'un sous-ensemble des éléments chimiques. Il s'agit de la chimie principale des organismes biologiques, centrée autour de l'atome de carbone. La chimie organique étant notre unique domaine d'application, nous utiliserons parfois plus simplement le terme de chimie pour désigner implicitement la chimie organique.

## Atomes

**Composition** Un atome est composé d'un noyau, dont le nombre de protons (particules chargées positivement) définit son type. Le nombre de protons est appelé numéro atomique. Un atome est également composé d'un ensemble d'électrons (particules chargées

négativement) qui orbitent à proximité du noyau. Un atome est neutre électriquement, ce qui signifie qu'il possède autant de protons que d'électrons. Lorsque le nombre d'électrons diffère du nombre de protons, on parle alors d'un ion, qui est chargé négativement ou positivement. Le terme d'élément chimique, basé uniquement sur le nombre de protons, permet de se référer génériquement à un atome ou un ion. Afin de simplifier les appellations au sein de ce mémoire, nous serons susceptibles d'utiliser abusivement le terme d'« atome chargé » pour qualifier un ion. Deux atomes à proximité l'un de l'autre ont la possibilité de partager une ou plusieurs paires d'électrons, afin de maximiser l'attraction entre électrons et noyaux. On parle alors de la formation d'une liaison covalente. Cela permet à l'ensemble moléculaire d'avoir une énergie totale plus faible, ce qui correspond à une meilleure stabilité.

**Tableau périodique** La Figure 1.1 représente le tableau périodique des éléments, c'est-à-dire l'ensemble des éléments connus ou dont l'existence est supposée. Les éléments sont organisés de telle sorte que les éléments d'une colonne possèdent généralement des propriétés similaires. Les éléments colorés du tableau périodique correspondent aux éléments de la chimie organique ou assimilables à la chimie organique. Parmi les éléments qui ne font pas partie de la chimie organique, représentés en gris, on retrouve notamment les métaux. Les métaux correspondent à toute la partie inférieure et gauche du tableau périodique. En gris, on retrouve également une partie des metalloïdes, qui sont des éléments possédant des propriétés qui les classent entre le groupe des métaux et des non-métaux. Ce groupe contient entre autres les éléments Ge, As, Sb, Te et At. La dernière famille dont les éléments sont représentés en gris correspond aux gaz rares, situés dans la dernière colonne à droite du tableau. Les gaz rares sont des éléments extrêmement stables, qui ne peuvent généralement pas former de liaison covalente et que l'on ne retrouve donc généralement pas dans des molécules.

**Chimie organique** Les différentes couleurs de la Figure 1.1 nous permettent de définir plusieurs sous-ensembles des atomes de la chimie organique. En bleu, nous représentons les quatre atomes au cœur de la chimie organique (H, C, N, et O). Dans la suite de ce mémoire, nous utiliserons régulièrement un sous-ensemble restreint des atomes de la chimie organique, qui contient en plus de ces éléments le fluor (F), représenté en bleu-vert. Ce sous-ensemble est basé sur le choix effectué lors de la conception de QM9, un jeu de données de référence de l'état de l'art (voir la section 1.1.4 de ce chapitre). Nous uti-

liserons également un ensemble d’éléments correspondant à une définition plus étendue de la chimie organique. Cet ensemble contient en plus des éléments en bleu et bleu-vert, les quatre éléments en violet (P, S, Cl et Br). Cela forme un ensemble d’éléments communément accepté par les chimistes comme relevant de la chimie organique. Cependant, les frontières de la chimie organique sont discutables. Ainsi, nous aurions également pu inclure les éléments représentés en rose, à savoir B, Si, Se et I. À l’exception de l’iode (I), ces éléments sont également des métalloïdes.

La chimie organique est étudiée par des chimistes appartenant à diverses communautés. Elle est étudiée en chimie pharmaceutique car elle correspond à la chimie principale des organismes biologiques. Au sein de ce mémoire, nous nous intéresserons également au domaine de la chimie des matériaux moléculaires organiques. Il s’agit du champ d’étude des molécules de la chimie organique possédant des propriétés telles que l’absorption ou l’émission de photons (lumière), ou encore des propriétés photovoltaïques. Ces propriétés, dites électroniques, sont très liées à l’organisation des électrons au sein de la molécule.

En Table 1.1, nous reportons un ensemble de caractéristiques des types d’atomes que nous allons manipuler au sein de ce mémoire. Ce tableau permet de faire le lien entre le nom des atomes et le symbole qui les représente usuellement, comme au sein du tableau périodique. Nous définissons également deux termes pour évoquer simplement des groupes de types d’atomes. Les atomes dits « lourds » correspondent à l’ensemble de tous les types d’atomes à l’exception de l’hydrogène. Les hétéroatomes correspondent à l’ensemble de tous les atomes lourds, à l’exception du carbone.

## Liaisons

**Valence** En Table 1.1, nous reportons également pour chaque type d’atome sa valence. La valence d’un atome correspond au nombre de liaisons qu’il peut former, ou plus exactement au nombre d’électrons qu’il peut partager avec un autre atome. Notons la capacité de l’atome de carbone à former 4 liaisons, ce qui lui permet de former des structures complexes et variées. Notons également que le phosphore et le soufre ont une valence usuelle de 3 et 2, mais qu’ils peuvent partager jusqu’à 5 et 6 électrons respectivement. Plus précisément, nous considérons que l’atome de phosphore peut partager 3 ou 5 électrons, et que l’atome de soufre peut en partager 2, 4 ou 6. En pratique, nous allons considérer que la valence de ces types d’atomes est dépendante du nombre de liaisons qu’ils forment de manière explicite, c’est-à-dire avec des atomes lourds. Si un atome de phosphore lie 0, 1, 2 ou 3 de ses électrons avec des atomes lourds alors nous considérons que sa valence est 3.

Nom	Symb.	Num.	Valence	Lourd	Hétéroatome	Repr.	F. brute
Hydrogène	H	1	1				
Carbone	C	6	4	✓		— C <sub>2</sub> H <sub>6</sub>	
Azote	N	7	3	✓	✓	—NH <sub>2</sub> CH <sub>5</sub> N	
Oxygène	O	8	2	✓	✓	—OH CH <sub>4</sub> O	
Fluor	F	9	1	✓	✓	—F CH <sub>3</sub> F	
Phosphore	P	15	3, 5	✓	✓	—PH <sub>2</sub> CH <sub>5</sub> P	
Soufre	S	16	2, 4, 6	✓	✓	—SH CH <sub>4</sub> S	
Chlore	Cl	17	1	✓	✓	—Cl CH <sub>3</sub> Cl	
Brome	Br	35	1	✓	✓	—Br CH <sub>3</sub> Br	

TABLE 1.1 – Tableau récapitulatif des atomes de la chimie organique considérés dans ce mémoire. Partie centrale : symbole, numéro atomique et valence (nombre de liaisons) des différents atomes. Les coches permettent de définir les ensembles d'atomes lourds et d'hétéroatomes. Partie droite : dessin moléculaire et formule brute de chaque atome lorsqu'il est lié par une liaison simple à un atome de carbone. Pour la ligne correspondant à l'atome de carbone, cela correspond à la molécule d'éthane. Pour la ligne correspondant à l'atome d'azote, cela correspond à la molécule de méthylamine.

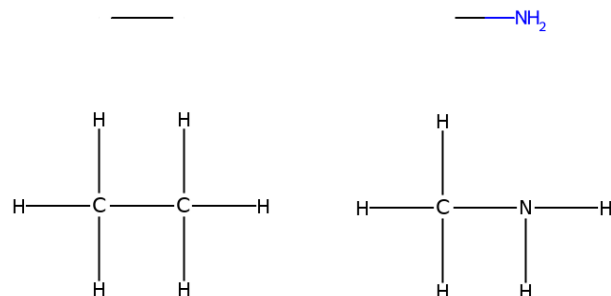


FIGURE 1.2 – Partie supérieure : dessin moléculaire de la molécule d'éthane (à gauche) et de méthylamine (à droite). Partie inférieure : dessins moléculaires explicites correspondants (formules développées).

S'il en lie strictement plus de 3 alors nous considérons que sa valence est 5. Pour l'atome de soufre, s'il lie 0, 1 ou 2 électrons avec des atomes lourds alors nous considérons que sa valence est 2. S'il en lie 3 ou 4 alors nous considérons que sa valence est 4. Finalement, s'il en lie strictement plus de 4 alors nous considérons que sa valence est 6. Après avoir introduit les dessins moléculaires dans le paragraphe suivant, nous proposerons une visualisation de ce comportement.

**Dessin moléculaire** La Table 1.1 contient également des exemples de représentation sous forme de dessin moléculaire des atomes, dans un contexte où ils forment une liaison impliquant un unique électron avec un atome de carbone. Nous faisons le choix de les présenter dans ce contexte, puisque les atomes seront toujours ou presque liés à d'autres atomes lourds dans les représentations de molécules que nous présenterons au sein de ce mémoire. Dans les dessins moléculaires, les atomes de carbone ne sont pas représentés par leur symbole lorsqu'ils sont liés à un atome lourd. Ils sont représentés de façon implicite, par la présence d'une liaison de couleur noire si elle implique deux atomes de carbone (ligne Carbone), ou d'une liaison de couleur noire du côté de la liaison contenant un atome de carbone, et de la couleur du second atome sur la deuxième moitié (ligne Azote, par exemple). Ces deux exemples correspondent en réalité à des molécules qui existent réellement, à savoir la molécule d'éthane et de méthylamine respectivement. En Figure 1.2, nous représentons ces deux molécules selon deux formes de représentation graphique. Nous reportons d'abord les dessins moléculaires de ces deux molécules, qui sont également présentés au sein de la Table 1.1. Dans la deuxième représentation, que nous nommons dessin moléculaire explicite en opposition à la première représentation, tous les

atomes et toutes les liaisons sont représentés explicitement. Le dessin moléculaire explicite correspond à une représentation communément nommée formule développée par les chimistes, en opposition au dessin moléculaire qui est équivalent à la notation sous forme de formule semi-développée. Au sein de ce mémoire, nous utiliserons exclusivement la représentation sous forme de dessin moléculaire (formule semi-développée) car elle est plus compacte. La comparaison des deux représentations nous permet ici de comprendre plus aisément les éléments qui sont représentés implicitement dans les dessins moléculaires.

Dans les dessins moléculaires, les atomes d'hydrogène ne sont pas représentés lorsqu'ils sont liés à un atome de carbone, mais sont sous-entendus. Ainsi, chacun des deux atomes de carbone du dessin moléculaire de l'éthane est également lié à 3 atomes d'hydrogène non représentés, de sorte que la valence des atomes est respectée (deuxième ligne de la Table 1.1). Chacun des atomes de carbone forme ainsi une liaison explicite avec l'autre atome de carbone et trois liaisons implicites avec des atomes d'hydrogène. Pour les hétéroatomes, la présence d'une liaison avec un ou plusieurs atomes d'hydrogène est en revanche marquée explicitement sur les dessins moléculaires. Les liaisons ne sont pas représentées, mais le symbole H ainsi que le nombre d'atomes d'hydrogène liés est accolé au symbole de l'hétéroatome. On observe ainsi que la présence de deux liaisons entre l'atome d'azote et les atomes d'hydrogène pour la molécule de méthylamine apparaît explicitement dans le dessin moléculaire de la molécule en partie supérieure de la Figure 1.2.

Les dessins moléculaires nous permettent de visualiser le comportement de la valence des atomes de phosphore et de soufre, qui dépend du nombre d'électrons impliqués dans des liaisons explicites formées avec des atomes lourds (voir le paragraphe Valence). Ces dessins moléculaires sont représentés en Figure 1.3, pour un nombre de liaisons explicites variant de 1 à 5 pour le phosphore et de 1 à 6 pour le soufre. Pour le phosphore, on observe ainsi que dans les quatre premiers dessins, lorsque le nombre d'électrons impliqués dans des liaisons explicites est compris entre 0 et 3, la valence de l'atome est 3. Pour rappel, la valence prend en compte les liaisons formées explicitement avec les atomes lourds et les liaisons formées implicitement avec des atomes d'hydrogène. Lorsque le nombre d'électrons impliqués dans des liaisons est supérieur, la valence est 5. Pour l'atome de soufre, nous observons un comportement très similaire. Lorsque le nombre d'électrons impliqués dans des liaisons explicites est compris entre 0 et 2, la valence de l'atome est 2. Lorsqu'il est compris entre 3 et 4, la valence est 4 et lorsqu'il est supérieur alors la valence est 6.

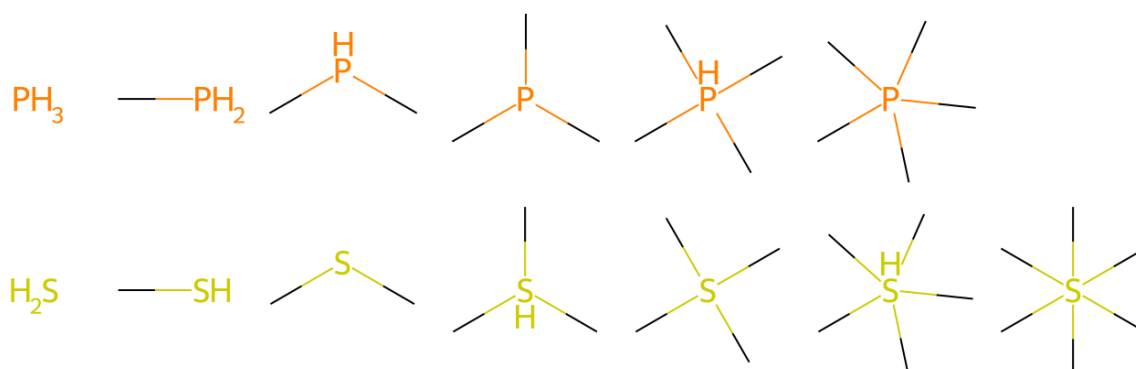


FIGURE 1.3 – Dessins moléculaires d'atomes de phosphore formant 0 à 5 liaisons explicites avec des atomes de carbone (première ligne) et d'atomes de soufre formant 0 à 6 liaisons explicites avec des atomes de carbone (seconde ligne)

**Formule brute** En Table 1.1, nous représentons finalement la formule brute des molécules de petite taille formées par la liaison des différents types d'atomes avec un atome de carbone. La formule brute correspond à la concaténation des symboles des atomes contenus, en faisant suivre chaque type d'atome par son nombre d'occurrences dans la molécule s'il est présent plusieurs fois. À titre d'exemple, la molécule d'eau qui est composée d'un atome d'oxygène et deux atomes d'hydrogène a pour formule brute  $H_2O$ . La molécule d'éthane est composée de deux atomes de carbone formant chacun trois liaisons avec des atomes d'hydrogène (voir la Figure 1.2). Ainsi, elle possède comme formule brute  $C_2H_6$ . De façon très similaire, la molécule de méthylamine possède comme formule brute  $CH_5N$ , car elle est composée d'un atome de carbone qui forme trois liaisons implicites avec des atomes d'hydrogène, et un atome d'azote qui forme deux liaisons implicites avec des atomes d'hydrogène. Il est intéressant de remarquer qu'en ligne 6 de la Table 1.1, l'atome de phosphore ne forme qu'une unique liaison explicite impliquant un unique électron avec un atome lourd. Selon la règle énoncée dans le paragraphe Valence, cet atome possède une valence de 3 car strictement moins de 4 électrons sont impliqués dans des liaisons explicites. La formule brute est donc  $CH_5P$  puisque l'atome de carbone forme trois liaisons implicites avec des atomes d'hydrogène et l'atome de phosphore en forme 2. De façon très similaire, on peut déduire dans la ligne suivante que la valence de l'atome de soufre est 2 et qu'il forme donc une unique liaison implicite en plus des trois liaisons formées par l'atome de carbone. Par conséquent, la formule brute est  $CH_4S$ .




Liaison	Représentation	Formule brute
Simple		CH <sub>5</sub> N
Double		CH <sub>3</sub> N
Triple		CHN

TABLE 1.2 – Types de liaisons, représentations graphiques sous forme d’un dessin moléculaire et formules brutes correspondantes, avec l’exemple d’une liaison entre un atome de carbone et un atome d’azote.

**Types de liaisons** Il existe plusieurs types de liaisons covalentes, qui dépendent du nombre d’électrons partagés par les atomes impliqués dans la liaison. Dans la Table 1.1, nous avons représenté le cas de liaisons dites « simples », qui correspond au partage d’une unique paire d’électrons. Il existe également des liaisons correspondant au partage de deux et trois paires d’électrons. Ces liaisons sont nommées respectivement « doubles » et « triples ». Chimiquement, plus une liaison implique d’électrons, plus elle est forte. Cela implique que les atomes sont plus près dans l’espace, et que l’énergie nécessaire pour casser la liaison est plus élevée. En réalité, la discrétisation des liaisons sous la forme simple, double ou triple est une simplification des interactions électroniques qui ont lieu dans un espace géométrique continu au sein des molécules. Cela permet toutefois de représenter de façon pertinente la majorité des interactions, et permet de simplifier grandement la représentation des molécules. En Table 1.2, nous représentons le dessin moléculaire ainsi que la formule brute d’une liaison entre un atome de carbone et un atome d’azote selon le type de la liaison. Nous pouvons remarquer que plus la liaison est forte, moins il existe d’électrons disponibles pour former des liaisons implicites avec des atomes d’hydrogène et donc moins la molécule formée contient d’atomes d’hydrogène. Pour la liaison simple, l’atome de carbone forme trois liaisons implicites et l’atome d’azote en forme deux. Pour la liaison double, l’atome de carbone forme deux liaisons implicites et l’atome d’azote n’en forme qu’une. Pour la liaison triple, l’atome de carbone ne forme qu’une liaison implicite et l’atome d’azote n’en forme aucune.

## Molécules

Chimiquement, une molécule est un ensemble d’atomes qui forment des liaisons covalentes dans un espace géométrique. La formation de liaisons stabilise énergétiquement



l'ensemble par rapport à des atomes considérés indépendamment. Nous utilisons une définition générique des atomes, selon laquelle les ions sont considérés comme des atomes chargés. Par conséquent, nous considérons qu'une molécule peut contenir des atomes neutres ou des ions. Notons qu'en chimie, on parle généralement d'un ion polyatomique ou simplement d'un ion pour désigner une molécule qui contient un ou plusieurs ions. Si la charge totale est nulle (la molécule contient autant de charges positives que négatives), on parle d'un zwitterion. Dans notre domaine d'application, qui est l'utilisation de méthodes d'intelligence artificielle pour la recherche de molécules satisfaisant des propriétés, les chimistes sont en réalité susceptibles de rechercher des molécules neutres, mais également des ions et des zwitterions. Pour cette raison, nous choisissons de désigner tous ces objets par le terme de molécule.

**Radical** Les ensembles d'atomes présentés jusqu'à présent sont composés d'atomes formant un nombre de liaisons correspondant à leur valence. Cela signifie que tous les électrons qui sont susceptibles de s'apparier forment une liaison covalente avec un autre atome. Cependant, il existe dans le monde réel des cas où l'un ou plusieurs de ces électrons est non apparié. On parle dans ce cas d'un radical plutôt que d'une molécule. Dans le cadre des problèmes de génération moléculaire que nous serons amenés à traiter au sein de ce mémoire, les radicaux ne correspondent pas à des structures chimiques pertinentes. Ils sont cependant susceptibles d'être présents dans les bases de données moléculaires.

**Aromaticité** Les atomes d'une molécule peuvent former des cycles. Certains cycles induisent une propriété reconnue expérimentalement nommée aromaticité. En réalité, ce terme recouvre plusieurs phénomènes et son évaluation est toujours en débat. L'une des approches pour la caractériser est de compter le nombre d'électrons délocalisables partagés au sein d'un cycle. On considère ainsi qu'un cycle est aromatique s'il existe un entier naturel  $n$  tel que ce nombre est égal à  $4n + 2$ , avec  $n$  un entier naturel quelconque, selon une règle dérivée des travaux de [HÜCKEL 1931]. La notion d'électron délocalisable fait intervenir des notions de chimie quantique qui sont hors de la portée de ce mémoire. Nous traiterons l'aromaticité comme une propriété moléculaire binaire détectable à l'aide d'une fonction implémentée dans la bibliothèque logicielle RDKit [LANDRUM 2010]. À titre d'exemple, et bien que notre définition n'est pas suffisante pour nous permettre d'identifier l'aromaticité, nous représentons en Figure 1.4 quatre molécules cycliques dont deux sont aromatiques. Le benzène (a) ainsi que le thiophène (b) possèdent un cycle

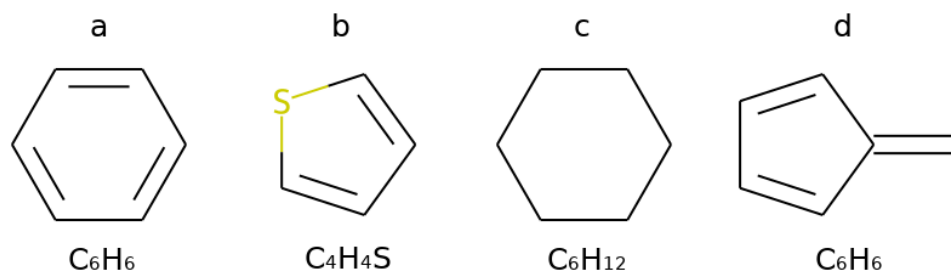


FIGURE 1.4 – Dessin moléculaire et formule brute des molécules de benzène (a), thiophène (b), cyclohexane (c) et fulvène (d).

aromatique. Le benzène est la molécule typique et représentative de l'aromaticité. Le cyclohexane (c) et le fulvène (d) ne possèdent pas de cycle aromatique. Notons que les molécules de benzène et de cyclohexane diffèrent uniquement par les types de liaisons qui sont formées. Cette différence apparaît également dans la formule brute avec les 6 atomes d'hydrogène supplémentaires dans le cyclohexane, qui correspondent aux 6 électrons qui seraient impliqués pour transformer 3 liaisons simples du cyclohexane en 3 liaisons doubles et ainsi former la molécule de benzène.

### Espaces moléculaires

L'espace moléculaire correspond à un espace contenant l'ensemble des molécules. Il s'agit d'un concept assez intuitif, qui représente un espace virtuel dans lequel les chimistes recherchent des molécules pour répondre à des problématiques. Cependant, la définition formelle de cet espace est très complexe. Pour traiter des molécules de façon automatique dans un système informatique, il est nécessaire de définir une représentation formelle des molécules. Cependant, il n'existe pas de représentation moléculaire universelle. La représentation des molécules sous la forme d'un objet dans un espace géométrique permet au mieux la modélisation des interactions entre les électrons en chimie quantique. Souvent, les chimistes utilisent une représentation simplifiée basée sur la discrétisation des types de liaisons et dépourvue d'informations géométriques. Cette représentation est à la base des dessins moléculaires que nous avons présentés précédemment. Les espaces moléculaires que l'on peut définir à partir de ces deux représentations ne sont pas équivalents. Il faut retenir que l'espace moléculaire est un concept intuitif mais qu'en l'absence de définition formelle cet espace ne peut être exploré de manière automatique. Pour cela, il est nécessaire de définir une représentation moléculaire formelle, qui conditionne les caractéristiques qui

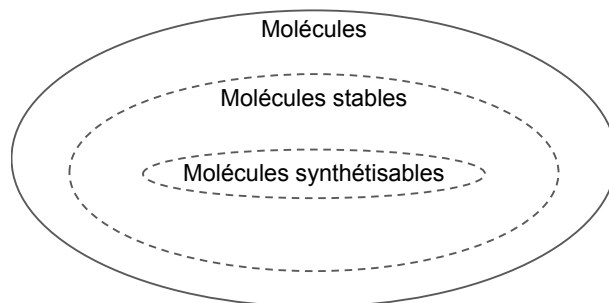


FIGURE 1.5 – Représentation schématique de l'espace moléculaire et de sous-espaces pertinents dans le cadre de la recherche de molécules satisfaisant des propriétés moléculaires.

peuvent être représentées et qui définit donc un sous-ensemble de l'espace moléculaire. Nous décrivons plusieurs de ces représentations dans la suite de cette section.

**Taille** Bien que la définition d'espace moléculaire soit complexe à établir, il est reconnu qu'il est de très grande taille. Il existe plusieurs estimations de la taille de l'espace moléculaire dans la littérature. Par raisonnement combinatoire, des auteurs estiment qu'il existe entre  $10^{33}$  et  $10^{60}$  molécules organiques de taille moyenne, contenant jusqu'à une trentaine d'atomes lourds [BOHACEK et al. 1996 ; POLISHCHUK et al. 2013]. Dans le cadre où l'on cherche des molécules optimales selon une propriété dans cet espace de recherche, cela rend inenvisageable une approche par énumération, et justifie la définition de méthodes pour l'optimisation efficace au sein de l'espace moléculaire.

**Sous-espaces moléculaires** Parmi la totalité des molécules que l'on peut imaginer, une très faible fraction est réellement pertinente à considérer lors de la recherche de molécules satisfaisant des propriétés moléculaires cibles. La raison est d'abord que pour répondre à la plupart des problématiques dans le monde réel, la molécule doit impérativement être stable dans des conditions normales de pression et de température. Cela exclut une grande partie de l'espace moléculaire. Il est cependant difficile de caractériser ce sous-espace ainsi que d'estimer sa taille. La raison est que l'estimation de la stabilité dépend de calculs coûteux voire très coûteux en chimie quantique (voir la section 1.1.2 de ce chapitre). De plus, afin de pouvoir être utilisées en pratique, les molécules doivent être synthétisées expérimentalement. Or, seule une partie des structures moléculaires que l'on peut imaginer peuvent être synthétisées selon la connaissance des chimistes. Cette connaissance évolue continuellement puisque la recherche de voies de synthèses est un pan important de la

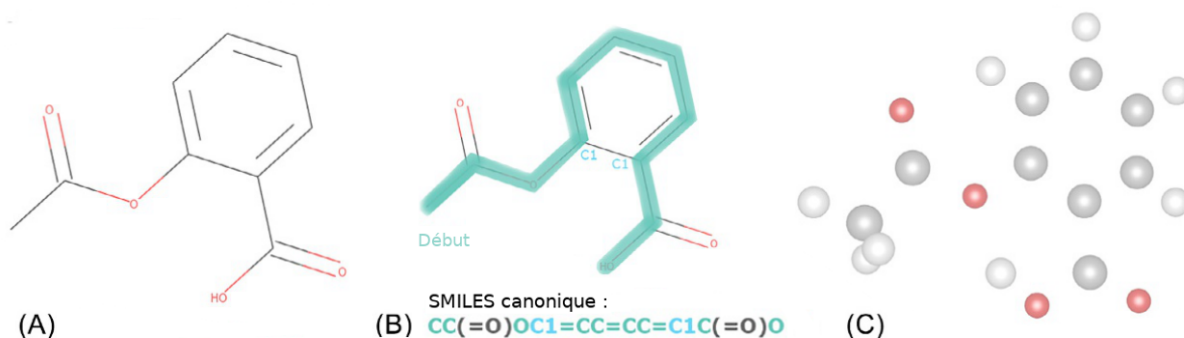


FIGURE 1.6 – Représentations moléculaires de la molécule d’acide acétylsalicylique. A : graphe moléculaire, B : SMILES et parcours de graphe correspondant, C : nuage d’atomes 3D (gris : carbone, rouge : oxygène, blanc : hydrogène).

recherche en chimie. En Figure 1.5, nous représentons de manière schématique l’espace moléculaire et ces deux sous-espaces, afin de faire apparaître les relations d’inclusion. Dans ce schéma, nous considérons uniquement l’espace des molécules synthétisables qui sont également stables. En réalité, il est possible de synthétiser brièvement certaines molécules instables. Dans un contexte de recherche de molécules satisfaisant des propriétés moléculaires pour répondre à des problèmes réels, ces solutions ne seront a priori pas utilisables en pratique et par conséquent nous choisissons de ne pas les considérer au sein de ce schéma. Les bordures internes sont représentées par des tirets pour faire apparaître que la caractérisation de ces espaces est un problème complexe et que les frontières peuvent être poreuses.

### 1.1.2 Représentation des molécules

Dans la section précédente, nous avons donné une définition générique des molécules. Pour définir un espace de recherche exploitable par un algorithme de génération de molécules, ou plus généralement pour traiter des molécules de façon automatique, il est nécessaire de définir une représentation moléculaire formelle. Dans cette section, nous présentons les représentations moléculaires majeures. Il s’agit premièrement de la représentation sous forme de graphe moléculaire. Il s’agit également des représentations sous forme de texte qui correspondent à un parcours du graphe moléculaire, ou encore de la représentation géométrique sous la forme d’un nuage de noyaux atomiques. Pour cette dernière, nous présentons également plusieurs méthodes de calcul et estimons leur coût.

## Représentation sous forme de graphe moléculaire

Considérer les molécules comme un graphe dans lequel les atomes sont les sommets et les liaisons sont des arêtes est une représentation très naturelle pour les chimistes. Il s’agit d’ailleurs de la représentation utilisée dans les dessins moléculaires. En partie (A) de la Figure 1.6, nous représentons le dessin moléculaire correspondant au graphe moléculaire de la molécule d’acide acétylsalicylique, commercialisée comme médicament sous le nom d’aspirine.

Formellement, nous pouvons représenter un graphe moléculaire comme un quintuplet  $G = (V, E, f_a, f_b, f_c)$ , avec  $V$  l’ensemble des sommets (atomes),  $E$  l’ensemble des arêtes (liaisons),  $f_a : V \rightarrow \{C, N, O, F, \dots\}$  une fonction étiquetant les types d’atomes,  $f_b : E \rightarrow \{1, 2, 3\}$  une fonction étiquetant les types de liaisons et  $f_c : V \rightarrow \mathbb{Z}$  une fonction étiquetant les charges formelles. Un graphe moléculaire ne correspond pas nécessairement à une molécule valide. Notre définition ne prend en effet pas en compte le respect des règles de valence. Souvent, les atomes d’hydrogène sont représentés de manière implicite au sein des graphes moléculaires. Chaque sommet représentant un atome lourd est implicitement lié à autant d’atomes d’hydrogène que nécessaire afin que sa valence soit atteinte. Cela permet de simplifier le traitement automatisé des graphes moléculaires, et notamment pour vérifier le respect des règles de valence. Lorsque les atomes d’hydrogène sont représentés de manière implicite, il est uniquement nécessaire de vérifier que la valence des atomes lourds n’est pas dépassée. Nous y reviendrons au sein du Chapitre 2. Au sein de ce mémoire, nous considérerons systématiquement les atomes d’hydrogène de manière implicite lorsque nous serons amenés à manipuler des graphes moléculaires. Notons que lorsque les graphes moléculaires sont représentés sous forme d’un dessin moléculaire, comme en partie (A) de la Figure 1.6, seuls les atomes d’hydrogène liés à des atomes de carbone sont représentés de manière implicite. Dans la représentation interne des graphes moléculaires en revanche, tous les atomes d’hydrogène sont considérés implicitement. Cela inclurait donc l’atome d’hydrogène lié à l’un des atomes d’oxygène de la molécule d’acide acétylsalicylique.

Dans les graphes moléculaires, les liaisons impliquant des atomes lourds sont représentées explicitement et sont associées à un type discret (simple, double ou triple). Or, il existe des structures moléculaires qui ne peuvent pas être représentées aisément par cette discrétisation. La molécule de nitrométhane en est un exemple. Elle est composée d’un atome d’azote central de charge positive, qui peut partager 4 électrons de valence et non 3 en raison de sa charge. L’un de ces électrons de valence est lié à un atome de carbone,

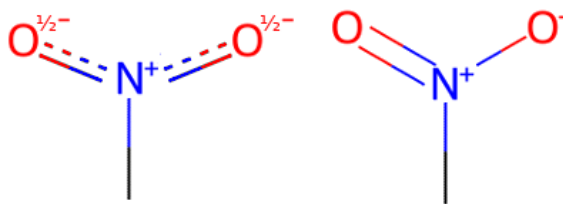



FIGURE 1.7 – Représentations possibles de la molécule de nitrométhane sous forme de graphe moléculaire. À gauche, représentation basée sur des liaisons de type  $\frac{3}{2}$  et des charges de type 1 et  $\frac{1}{2}$ . À droite, représentation alternative basée sur des charges entières uniquement. La représentation de gauche est adaptée de Wikimedia (Zirguezi) .

et les trois autres sont partagés avec deux atomes d'oxygène. Dans la nomenclature chimique, ce groupement d'atomes est nommé « groupe nitro ». La façon la plus correcte de représenter cette structure serait de considérer des liaisons de type  $\frac{3}{2}$  et des charges de type  $\frac{1}{2}$ . La Figure 1.7 présente en partie gauche une représentation sous forme de dessin moléculaire de la molécule de nitrométhane en considérant ces liaisons et charges non entières. Les liaisons de type  $\frac{3}{2}$  sont représentées à l'aide d'une ligne pleine et de tirets. Pour représenter cette structure sous forme de graphe moléculaire, il est nécessaire de considérer un type de liaison spécifique non entier et la possibilité de considérer des charges non entières. Bien que cela soit possible, nous choisissons de ne pas adopter ce genre de liaisons et charges dans nos travaux car cela complexifierait grandement le traitement automatique des molécules. Pour qu'une structure moléculaire utilisant ces liaisons et charges soit valide, elles doivent nécessairement être présentes par paires afin que la somme des électrons impliqués dans des liaisons covalentes soit un nombre entier. Nous considérons que cette vérification serait trop complexe dans les travaux que nous mènerons au sein de ce mémoire, et en particulier pour la définition d'un voisinage moléculaire valide dans le Chapitre 2. La Figure 1.7 présente en partie droite une façon alternative de représenter la molécule de nitrométhane en l'absence de liaisons et de charges non entières. Cette représentation est moins correcte en chimie théorique car les électrons ne sont pas distribués de façon homogène dans la molécule, mais elle permet de considérer cette fonction chimique au sein d'un graphe moléculaire classique. Cet exemple permet de mettre en évidence une structure limite pour laquelle une représentation basée sur la géométrie (voir la suite de cette section) serait plus pertinente que les graphes moléculaires, bien que cette structure puisse être représentée sous forme de graphe moléculaire.

Selon notre définition, les graphes moléculaires n'encodent pas d'informations relatives au placement géométrique des atomes dans l'espace. Il serait possible de prendre en

compte une partie de cette information, et notamment les informations de stéréochimie qui encodent de façon catégorielle la position relative des atomes dans des zones de la molécule, lorsque cette information est disponible. Dans le cadre des travaux que l'on présente dans ce mémoire, nous ne prenons pas en compte cette information car cela nous permet de travailler sur un modèle plus simple de l'espace moléculaire. Ainsi, il n'existe pas plusieurs versions d'un même graphe moléculaire selon notre définition  $G$ . De plus, cela permet de ne pas traiter le problème de la définition de l'égalité entre des graphes moléculaires possédant ou ne possédant pas d'informations de stéréochimie.

### Représentations sous forme de texte

Une autre représentation très populaire des molécules consiste à représenter un parcours du graphe moléculaire sous une forme textuelle. Cela permet de représenter les molécules sous une forme assez compréhensible à l'œil initié et permettant un archivage efficace. La représentation texte la plus populaire est le SMILES [WEININGER 1988], dont nous représentons un exemple en partie (B) de la Figure 1.6. Le SMILES représente les atomes selon leur symbole, les branchements par des couples de parenthèses et les cycles par des étiquettes numériques qui décrivent la création de liaisons supplémentaires. La succession de deux atomes implique qu'ils partagent une liaison. Il s'agit d'une liaison simple par défaut. Si le symbole « = » est intercalé entre les deux atomes, il s'agit d'une liaison double. Le symbole « # » est utilisé pour représenter les liaisons triples.

Un SMILES correspond à un parcours de graphe moléculaire, et il existe au moins autant de parcours que d'atomes dans un graphe moléculaire. Il est possible que plusieurs parcours différents forment le même SMILES, mais dans le cas général il existe plusieurs SMILES pour décrire une seule molécule. Des travaux ont cependant proposé des algorithmes exploitant les caractéristiques des molécules pour proposer un parcours déterministe [WEININGER et al. 1989]. Cela permet de définir une version dite canonique des SMILES, qui permet de représenter de manière unique toute molécule que l'on peut représenter sous la forme d'un graphe moléculaire. Notons que cela permet de tester simplement l'égalité entre deux graphes moléculaires quelconques, par conversion en SMILES canoniques.

Par définition des SMILES, il est possible et simple de convertir un graphe moléculaire en SMILES et réciproquement, puisque ces représentations sont en fait équivalentes. Cela implique également que les limitations des graphes moléculaires s'appliquent également aux SMILES. Comme pour les graphes moléculaires, nous choisissons au sein de ce

mémoire de ne pas considérer les informations de stéréochimie pour les SMILES.

Les SMILES possèdent une limite potentielle supplémentaire, qui correspond au fait qu'il est difficile de définir une fonction de voisinage permettant de passer d'un SMILES valide à un autre SMILES valide. Nous distinguons deux niveaux de validité. D'abord, le SMILES doit être valide au sens syntaxique, c'est-à-dire qu'il doit pouvoir être converti en un graphe moléculaire lui-même valide. De plus, il doit être valide au sens sémantique, c'est-à-dire qu'il doit correspondre à une molécule dont la valence des différents atomes est correctement respectée. Des auteurs ont montré que l'utilisation d'une grammaire non contextuelle des SMILES permet de définir une procédure de construction et de voisinage garantissant la validité syntaxique [YOSHIKAWA et al. 2018], mais il n'est à notre connaissance pas établi de pouvoir garantir la validité sémantique. D'autres représentations sous forme de texte ont été définies pour pallier cela. En particulier, les SELFIES permettent de garantir à la fois la validité syntaxique et sémantique des molécules représentées, en s'appuyant sur un décodeur qui ignore dynamiquement des portions de la représentation correspondant à des caractéristiques invalides [KRENN et al. 2020].

## Représentation géométrique

Les molécules sont des objets dans un espace géométrique. Il est très souvent raisonnable de considérer les noyaux comme immobiles en comparaison aux électrons, car ils sont bien plus lourds. Par conséquent, il est sensé de représenter les molécules comme un nuage de noyaux atomiques dans un repère cartésien orthonormé. Chaque atome est décrit par son type et sa position. Dans cette représentation, tous les atomes sont représentés explicitement, y compris les atomes d'hydrogène. En revanche, les liaisons sont seulement sous-entendues par les distances entre les couples d'atomes. La partie (C) de la Figure 1.6 illustre cette représentation.

**Mécanique moléculaire** Les positions relatives des atomes au sein des molécules dépendent des interactions entre les électrons. La représentation sous forme d'un nuage de noyaux atomiques obéit donc à des règles chimiques complexes et ne peut donc pas être spécifiée arbitrairement. Il existe plusieurs approches pour obtenir une représentation géométrique à partir d'une représentation sous forme de graphe moléculaire. Ces approches sont plus ou moins précises et coûteuses. La mécanique moléculaire est une approche très populaire qui permet d'obtenir une estimation de la géométrie à bas coût. Cela peut être considéré comme un calcul heuristique de la géométrie, à partir d'un ensemble de règles



(ou « champ de force ») obtenues par l'étude de distances typiques entre des couples d'atomes. Ces règles sont intégrées au sein d'une procédure d'optimisation continue de la géométrie. Au sein de ce mémoire, nous utilisons le champ de force MMFF94, très populaire dans le domaine de la chimie organique [HALGREN 1996]. Nous utilisons en particulier les implémentations du programme OpenBabel [OLBOYLE et al. 2011] et de la bibliothèque RDKit [TOSCO et al. 2014].

**Calcul géométrique en chimie quantique** La mécanique moléculaire permet une estimation rapide de la géométrie moléculaire, mais cette estimation est insuffisante pour certaines applications. En particulier, dans le domaine de la chimie des matériaux moléculaires, les propriétés d'intérêt sont dépendantes de la densité électronique qui est elle-même très dépendante de la position relative des noyaux atomiques. La densité électronique correspond à une fonction décrivant la probabilité de présence des électrons au sein de la molécule, dans un espace en trois dimensions. Dans ce contexte, il est nécessaire d'effectuer des calculs en chimie quantique qui permettent d'obtenir la géométrie moléculaire, les fonctions mathématiques décrivant chaque électron ainsi que les propriétés d'intérêt. Il existe plusieurs méthodes pour effectuer ces calculs, qui sont nécessairement des approximations de la réalité car la résolution exacte des équations en chimie quantique est insoluble pour des systèmes à plusieurs électrons.

**DFT** Une méthode de chimie quantique très populaire car elle permet d'obtenir un compromis intéressant entre précision des résultats et coût de calcul est la DFT, abréviation de « *density functional theory* ». Le calcul de la géométrie en DFT est une double procédure d'optimisation continue, qui vise à minimiser l'énergie totale de la molécule afin de déterminer une position optimale des noyaux, tout en minimisant l'énergie de répulsion entre les électrons afin de déterminer une fonction de densité électronique optimale. Précisons que ce compromis implique néanmoins des coûts de calculs importants voire très importants. Il est difficile de déterminer avec exactitude la complexité de ces calculs, car les implémentations sont opaques et la question semble peu étudiée. Nous savons toutefois que le coût est dépendant du nombre d'électrons dans la molécule, qui peut augmenter rapidement avec le nombre d'atomes. Dans le cadre de ce mémoire, nous utilisons le programme Gaussian pour effectuer les calculs DFT [FRISCH et al. 2009]. Notons que la DFT peut également être utilisée pour effectuer des calculs supplémentaires, comme le calcul des fréquences ou des états excités. Dans le cadre de ce mémoire, nous

utilisons uniquement la procédure d'optimisation. Celle-ci permet d'obtenir les énergies des électrons et la position des noyaux correspondant à un état d'équilibre.

En termes de paramétrage, nous utilisons de façon systématique la fonctionnelle B3LYP avec la base de calcul 3-21G\*. Précisons qu'il s'agit d'un paramétrage assez économe en coût de calcul. La procédure DFT nécessite une géométrie initiale qui est utilisée comme point de départ de la procédure d'optimisation. Nous utilisons pour cela une procédure d'optimisation en mécanique moléculaire.

**Procédure de calcul de la géométrie moléculaire** La mécanique moléculaire ainsi que la DFT sont des procédures d'optimisation continue qui ne garantissent pas un succès d'exécution. Chaque calcul est donc susceptible d'échouer. Il est possible que la procédure d'optimisation ne converge pas vers un état stable. Il est également possible que la procédure converge vers un état stable mais qui ne correspond pas au graphe moléculaire donné en entrée. Cela correspond à une réorganisation des électrons qui peut causer des changements de liaisons au sein de la molécule, voire même mener à la division de la molécule en plusieurs molécules distinctes. Finalement, il peut arriver pour la DFT que l'exécution soit stoppée subitement en raison de l'exécution d'une instruction interdite par le système d'exploitation.

Pour ces raisons, nous définissons une procédure de vérification du succès des optimisations géométriques. Pour la mécanique moléculaire, nous nous assurons de la convergence du processus, et nous comparons la molécule en sortie à la molécule en entrée. Les deux doivent posséder le même graphe moléculaire. Nous vérifions ainsi que la procédure d'optimisation n'a pas divergé de la molécule initiale pour converger vers une autre molécule. Le programme OpenBabel nous permet de convertir une représentation sous forme de nuage d'atomes en représentation sous forme de graphe moléculaire [OLBOYLE et al. 2011]. Il s'agit d'une procédure assez sensible, car elle consiste à transformer les liaisons de l'espace géométrique continu en catégories discrètes (absence de liaison covalente, liaison simple, liaison double, etc.). Cette discrétisation est encodée selon des règles qui sont nécessairement arbitraires, puisqu'en chimie théorique il existe un continuum entre les différents types de liaisons.

La procédure d'optimisation DFT nécessite d'abord un calcul en mécanique moléculaire, qui permet d'obtenir une géométrie initiale qui sert de point de départ à l'optimisation DFT. La procédure d'optimisation de molécules en DFT commence donc par l'application de la procédure de calcul en mécanique moléculaire et de vérification du ré-

sultat décrite dans le paragraphe précédent. Nous appliquons ensuite les calculs en DFT, puis nous vérifions ensuite qu’ils ont convergé avec succès. Finalement, nous vérifions également que les graphes moléculaires de la molécule obtenue en sortie et de la molécule donnée en entrée sont identiques.

**Coût de calcul** Nous menons une brève étude des coûts de calcul de la mécanique moléculaire et de la DFT. Pour cela, nous extrayons aléatoirement des molécules de deux jeux de données de référence, QM9 et ChEMBL. Ces jeux sont décrits dans la section 1.1.4 de ce chapitre. Nous mesurons les temps de calcul de la mécanique moléculaire selon l’implémentation de OpenBabel ainsi que de RDKit, et les temps de calculs correspondants en DFT. Nous appliquons un filtrage et une transformation de ces deux jeux de données afin qu’ils correspondent à l’espace moléculaire que nous serons amenés à manipuler dans la suite de ce mémoire. Ainsi, nous ne considérons pas dans cette expérience les molécules contenant des charges formelles ainsi que les radicaux. De plus, nous supprimons les informations de stéréochimie des molécules, sans pour autant supprimer les molécules qui les contiennent. Pour rappel, la stéréochimie correspond à un encodage catégoriel des informations de placement géométrique relatif des atomes des molécules.

Afin que les expériences soient comparables, nous retirons de ChEMBL les molécules qui contiennent des types d’atomes lourds qui ne font pas partie de QM9, c’est-à-dire qui ne sont pas compris parmi l’ensemble {C, N, O, F}. Nous étudions les temps de calcul en fonction de la taille maximale des molécules. Nous définissons pour cela un filtre qui ignore les molécules contenant plus de 9 ou 30 atomes lourds avant le tirage aléatoire. Nous effectuons 1000 mesures pour la mécanique moléculaire, 100 mesures pour la DFT lorsque les molécules contiennent jusqu’à 9 atomes lourds et 10 mesures lorsqu’elles en contiennent jusqu’à 30. Les mesures sont effectuées sur le même ensemble de 1000 molécules pour chaque jeu de données filtré selon une taille maximale. Lorsque le nombre de mesures est réduit, les 100 premières ou les 10 premières molécules de l’ensemble de taille 1000 sont sélectionnées.

En Table 1.3, nous reportons les résultats numériques de cette étude. On y observe d’abord que la DFT est effectivement très coûteuse par rapport à la mécanique moléculaire. On estime cette différence de 3 à 4 ordres de grandeur. Les temps de calculs moyens en DFT sont de l’ordre de la minute lorsque les molécules contiennent jusqu’à 9 atomes lourds, et de l’ordre de l’heure lorsqu’elles en contiennent jusqu’à 30. Les calculs en mécanique moléculaire sont tous de l’ordre du dixième de seconde. Les calculs effectués

Jeu de données (taille mol.)	Mécanique moléculaire		DFT
	RDKit	OpenBabel	
QM9 ( $\leq 9$ )	0.06 (820/1000)	0.18 (804/1000)	238.18 (77/100)
ChEMBL ( $\leq 9$ )	0.06 (906/1000)	0.15 (814/1000)	159.60 (86/100)
ChEMBL ( $\leq 30$ )	0.15 (887/1000)	0.30 (458/1000)	3740.65 (05/10)

TABLE 1.3 – Temps moyen en secondes pour calculer la géométrie moléculaire et nombre de succès sur le nombre de calculs, en fonction de la méthode de calcul et des données moléculaires. Un filtrage est appliqué sur la taille des molécules (jusqu'à 9 ou jusqu'à 30 atomes lourds). Les temps reportés prennent seulement en compte les calculs effectués avec succès.

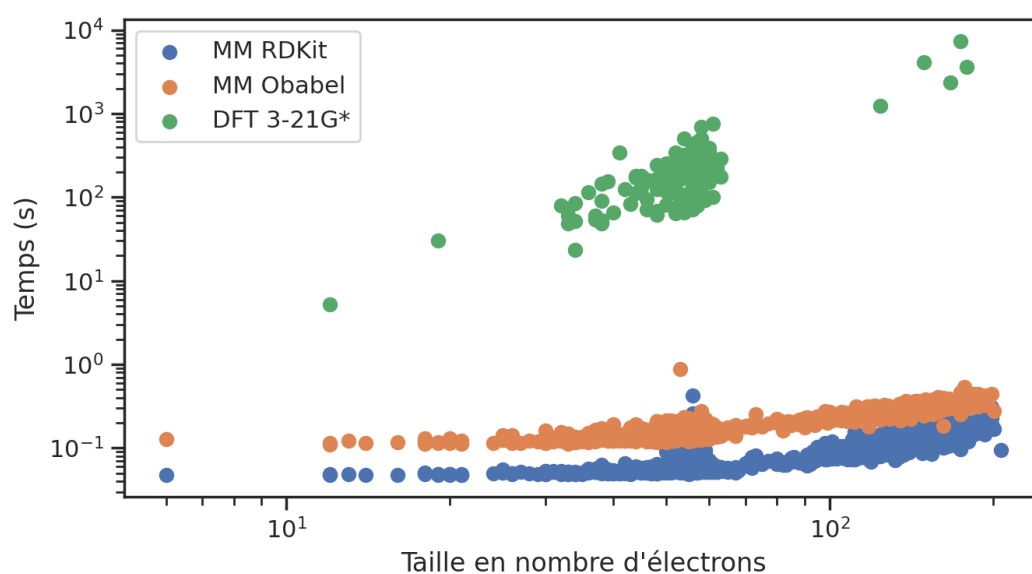


FIGURE 1.8 – Temps de calcul en fonction de la taille des molécules en nombre d'électrons pour différentes méthodes de calcul de la géométrie moléculaire. Seuls les points correspondant à des optimisations qui se sont déroulées avec succès sont reportés.

par RDKit sont systématiquement plus rapides que les calculs effectués par OpenBabel. Concernant le taux de succès, il semble plus élevé sur les données de QM9 que de ChEMBL et il semble évoluer négativement lorsque la taille des molécules augmente, en particulier pour l’implémentation d’OpenBabel.

En Figure 1.8, nous représentons le temps de calcul en fonction de la taille des molécules. La taille des molécules est ici représentée en nombre d’électrons puisque l’on s’attend à ce que le coût de calcul y soit proportionnel. On observe comme attendu que le temps de calcul de la mécanique moléculaire reste raisonnable (inférieur à la seconde) lorsque la taille des molécules augmente jusqu’à contenir environ 200 électrons, et ce pour les deux méthodes de calcul (RDKit et OpenBabel). Les calculs en DFT au contraire nécessitent un coût de calcul supérieur à la dizaine de secondes même pour les molécules de petite taille. Ce coût peut atteindre  $10^4$  secondes soit plusieurs heures pour l’évaluation d’une molécule contenant environ 200 électrons. Rappelons que nous utilisons pourtant un paramétrage des calculs en DFT relativement économe. Ces résultats permettent de mettre en évidence graphiquement le coût très important de la DFT, dans l’absolu et en comparaison à la mécanique moléculaire.

### 1.1.3 Descripteurs moléculaires

Pour certaines tâches, les représentations moléculaires que nous avons présentées dans la section précédente manquent de pertinence. C’est le cas en particulier pour la description des molécules en entrée des modèles d’apprentissage artificiel. La raison principale est que les représentations moléculaires telles que le graphe moléculaire ou le nuage de noyaux ne sont pas invariantes à l’ordre des atomes. Le nuage de noyaux n’est pas non plus invariant à la translation ni à la rotation géométrique. Des descripteurs moléculaires ont été proposés afin d’introduire ces invariances, et également éventuellement de mettre en évidence des caractéristiques moléculaires pertinentes pour les problèmes d’apprentissage.

Nous faisons une distinction explicite entre les représentations et les descripteurs moléculaires. Cette distinction est illustrée en Figure 1.9. Les molécules correspondent à des objets du monde réel qui ne peuvent pas être traités informatiquement sans l’intermédiaire d’une représentation moléculaire. Nous avons présenté un ensemble de représentations et montré que chaque représentation définit en réalité un sous-ensemble de l’espace moléculaire défini par les caractéristiques qui peuvent être représentées. Un descripteur est une fonction  $f_{desc}$  transformant la représentation d’une molécule en un point de  $\mathbb{R}^d$ ,  $d \in \mathbb{Z}_*^+$ . Cette fonction est généralement définie pour posséder des invariances et pour mettre en

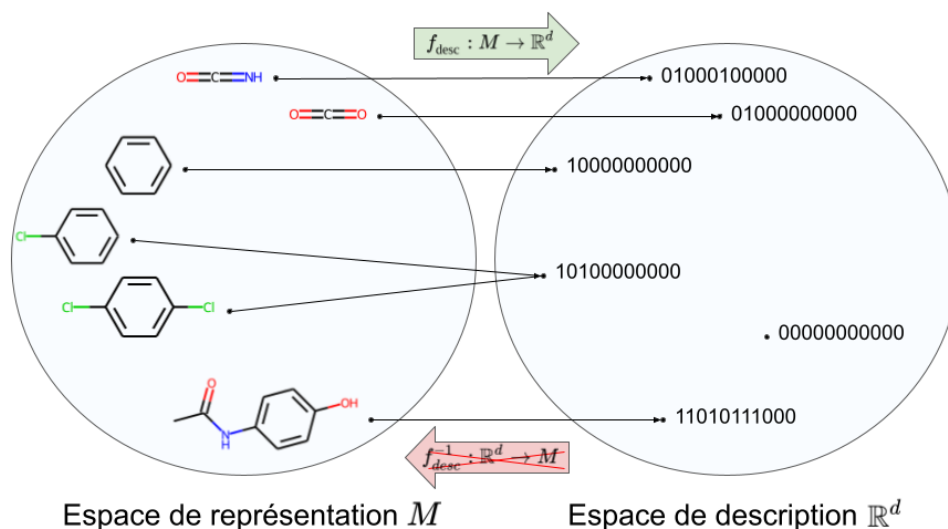


FIGURE 1.9 – Représentation schématique d'un espace  $M$  de représentation de l'espace moléculaire sous forme de graphes moléculaires, et d'un espace de descripteurs inclus dans  $\mathbb{R}^d$ ,  $d \in \mathbb{Z}_*^+$ . Le descripteur moléculaire est défini par une fonction  $f_{desc}$  qui permet de convertir toute molécule de  $M$ . Cette fonction n'admet pas de réciproque.

évidence des caractéristiques moléculaires pertinentes dans un contexte spécifique. Cependant, cette fonction n'admet pas de réciproque et est susceptible de proposer une description unique pour plusieurs molécules différentes.

Les descripteurs moléculaires sont principalement utilisés en entrée des modèles d'apprentissage artificiel, que nous présentons en section 1.2 de ce mémoire. Une autre application potentielle des descripteurs moléculaires est la définition d'une mesure de distance moléculaire. Comme pour les représentations moléculaires, il existe des descripteurs basés sur le graphe ou sur la géométrie moléculaire. Nous présentons d'abord deux descripteurs basés sur le graphe puis un descripteur basé sur la géométrie. Finalement, nous présentons une mesure de la distance moléculaire qui est basée sur un descripteur moléculaire.

### Descripteurs basés sur le graphe moléculaire

**Shingles** Une façon directe de définir un descripteur moléculaire consiste à extraire des caractéristiques génériques des graphes moléculaires, notamment sous la forme de sous-graphes locaux. Cette façon de considérer les molécules comme un assemblage de « briques » locales est assez intuitive pour les chimistes, car elle s'apparente à la définition des groupes (ou fonctions) chimiques. Ces derniers correspondent à des structures

locales qui sont connues pour leurs propriétés (par exemple les groupes alcool constitués d’un atome d’oxygène lié par des liaisons simples à un atome de carbone et un atome d’hydrogène). Parmi cette famille de descripteurs, nous pouvons mentionner les *shingles* [PROBST et REYMOND 2018]. Ils sont associés à un paramètre  $r$  définissant leur rayon maximal. Un *shingle* correspond à un sous-graphe moléculaire de rayon strictement positif inférieur ou égal à  $r$  centré sur un atome d’une molécule. Le rayon est exprimé en nombre de liaisons covalentes. Un *shingle* de rayon 1 centré sur un atome contient l’atome central ainsi que l’ensemble des liaisons et des atomes liés directement à l’atome central. Si l’on considère un rayon 2, deux *shingles* peuvent être définis à partir de l’atome central, à savoir le sous-graphe de rayon 1 et le sous-graphe contenant l’ensemble des atomes et des liaisons à distance de 2 liaisons et moins de l’atome central. Précisons que les *shingles* utilisent un type spécifique supplémentaire pour décrire les liaisons contenues dans des cycles aromatiques. En partie centrale de la Figure 1.10, nous proposons une représentation des *shingles* obtenus avec  $r = 1$  pour un graphe moléculaire relativement simple. Les *shingles* correspondent à des sous-graphes moléculaires qui ne sont pas nécessairement valides selon les règles de valence. Ils sont extraits en supprimant des atomes et des liaisons du graphe moléculaire complet. Les liaisons explicites supprimées ne sont pas remplacées par des liaisons implicites avec des atomes d’hydrogène puisque cela reviendrait à représenter des structures moléculaires différentes. À titre d’exemple, on observe sur la Figure 1.10 que les atomes d’azote du 2<sup>ÈME</sup> et du 5<sup>ÈME</sup> *shingle* représentés possèdent une liaison simple avec un atome de carbone et une unique liaison avec un atome d’hydrogène alors que la valence des atomes d’azote est 3. Ce phénomène existe également pour les autres *shingles* représentés mais n’est pas visible car il concerne des atomes de carbone pour lesquels les liaisons avec des atomes d’hydrogène ne sont pas représentées dans les dessins moléculaires.

Afin de définir un descripteur moléculaire sous forme de vecteur, il est possible de définir un vecteur binaire représentant la présence ou l’absence d’un ensemble de *shingles*. Une alternative est de définir un vecteur entier représentant le nombre d’occurrences d’un ensemble de *shingles*. Dans les deux cas, la dimension du descripteur est dépendante du nombre de caractéristiques considérées. Sauf par énumération des *shingles*, il n’en existe pas un ordre prédéfini. Par conséquent, nous choisissons de construire le descripteur dynamiquement selon les graphes moléculaires préalablement rencontrés et décrits. À chaque fois qu’une nouvelle caractéristique est rencontrée, elle est représentée au premier indice non utilisé dans le descripteur. La dimension du descripteur est choisie au préalable et

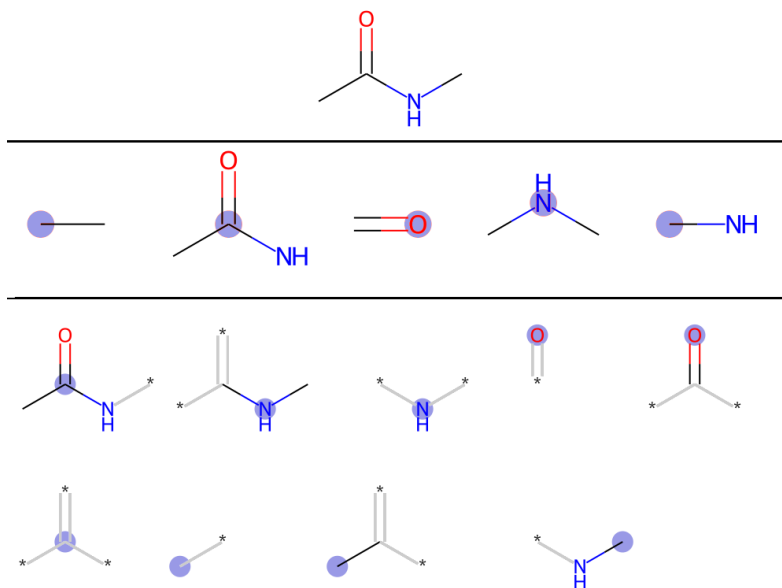


FIGURE 1.10 – Partie supérieure : molécule de N-méthylacetamide. Partie centrale : représentation des caractéristiques correspondantes des *shingles* de paramètre de rayon 1. Partie inférieure : caractéristiques correspondantes ECFP de paramètre de rayon 1 (ECFP2). Les atomes centraux des descripteurs sont mis en évidence par une pastille bleue.

doit être suffisamment grande pour encoder toutes les molécules qui devront être décrites.

**Caractéristiques ECFP** Il existe un autre descripteur également basé sur le graphe moléculaire, qui peut être considéré comme une extension plus spécifique des *shingles*. Il s'agit des caractéristiques ECFP, qui sont très similaires aux *shingles*, mais qui encodent également les types des liaisons des atomes aux extrémités du sous-graphe moléculaire [ROGERS et HAHN 2010]. ECFP correspond à l'acronyme de « *Extended Connectivity FingerPrint* », que l'on pourrait traduire par « empreinte moléculaire de connectivité étendue ». On peut considérer que ces caractéristiques encodent en réalité l'information contenue dans un rayon de  $r + \frac{1}{2}$  par rapport à l'atome central. Comme les *shingles*, les caractéristiques ECFP sont construites pour tous les rayons strictement positifs inférieurs au paramètre de rayon maximal, et elles définissent un type spécifique pour les liaisons comprises dans des cycles aromatiques. Contrairement aux *shingles* en revanche, elles sont également construites pour le descripteur de rayon nul, qui prend en considération l'atome central et les types des liaisons qu'il forme. On accole généralement l'acronyme ECFP avec



le diamètre maximal des caractéristiques considérées. Ainsi, les caractéristiques ECFP de rayon 2 sont généralement nommées caractéristiques ECFP4. En partie inférieure de la Figure 1.10, nous représentons l’ensemble des caractéristiques ECFP2 (de rayon maximal 1) pour une molécule relativement simple. Le symbole « \* » représente un type d’atome quelconque. Les liaisons représentées en gris clair correspondent à des liaisons entre une extrémité du fragment représenté et un type d’atome quelconque. Pour un rayon donné, les caractéristiques ECFP sont plus spécifiques que les *shingles*, puisqu’elles encodent les types de liaisons formées aux extrémités du sous-graphe. À titre d’exemple, nous pouvons comparer le 3<sup>ÈME</sup> *shingle* avec la 5<sup>ÈME</sup> caractéristique ECFP de la Figure 1.10. Cette dernière est plus spécifique car elle encode que l’atome de carbone forme deux liaisons simples avec des atomes quelconques. En revanche, la 4<sup>ÈME</sup> caractéristique ECFP de rayon plus faible (nul) est plus générique car elle encode tout atome d’oxygène formant une liaison double avec un atome quelconque.

**Définition d’une mesure de similarité moléculaire** Dans plusieurs travaux, les caractéristiques ECFP sont utilisées comme base à la définition d’une mesure de similarité entre deux molécules [BROWN et al. 2019 ; OLIVECRONA et al. 2017]. Cette mesure est formée à partir de l’indice de Jaccard [JACCARD 1901], également appelé similarité de Tanimoto dans la littérature, et est définie de la façon suivante.

$$J(x_1, x_2) = \frac{|x_1 \cap x_2|}{|x_1 \cup x_2|} \quad (1.1)$$

$x_1$  et  $x_2$  correspondent ici à l’ensemble de caractéristiques ECFP de deux molécules dont on souhaite évaluer la similarité. La valeur de  $J(x_1, x_2)$  est un nombre réel nécessairement compris entre 0 et 1. Il est possible de définir également une mesure de distance à partir de  $J$ , qui vaut  $1 - J(x_1, x_2)$ . On parle souvent dans ce cas de distance de Tanimoto. Dans nos travaux, nous utiliserons également cette mesure de similarité ou de distance moléculaire car il s’agit de la mesure de référence dans la littérature. Notons cependant que le choix des caractéristiques ECFP est arbitraire et que d’autres descripteurs tels que le vecteur binaire de *shingles* auraient pu être utilisés.

## Descripteurs basés sur la géométrie moléculaire

Il existe également des descripteurs moléculaires qui encodent l’information géométrique des molécules, sous une forme invariante à l’ordre des atomes et à la translation et

rotation géométriques. Parmi ces descripteurs, nous pouvons en particulier citer la matrice de Coulomb, SOAP (acronyme de « *smooth overlap of atomic positions* ») et MBTR (acronyme de « *many-body tensor representation* »).

La matrice de Coulomb encode notamment l'énergie de répulsion entre les paires d'atomes, qui dépend des numéros atomiques et de la distance euclidienne [RUPP et al. 2012]. SOAP permet d'encoder des environnements locaux centrés sur chaque atome de la molécule. La densité électronique des atomes est modélisée par des fonctions gaussiennes sur lesquelles des transformations sont appliquées afin d'obtenir un descripteur invariant à la rotation [BARTÓK et al. 2013].

**MBTR** Dans le cadre de ce mémoire, nous allons utiliser en particulier le descripteur MBTR, que l'on décrit ici plus en détail. MBTR est un descripteur qui encode les caractéristiques géométriques sous une forme invariante à l'ordre des atomes, ainsi qu'à la translation et rotation géométriques. Ce descripteur a été proposé par [HUO et RUPP 2018], qui montrent qu'il permet d'obtenir de meilleurs résultats que la matrice de Coulomb et SOAP dans le cadre de la prédiction de propriétés électroniques. MBTR est obtenu à partir d'un ensemble de fonctions  $g_k$ , qui calculent une valeur réelle à partir de  $k$  atomes de la molécule. Nous suivons la formalisation et l'implémentation proposées par [HIMANEN et al. 2020]. Soit  $a, b$  et  $c$  trois atomes d'une molécule  $x$ .  $g_1(a)$  correspond au numéro atomique de  $a$ ,  $g_2(a, b)$  correspond à l'inverse de la distance euclidienne entre  $a$  et  $b$  et  $g_3(a, b, c)$  correspond au cosinus de l'angle formé par les atomes  $a, b$  et  $c$ . Ces trois fonctions sont utilisées pour définir des distributions gaussiennes  $D_1(a)$ ,  $D_2(a, b)$  et  $D_3(a, b, c)$ , centrées respectivement sur  $g_1(a)$ ,  $g_2(a, b)$  et  $g_3(a, b, c)$ . Le descripteur MBTR est ensuite construit à partir de distributions que l'on définit de la façon suivante.

$$\begin{aligned} \text{MBTR}_1^{Z_1}(x) &= \sum_a^{|Z_1|} D_1(a) \\ \text{MBTR}_2^{Z_1, Z_2}(x) &= \sum_a^{|Z_1|} \sum_b^{|Z_2|} w_2^{a,b} D_2(a, b) \\ \text{MBTR}_3^{Z_1, Z_2, Z_3}(x) &= \sum_a^{|Z_1|} \sum_b^{|Z_2|} \sum_c^{|Z_3|} w_3^{a,b,c} D_3(a, b, c) \end{aligned} \quad (1.2)$$

$\text{MBTR}_1^{Z_1}$  correspond à la somme des distributions  $D_1$  obtenues pour tous les atomes de numéro atomique  $Z_1$  dans la molécule  $x$ . De façon semblable, on peut construire la

	Molécules				Taille molécules		
	Total	{C, N, O, F}	Charges	Radicaux	Min.	Med.	Max.
QM9	133 885	133 885	1 845	0	1	9	9
ChEMBL	1 817 795	922 494	128 658	432	1	27	139

TABLE 1.4 – Description et statistiques des jeux de données QM9 et ChEMBL. La première partie du tableau correspond au comptage des molécules possédant différentes caractéristiques. Les colonnes représentent respectivement le nombre total de molécules, le nombre de molécules contenant uniquement des atomes lourds parmi {C, N, O, F}, le nombre de molécules contenant une ou plusieurs charges formelles, et le nombre de radicaux. La seconde partie du tableau représente la valeur minimale, médiane et maximale des tailles de molécules en nombre d’atomes lourds.

distribution  $\text{MBTR}_2^{Z_1, Z_2}(x)$  pour les couples de numéros atomiques  $Z_1$  et  $Z_2$  ainsi que la distribution  $\text{MBTR}_3^{Z_1, Z_2, Z_3}(x)$  pour les triplets de numéros atomiques  $Z_1$ ,  $Z_2$  et  $Z_3$ . Lorsque  $k$  est égal à 2 ou 3, la distribution  $D_k$  est pondérée par un poids  $w_2^{a,b}$  ou  $w_3^{a,b,c}$  qui est inversement proportionnel à la somme des distances entre les atomes considérés. Ce poids est défini dans l’objectif de donner plus d’importance aux caractéristiques décrivant des atomes proches. Chaque distribution pour chaque type d’atome, chaque couple de types d’atomes et chaque triplet de types d’atomes est échantillonnée à des intervalles réguliers. Le descripteur MBTR correspond à la concaténation de tous ces échantillonnages au sein d’un vecteur qui est normalisé de sorte que la norme euclidienne correspondant aux distributions de chaque terme  $k$  est unitaire.

### 1.1.4 Jeux de données en chimie moléculaire

Pour l’étude de modèles d’apprentissage artificiel supervisé de propriétés moléculaires ainsi que pour d’autres applications en chimie-informatique, il est nécessaire d’utiliser des jeux de données moléculaires. Nous présentons dans cette section les deux jeux de données que nous sommes amenés à manipuler en particulier au sein de ce mémoire. Nous mentionnons également plusieurs jeux de données de référence dans la littérature.

**QM9** QM9 est un jeu de données regroupant le résultat de calculs en DFT d’un ensemble d’environ 134 000 molécules de petites tailles [RAMAKRISHNAN et al. 2014]. Les molécules de QM9 sont issues d’un jeu de données synthétique de très grande taille nommé GDB-17, obtenu par énumération partielle de l’espace moléculaire [RUDDIGKEIT et al. 2012]. Cette énumération est soumise à un ensemble de contraintes assez conservatrices dont l’objectif est d’éviter la génération de structures instables. GDB-17 contient environ 166

milliards de molécules composées de 17 atomes lourds ou moins parmi {C, N, O, F, S, Cl, Br, I}. Les molécules de QM9 correspondent à un sous-ensemble de GDB-17, qui est composé de molécules contenant jusqu'à 9 atomes lourds parmi {C, N, O, F}. En Table 1.4, nous reportons un ensemble de caractéristiques du jeu de données QM9. Cela permet de mettre en évidence que QM9 contient relativement peu de molécules contenant des charges formelles, et aucun radical. On y observe également que la majorité des molécules contiennent 9 atomes lourds. Cette observation peut également être effectuée en Figure 1.11, qui représente la distribution des tailles de molécules.

**ChEMBL** ChEMBL est un jeu de données orienté pour la chimie pharmaceutique [GAULTON et al. 2017]. Dans le cadre de ce mémoire, nous considérons la version 25 de ce jeu de données. Contrairement à QM9, ChEMBL ne contient pas de résultats de calculs en DFT. Les molécules de ChEMBL possèdent une activité biologique ou sont suspectées d'en posséder une. Elles sont issues de plusieurs sources, dont principalement la littérature scientifique en chimie du médicament et diverses bases de données de médicaments ou de molécules possédant des activités biologiques. Contrairement à QM9, ChEMBL est composée de molécules dont l'existence est connue. ChEMBL contient près de deux millions de molécules, qui sont pour la plupart de plus grande taille que les molécules de QM9. Les types d'atomes lourds les plus représentés sont C, N, O, F, P, S, Cl et Br. D'autres types d'atomes sont également présents mais de façon marginale (dans plusieurs dizaines ou plusieurs centaines de molécules). En Table 1.4, nous pouvons observer que près d'un million de molécules sont composées d'atomes lourds compris exclusivement parmi {C, N, O, F}. Cela montre l'importance de ces atomes au sein de la chimie du médicament et au sein de la chimie organique en général. ChEMBL contient environ 100 000 molécules contenant des charges formelles, et 432 radicaux. L'étude des tailles des molécules (partie droite de la Table 1.4 et Figure 1.11) montre que ChEMBL contient des molécules de très grandes tailles, et que la distribution est assez large et centrée près de 30 atomes lourds.

**Autres jeux de données** Il existe également d'autres jeux de données de référence en chimie, que nous ne serons pas amenés à manipuler directement au sein de ce mémoire. Nous pouvons notamment mentionner la base de données PubChem, qui contient plus de 100 millions de molécules issues notamment de bases de données de molécules disponibles à l'achat et de publications scientifiques [KIM et al. 2021]. Dans des travaux antérieurs, notre équipe de recherche a proposé un jeu de données nommé PC9 qui correspond à

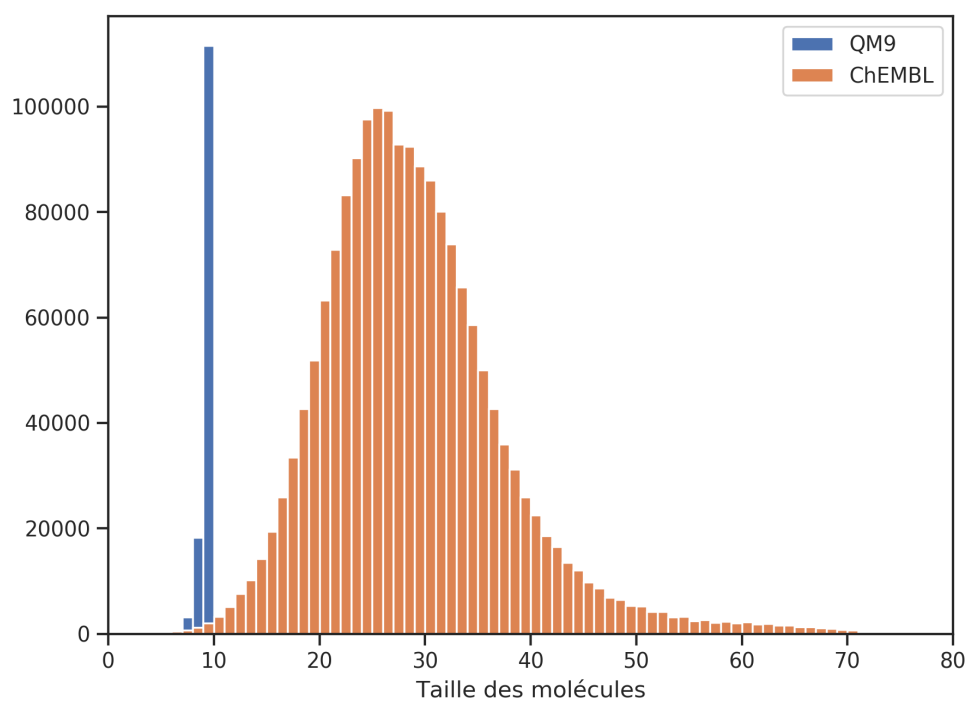


FIGURE 1.11 – Distribution des tailles de molécules en nombre d'atomes lourds au sein de QM9 et ChEMBL. La distribution est tronquée à partir de 80 atomes lourds, bien qu'il existe des molécules de plus grandes tailles au sein de ChEMBL.

l'ensemble des 99 234 molécules de PubChem contenant jusqu'à 9 atomes lourds parmi {C, N, O, F} [GLAVATSKIKH et al. 2019]. PC9 et QM9 forment deux sous-ensembles de l'espace des molécules respectant cette contrainte de nombre et de type d'atomes lourds. Nous pouvons finalement mentionner la base de données ZINC, qui contient plus d'un milliard de molécules disponibles dans différents catalogues commerciaux [IRWIN et al. 2020].

## 1.2 Apprentissage artificiel pour la chimie moléculaire

L'apprentissage artificiel et notamment l'apprentissage artificiel supervisé est le champ d'étude des méthodes permettant d'« apprendre » à estimer les valeurs d'une fonction quelconque à partir d'observations de ses valeurs. En chimie et notamment en chimie des matériaux moléculaires, l'estimation théorique des propriétés d'intérêt peut dépendre de calculs très coûteux. Ce coût est une motivation importante à la recherche de modèles d'apprentissage artificiel efficaces pour la prédiction des valeurs de propriétés de molécules quelconques.

Le domaine de l'apprentissage artificiel est un domaine très large étudié depuis de nombreuses années, au sein duquel des algorithmes très variés ont été proposés [CORNUÉJOLS et al. 2018]. Dans cette section, nous présentons d'abord les concepts élémentaires de ce domaine. Ensuite, nous présentons un ensemble d'algorithmes d'apprentissage artificiel que nous serons amenés à utiliser directement au sein de nos travaux, ou qui sont utilisés dans des travaux de référence de notre domaine de recherche. Finalement, nous évoquons un ensemble de modèles d'apprentissage artificiel définis pour la prédiction de propriétés moléculaires.

### 1.2.1 Concepts élémentaires

Le domaine de l'apprentissage artificiel se scinde principalement entre le champ de l'apprentissage supervisé, le champ de l'apprentissage non supervisé et le champ de l'apprentissage par renforcement. L'apprentissage supervisé consiste à apprendre un modèle d'une fonction  $f$  quelconque, à partir de données étiquetées selon les valeurs de  $f$ . Dans le cas général, il n'existe pas d'expression analytique de  $f$  ou cette expression n'est pas connue. Cela justifie la recherche d'un modèle de prédiction des valeurs de  $f$ . L'appren-

tissage non supervisé consiste à apprendre des relations au sein d'un ensemble de données non étiquetées. L'apprentissage par renforcement consiste à apprendre un comportement permettant de résoudre un problème donné. Dans le cadre des travaux que nous mènerons au sein de ce mémoire, nous serons uniquement amenés à manipuler des modèles d'apprentissage supervisé.

**Régression et classification** En fonction des caractéristiques de la fonction dont on souhaite apprendre à prédire les valeurs, différents types d'approches peuvent être utilisées. En premier lieu, on distingue généralement les tâches de régression qui consistent à prédire des valeurs d'une fonction  $f$  dont les valeurs sont numériques et ordonnées, et les tâches de classification qui consistent à prédire les valeurs d'une fonction  $f$  dont l'ensemble de valeurs est discret et fini. Les algorithmes (ou méthodes, ou encore modèles) d'apprentissage artificiel sont généralement définis initialement pour la régression ou pour la classification, bien qu'ils peuvent souvent être adaptés à l'autre type de tâche.

**Apprentissage supervisé** Nous définissons de façon générique un algorithme d'apprentissage supervisé comme une fonction  $g : E \rightarrow F$ .  $F$  correspond à l'espace des valeurs prédites, qui est un espace réel ou catégoriel, généralement de dimension 1. Le domaine de définition  $E$  est un ensemble de variables (également nommées caractéristiques) qui peuvent être réelles ou catégorielles, et qui permettent de qualifier les points de l'espace de définition de la fonction  $f$  dont les valeurs sont estimées. Précisons que le domaine de définition de  $g$  et  $f$  n'est pas nécessairement le même. Pour illustrer cela, nous considérons le cas de la prédiction de propriétés moléculaires. Nous nous attendons à ce que le domaine de définition  $E$  de  $f$  soit l'espace moléculaire, puisque  $f$  représente une propriété des molécules. En pratique cependant,  $f$  est une expression formelle ou une simulation informatique qui évalue une représentation des molécules, et son domaine de définition est donc une représentation moléculaire. Dans certains cas, le domaine  $E$  de  $g$  peut correspondre à une représentation moléculaire si le modèle  $g$  est adapté à cette représentation, mais dans le cas général il s'agit d'un descripteur moléculaire. Les concepts de représentation et descripteur moléculaires sont définis dans les sections 1.1.2 et 1.1.3 de ce chapitre.

**Phases d'entraînement et de prédiction** De façon générale, l'utilisation de modèles d'apprentissage supervisé est constituée de deux phases génériques distinctes. La première est la phase d'entraînement, pendant laquelle un modèle  $g$  est construit à partir d'un

ensemble de données étiquetées par  $f$ . Une fois le modèle construit, il peut être utilisé pour prédire les valeurs de nouvelles données qui a priori ne sont pas étiquetées. Il s'agit de la phase de prédiction. Pour la plupart des modèles d'apprentissage artificiel, les données d'entraînement sont considérées uniquement pendant la phase d'entraînement et ne sont donc plus nécessaires pour la phase de prédiction.

**Paramètres et hyper-paramètres** Nous faisons la différence entre les paramètres du modèle d'apprentissage artificiel, qui correspondent à la modélisation de la fonction  $f$  par le modèle, et les hyper-paramètres qui sont définis a priori et qui règlent la façon dont les paramètres sont calculés pendant la phase d'entraînement. À titre d'exemple, un modèle de régression linéaire simple  $g(x) = w_0 + w_1x_1 + \dots + w_nx_n$  dépend uniquement d'un vecteur de paramètres  $\theta = (w_0, \dots, w_n)$ . Des modèles plus complexes comme les réseaux de neurones que nous présentons dans la suite de cette section dépendent également d'hyper-paramètres, comme par exemple le choix des fonctions d'activation.

## 1.2.2 Modèles

Dans cette section, nous décrivons le fonctionnement de plusieurs types de modèles d'apprentissage artificiel supervisé que nous serons amenés à utiliser directement ou indirectement au sein de nos travaux. Nous présentons les modèles de régression par processus gaussien ainsi qu'un ensemble d'architectures de modèles d'apprentissage profond qui sont définies notamment pour la génération de données.

### Régression par processus gaussien

**Définition** Les modèles de régression par processus gaussien sont des modèles probabilistes permettant la régression [RASMUSSEN et WILLIAMS 2006]. Ces modèles sont souvent nommés par l'acronyme GPR, qui correspond à l'abréviation de « *Gaussian process regression* ». Ils sont basés sur la notion de processus gaussien, qui correspond à une collection possiblement infinie de variables de probabilité gaussiennes. Dans le cadre des modèles GPR, ces variables permettent de modéliser la valeur de  $f$  pour chaque point de l'espace de description. Un processus gaussien est défini par une fonction de covariance (ou fonction noyau)  $k(x, x')$  et une fonction moyenne  $m(x)$ . La régression par processus gaussien consiste à conditionner un processus gaussien selon le vecteur de données observées  $X$  avec comme valeurs de fonction cible  $y = f(X)$ , pour obtenir un processus gaussien pos-



térieur qui peut estimer la distribution multivariée  $f_*$  d’un ensemble de nouveaux points  $X_*$ . Cette distribution peut être exprimée de la façon suivante.

$$f_*|X, y, X_* \sim \mathcal{N}(\bar{f}_*, \Sigma_{f_*}) \quad (1.3)$$

Les paramètres de la distribution  $f_*$  ( $\bar{f}_*$  et  $\Sigma_{f_*}$ ) peuvent être calculés selon l’équation (1.4). Pour tous les ensembles  $Y$  et  $Y'$  contenant respectivement  $n$  et  $n'$  éléments de  $\mathbb{R}^d$ , la notation  $K(Y, Y')$  représente la matrice de taille  $n \times n'$  obtenue par le calcul de la fonction  $k$  sur toutes les paires d’éléments de  $Y$  et  $Y'$ .  $\bar{f}_*$  correspond à la moyenne de la distribution prédite pour chaque variable aléatoire et donc à la valeur prédite pour chaque élément de  $X_*$ . La variance des distributions prédites peut être extraite de la diagonale de  $\Sigma_{f_*}$ . S’il existe un bruit gaussien non nul dans les données, ce bruit peut être considéré sous la forme d’un hyper-paramètre  $\sigma_n^2$  qui modélise sa variance. Ce paramètre est inséré dans le terme  $K_n = K(X, X) + \sigma_n^2 I$ .  $I$  correspond ici à la matrice identité. Notons que ces calculs dépendent de l’inversion de la matrice  $K_n$  ainsi que de produits matriciels. Or, la complexité algorithmique de ces opérations est cubique. Par conséquent, l’utilisation d’un modèle GPR peut être relativement coûteuse et n’est pas compatible avec des jeux de données de grandes tailles.

$$\begin{aligned} \bar{f}_* &= K(X_*, X)K_n^{-1}y \\ \Sigma_{f_*} &= K(X_*, X_*) - K(X_*, X)K_n^{-1}K(X, X_*) \end{aligned} \quad (1.4)$$

Il est intéressant de remarquer que ces équations ne décrivent pas un modèle de la fonction cible  $f$  permettant de se détacher des données d’entraînement. En réalité, il n’existe pas à proprement parler de phase d’entraînement, puisque la phase de prédiction dépend explicitement des données d’apprentissage  $(X, f(x))$ . On parle dans ce cas de prédiction par interpolation entre les points connus, en opposition à la prédiction par extrapolation lorsqu’un modèle explicite de  $f$  est construit. En pratique, il est toutefois possible d’effectuer un pré-calcul de la matrice  $K(X, X)$ , ce qui peut être apparenté à une phase d’entraînement.

**Fonction noyau et optimisation des hyper-paramètres** La définition des modèles de régression par processus gaussien s’appuie sur une fonction noyau  $k$ . Nous présentons en équation (1.5) deux fonctions susceptibles d’être utilisées en tant que fonction noyau.

La fonction  $k_{\text{DOTPRODUCT}}$  est basée sur l'opérateur  $\cdot$  correspondant au produit scalaire des vecteurs  $x$  et  $x'$ . Cette fonction dépend de deux hyper-paramètres  $\sigma_s^2$  et  $\sigma_0^2$ . La fonction  $k_{\text{RBF}}(x, x')$  est basée sur la fonction exponentielle et sur la distance euclidienne calculée par la fonction  $d$ . Elle dépend de deux hyper-paramètres  $\sigma_s^2$  et  $l$ . RBF correspond à l'abréviation de « *radial basis function* », qui signifie « fonction à base radiale ». Il s'agit d'une fonction  $\varphi$  ayant pour propriété que  $\varphi(x)$  ne dépend que de la distance entre  $x$  et un point de référence  $c$  de l'espace de recherche (ici,  $c = x'$ ). Nous considérons ici uniquement la fonction RBF exponentielle.

$$\begin{aligned} k_{\text{DOTPRODUCT}}(x, x') &= \sigma_s^2(\sigma_0^2 + x \cdot x') \\ k_{\text{RBF}}(x, x') &= \sigma_s^2 \exp\left(-\frac{d(x, x')^2}{2l^2}\right) \end{aligned} \quad (1.5)$$

Une caractéristique intéressante de la méthode de régression par processus gaussien est qu'il est possible d'obtenir dynamiquement une valeur optimale des hyper-paramètres de la fonction noyau, par maximisation d'une fonction de vraisemblance de la modélisation par rapport aux données [RASMUSSEN et WILLIAMS 2006]. L'intérêt est que le modèle résultant ne dépend d'aucun hyper-paramètre qui doit être réglé par l'utilisateur, à l'exception du paramètre  $\sigma_n^2$  modélisant la variance du bruit. Pour modéliser le bruit gaussien des données tout en le considérant comme un hyper-paramètre qui peut être optimisé dynamiquement, il est possible de considérer une valeur nulle ou négligeable pour  $\sigma_n^2$  et de définir un nouveau paramètre de bruit  $\sigma_n'^2$  intégré au sein d'une fonction noyau. Cette fonction est définie en équation (1.6), et peut être associée en combinaison avec une autre fonction noyau, notamment sous la forme d'une somme des deux fonctions.

$$k_{\text{WHITE}}(x, x') = \sigma_n'^2 \text{ si } x = x' \text{ sinon } 0 \quad (1.6)$$

## Apprentissage profond et architectures

Les modèles d'apprentissage profond ont reçu énormément d'attention et sont devenus très populaires dans la dernière décennie [LECUN et al. 2015]. Dans le cadre de ce mémoire, nous ne sommes pas amenés à les utiliser de manière directe. En revanche, de nombreux travaux de l'état de l'art de notre domaine d'application sont basés sur ce genre d'approche, pour la prédiction de propriétés moléculaires ainsi que pour la génération de molécules. Dans les paragraphes suivants, nous décrivons brièvement le fonctionnement

des modèles d'apprentissage profond et nous en présentons les principales architectures.

**Réseau de neurones** Les modèles d'apprentissage profond sont basés sur des couches successives de neurones artificiels (on parle d'un réseau de neurones). Un neurone artificiel est composé d'un ensemble de poids qui paramètre une combinaison linéaire des valeurs numériques en entrée du neurone. La valeur de cette combinaison linéaire est passée à travers une fonction non linéaire dite d'activation, qui définit la sortie numérique du neurone. L'entraînement du réseau de neurones consiste à régler les poids des neurones qui le composent. Il existe des algorithmes permettant le réglage de ces poids par rétro-propagation d'un gradient de l'erreur après le passage d'un ensemble de données d'entraînement dans le réseau.

**Réseau entièrement connecté** Le terme « profond » sous-entend la présence de plusieurs couches de neurones artificiels. Souvent, les neurones sont organisés d'une manière spécifique qui permet la résolution efficace d'un type de problème d'apprentissage en particulier. On utilise généralement le terme d'architecture pour qualifier une organisation générique des neurones. L'architecture la plus simple consiste à organiser les neurones dans des couches successives entièrement connectées (tous les neurones d'une couche sont connectés à tous les neurones de la couche suivante). On parle alors d'un réseau de neurones entièrement connecté.

**Réseaux récurrents** Les réseaux de neurones récurrents ont la particularité de faire passer l'information de manière récurrente dans les neurones artificiels, ce qui leur permet d'être adaptés à des séries de valeurs de taille indéterminée, comme le texte. Le neurone récurrent admet au sein de son entrée sa sortie à l'étape temporelle précédente. Souvent, les réseaux de neurones récurrents sont définis à partir de neurones artificiels spécifiques permettant de stocker de l'information, tels que les cellules LSTM (abréviation de « *long short-term memory* »), proposées par [HOCHREITER et SCHMIDHUBER 1997].

**Auto-encodeur variationnel** L'auto-encodeur est une architecture dont l'objectif est de permettre l'apprentissage d'une représentation compressée des données. Un auto-encodeur est composé d'un réseau de neurones nommé encodeur et d'un réseau de neurones nommé décodeur. L'objectif du réseau complet qui est une concaténation de l'encodeur et de décodeur est d'obtenir en sortie les données données en entrée. Typiquement, la couche de neurones centrale contient moins de neurones que la dimension des données en

entrée, afin de forcer une compression de l'information. Les auto-encodeurs variationnels (ou VAE, abréviation de « *variational autoencoder* ») constituent une variante probabiliste de l'auto-encodeur [KINGMA et WELLING 2014]. Ils permettent l'apprentissage d'une représentation interne qui correspond à un espace latent dont les points peuvent être décodés de façon crédible et sensée par le décodeur. Il s'agit d'une architecture utilisée pour la génération de données.

**Generative adversarial network** L'architecture GAN (abréviation de « *generative adversarial network* ») est une architecture pour la génération de données basée sur un réseau de neurones nommé générateur, et un réseau de neurones nommé discriminateur [GOODFELLOW et al. 2014]. Les deux réseaux sont entraînés simultanément. L'objectif du générateur est de générer des données crédibles, c'est-à-dire appartenant à la même distribution que les données d'entraînement. Le discriminateur est entraîné à différencier les données du jeu d'entraînement des données générées par le générateur, qui est lui-même entraîné à tromper le générateur. Cette architecture permet généralement de générer des données très réalistes, mais est en revanche sensible au « *mode collapse* », ce qui correspond à un état dans lequel le générateur produit des données avec très peu de diversité.

**Réseaux de neurones graphes** Les réseaux de neurones graphes ou réseaux de neurones pour graphes, traduction de « *graph neural networks* » (GNN) sont une famille générique d'architectures permettant de traiter des données sous forme de graphe à l'aide d'un modèle d'apprentissage profond [ZHOU et al. 2020]. La plupart de ces approches sont basées sur l'adaptation des opérateurs de convolution des architectures spécialisées pour le traitement des images à des données graphes, qui sont définies dans un espace non cartésien.

### 1.2.3 Apprentissage de propriétés moléculaires

En raison du coût d'estimation par la DFT des propriétés électroniques qui sont de grande importance en chimie des matériaux moléculaires organiques, des modèles d'apprentissage artificiel ont été proposés dans la littérature pour leur prédiction. [FABER et al. 2017] proposent d'étudier un ensemble de modèles pour la prédiction de propriétés électroniques, dont des variantes de modèles linéaires et des modèles basés sur des réseaux de neurones. Il existe également des travaux basés sur des modèles de régression par processus gaussien [BARTÓK et al. 2017; DERINGER et al. 2021]. Des modèles

d’apprentissage profond ont également été proposés. Nous pouvons notamment mentionner le modèle SchNet [SCHÜTT et al. 2018], basé sur une architecture représentant les atomes selon leurs coordonnées et leur numéro atomique. SchNet intègre un ensemble de transformations permettant de modéliser les interactions entre atomes et d’introduire une invariance à l’ordre des atomes ainsi qu’à la translation et rotation géométriques.

Dans des travaux qui se situent hors du cadre des contributions de cette thèse, notre équipe de recherche a montré que les performances de ces modèles sont susceptibles d’être altérées par un manque de diversité moléculaire dans les données d’entraînement [GLAVATSKIKH et al. 2019]. Cette étude est basée sur le modèle SchNet et sur les jeux de données QM9 et PC9, présentés en section 1.1.4 de ce chapitre. Nous montrons qu’en raison d’un manque de diversité moléculaire au sein de QM9, les performances de SchNet pour la prédiction de propriétés moléculaires de données inconnues sont meilleures lorsqu’il est entraîné sur les données de PC9 que lorsqu’il est entraîné sur les données de QM9. Ces résultats montrent l’importance de la sélection du jeu de données moléculaires de référence lors de l’utilisation de modèles d’apprentissage artificiel pour la chimie moléculaire.

## 1.3 Optimisation et génération moléculaire

De nombreux travaux en chimie-informatique proposent des méthodes basées sur le domaine de l’intelligence artificielle afin de générer des molécules satisfaisant des propriétés moléculaires. Certaines de ces méthodes s’appuient sur des modèles d’apprentissage profond génératifs. D’autres s’appuient sur des méthodes issues du domaine de l’optimisation. Il existe également un certain nombre d’approches combinant des algorithmes provenant des deux domaines. Nous parlerons génériquement de génération moléculaire pour faire référence à l’ensemble de ces approches.

Dans la section précédente, nous avons présenté le domaine de l’apprentissage artificiel et nous avons évoqué plusieurs architectures de modèles d’apprentissage profond permettant la génération de données. Dans cette section, nous présentons d’abord le domaine de l’optimisation et en particulier les domaines de l’optimisation combinatoire et des méta-heuristiques, dont sont issus les algorithmes évolutionnaires. Par la suite, nous effectuons une revue d’un ensemble de méthodes de génération moléculaire, que nous classons dans trois catégories selon la façon dont elles génèrent des molécules. La première est celle des méthodes appliquant un ensemble de perturbations sur une représentation moléculaire explicite. La deuxième est celle des méthodes d’optimisation d’un espace moléculaire appris

par un modèle d'apprentissage profond. La dernière correspond aux méthodes d'apprentissage profond génératif.

### 1.3.1 Optimisation

Un problème d'optimisation est défini par un espace de recherche  $E$  et une fonction objectif  $f_{\text{obj}} : E \rightarrow \mathbb{R}$ . Il consiste à trouver une solution  $x^*$  de  $E$  maximisant ou minimisant  $f_{\text{obj}}$ . Au sein de ce mémoire, nous traiterons principalement des problèmes de maximisation. Ainsi, nous formalisons un problème d'optimisation générique de la façon suivante.

$$x^* = \operatorname{argmax}_{x \in E} f_{\text{obj}}(x) \quad (1.7)$$

Lorsque que cela est nécessaire, un problème de minimisation peut être converti en problème de maximisation de manière très simple. Il suffit pour cela de considérer l'opposé de la fonction objectif, comme dans l'équation suivante.

$$\operatorname{argmin}_{x \in E} f_{\text{obj}}(x) = \operatorname{argmax}_{x \in E} (-f_{\text{obj}}(x)) \quad (1.8)$$

#### Concepts élémentaires

**Optimisation combinatoire et optimisation continue** En fonction des caractéristiques des problèmes d'optimisation, différentes méthodes de résolution peuvent être appliquées. Il existe deux catégories majeures de problèmes qui forment deux champs relativement distincts. Il s'agit d'une part des problèmes dont le domaine de définition ( $E$ ) est continu, et d'autre part des problèmes dont le domaine de définition est discret. On parle respectivement d'optimisation continue et d'optimisation combinatoire. Au sein de ce mémoire, nous traiterons principalement d'optimisation combinatoire puisque nous considérerons l'optimisation de graphes moléculaires, qui sont des objets discrets. Nous évoquerons cependant des méthodes d'optimisation moléculaire continue.

**Optimisation boîte-noire** Certaines méthodes d'optimisation continue exploitent la formulation analytique de la fonction objectif  $f_{\text{obj}}$  lorsque cette dernière est connue et dérivable. La dérivation de  $f_{\text{obj}}$  permet d'obtenir un gradient qui est utilisé afin de déterminer la direction dans l'espace de recherche permettant de maximiser l'objectif. Pour différentes raisons, cette approche peut ne pas être applicable. C'est le cas notamment

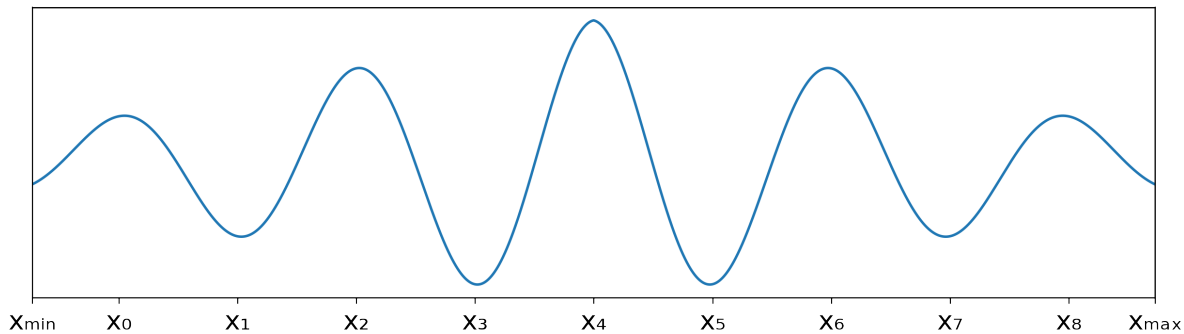


FIGURE 1.12 – Représentation d’une fonction  $f : [x_{\min}, x_{\max}] \rightarrow \mathbb{R}$  symétrique par rapport au point  $x_4$ . L’unique maximum global de  $f$  est la solution  $x_4$ . Les maximums locaux de  $f$  sont les solutions  $x_0, x_2, x_4, x_6$  et  $x_8$ . Les minimums globaux de  $f$  sont les solutions  $x_3$  et  $x_5$ . Les minimums locaux de  $f$  sont les solutions  $x_{\min}, x_1, x_3, x_5, x_7$  et  $x_{\max}$ .

si la formulation de la fonction objectif n’est pas connue, si elle n’est pas dérivable ou également simplement si l’espace de recherche n’est pas continu. On parle alors souvent d’optimisation « boîte-noire », terme qui fait référence au fait que la fonction objectif est considérée comme une boîte-noire qui peut uniquement être évaluée à des points de l’espace de recherche.

**Résolution exacte ou approchée** Certaines approches d’optimisation permettent de résoudre les problèmes de manière exacte, c’est-à-dire qu’elles garantissent que la solution obtenue possède la plus grande valeur possible de  $f_{obj}$  sur l’espace de définition du problème de maximisation et vérifie ainsi strictement l’équation (1.7). On nomme une solution de ce genre un maximum (ou génériquement un optimum) global de  $f_{obj}$ . Si  $f_{obj}$  n’est pas convexe, il est possible qu’il existe plusieurs maximums (ou optimums) dits locaux, c’est-à-dire des solutions ayant localement la valeur la plus élevée de  $f$ . En Figure 1.12, nous faisons apparaître les maximums et minimums locaux et globaux d’une fonction définie sur un sous-ensemble continu de  $\mathbb{R}$ .

La garantie d’optimalité globale des solutions pour une méthode de résolution exacte peut provenir de la résolution analytique de la fonction objectif, de l’énumération de l’ensemble des solutions si cet ensemble est fini ou encore de l’élagage de l’espace de recherche, qui permet de restreindre l’ensemble des solutions devant être évaluées. L’élagage de l’espace de recherche peut avoir lieu s’il est possible d’identifier qu’une solution globale ne peut pas exister dans un sous-ensemble défini de l’espace de recherche. Pour de nom-

breux problèmes d'optimisation, la résolution analytique n'est pas possible et l'espace de recherche est trop important pour être énuméré, même après élagage. On utilise dans ce cas des méthodes d'optimisation approchée. Le but est alors de chercher une solution (a priori un optimum local) possédant une valeur de fonction objectif « satisfaisante ». Il est possible que cette solution soit un optimum global, mais cela ne peut alors pas être prouvé.

**Intensification et exploration** De nombreuses méthodes d'optimisation approchée sont basées sur une notion de recherche par voisinage. Dans ce cadre, l'intensification et l'exploration sont deux notions très importantes qui permettent de décrire des comportements génériques des algorithmes de recherche. L'intensification (ou exploitation) consiste à chercher des solutions améliorantes dans un voisinage très proche d'une bonne solution connue. Ce comportement permet de converger vers des optimaux de la fonction objectif. Cependant, une approche basée exclusivement sur l'intensification est très susceptible de ne pas pouvoir échapper à un optimum local et donc de produire des solutions décevantes à l'échelle de l'espace de définition de la fonction objectif.

L'exploration (ou diversification) est un comportement qui consiste à explorer des zones éloignées de l'espace de recherche. Cela permet d'échapper à un optimum local pour découvrir des zones plus prometteuses de l'espace de recherche, mais une approche purement exploratoire ne permettra a priori pas d'obtenir des optimaux de la fonction objectif. Pour une optimisation efficace, il est donc nécessaire de combiner ces deux concepts. On parle généralement d'un *compromis* qui doit être effectué entre intensification et exploration.

**Optimisation multi-objectif** Certains problèmes d'optimisation peuvent être définis à partir d'un ensemble de fonctions objectif qui doivent être maximisées simultanément. On parle dans ce cas d'optimisation multi-objectif. Dans le cadre de l'optimisation multi-objectif au sens strict, ces fonctions sont considérées indépendamment et il n'existe pas de relation d'ordre entre deux solutions optimales selon deux fonctions objectif différentes. Il est possible de transformer un problème multi-objectif en problème mono-objectif en agrégeant l'ensemble des objectifs au sein d'une fonction objectif unique. Cela nécessite toutefois de pondérer chaque objectif selon son importance et cela crée une relation d'ordre entre l'ensemble des solutions.



## Méthodes d’optimisation

**Optimisation métaheuristique** Les métaheuristiques sont une classe d’algorithmes d’optimisation approchée qui intègrent la plupart du temps des mécanismes permettant d’échapper à des optimums locaux. Il s’agit d’un domaine très large qu’il est difficile de définir simplement de manière universelle. [BOUSSAÏD et al. 2013] remarquent cependant que la plupart des algorithmes métaheuristiques partagent les caractéristiques suivantes : (i) ils sont inspirés par des principes naturels comme l’évolution biologique des espèces, (ii) ils intègrent des processus stochastiques (aléatoires), (iii) ce sont des algorithmes d’optimisation boîte-noire, et (iv) ils dépendent d’hyper-paramètres qui doivent souvent être réglés en fonction du problème devant être résolu.

**Optimisation métaheuristique à base d’individu** La recherche locale est un algorithme métaheuristique simple, qui consiste à explorer itérativement l’espace de recherche à partir d’une solution de départ, d’une fonction de voisinage et d’une heuristique de sélection. Dans la version la plus simple de l’algorithme, on utilise pour la sélection une heuristique de meilleur améliorant. Cela consiste à sélectionner à chaque itération le meilleur améliorant dans le voisinage exhaustif de la solution courante  $x$ , c’est-à-dire à sélectionner la solution du voisinage possédant la plus haute valeur de  $f_{obj}$ , si cette valeur est plus élevée que  $f_{obj}(x)$ . L’heuristique de premier améliorant est très similaire mais ne nécessite pas l’énumération exhaustive du voisinage. Elle consiste simplement à sélectionner le premier voisin évalué comme améliorant. L’algorithme de recherche locale avec sélection du meilleur ou du premier améliorant est communément nommé *hill-climber*. Par définition, l’exécution converge vers un minimum local de la fonction objectif. Il s’agit d’une approche de pure intensification de l’espace de recherche.

Il existe des variations de la recherche locale, qui intègrent des mécanismes permettant d’échapper à des optimums locaux de la fonction objectif. Cela favorise l’obtention de solutions appartenant à de « bons » optimums locaux de la fonction objectif, voire à un optimum global. À titre d’exemple, la recherche taboue est une variation d’un algorithme de *hill-climbing* avec une relaxation des conditions de sélection d’un voisin, et l’ajout d’une liste dite « taboue » contenant les solutions visitées lors des itérations précédentes [GLOVER 1989]. L’heuristique de sélection d’un voisin de  $x$  consiste à sélectionner la meilleure solution du voisinage qui ne fait pas partie de la liste taboue, sans nécessité que cette solution soit un améliorant de  $x$ .

**Optimisation métaheuristique à base de population** Certains algorithmes méta-heuristiques sont basés sur une population de solutions plutôt qu'un unique individu. C'est notamment le cas des algorithmes évolutionnaires qui forment une grande famille d'algorithmes inspirés de la théorie évolutionnaire de Darwin [JONES 1998]. Les algorithmes génétiques sont les plus célèbres des algorithmes évolutionnaires [HOLLAND 1992]. Ils sont définis typiquement pour des problèmes d'optimisation combinatoire. Ils sont basés sur un opérateur de mutation définissant un voisinage local, ainsi que sur un opérateur de recombinaison permettant d'effectuer un *croisement* entre deux individus pour former une (ou classiquement deux) nouvelle(s) solution(s) possédant des caractéristiques issues des deux *parents*. À chaque étape d'optimisation, des *individus* (solutions de la population) sont sélectionnés pour être recombinaisonnés par paires. L'opérateur de mutation est alors appliqué sur les solutions *filles*, qui sont insérées dans la population, typiquement en remplacement de leurs parents. Dans la formulation initiale des algorithmes génétiques, les solutions doivent être encodées selon un vecteur binaire.

L'optimisation par essaim de particules, ou « *particle swarm optimization* » (PSO) en anglais est une autre métaheuristique basée sur une population de solutions [KENNEDY et EBERHART 1995]. Contrairement aux algorithmes génétiques, PSO est défini pour des problèmes d'optimisation dans un espace de recherche réel  $\mathbb{R}^d$ ,  $d \in \mathbb{Z}_*^+$ . La population est composée de solutions (*particules*) qui sont définies par leur position (coordonnées) dans l'espace de recherche. À chaque itération, les particules subissent un déplacement. Pour chaque particule, ce déplacement est calculé en fonction de la meilleure position qu'elle a connue précédemment, de la meilleure solution connue par l'ensemble de l'*essaim* de particules, et d'un facteur aléatoire. L'intuition de cette approche est que lorsqu'une bonne solution est découverte, une proportion importante de l'essaim est guidée vers cette position. Les particules ont toutefois une part d'indépendance qui leur permet de découvrir de nouvelles zones de l'espace de recherche. Ces comportements permettent de définir un compromis entre intensification et exploration de l'espace de recherche.

Nous pouvons également mentionner l'algorithme « *conformational space annealing* », abrégé CSA [JOUNG et al. 2018]. Il s'agit d'un algorithme proche des algorithmes génétiques, qui considère une population de solutions ainsi qu'un opérateur de croisement et une procédure d'optimisation locale telle qu'un algorithme de *hill-climbing*. À chaque étape d'optimisation, les opérateurs de croisement ainsi que la procédure de recherche locale sont utilisés pour générer de nouvelles solutions. Une différence importante avec les algorithmes évolutionnaires classiques est que chaque solution est en réalité représenta-

tive d’un groupe de solutions correspondant à un optimum local de l’espace de recherche. L’algorithme dépend d’une mesure de distance entre deux solutions quelconques et d’une variable  $D_{cut}$  paramétrant une distance minimale pour qu’une solution de l’espace de recherche soit considérée comme appartenant au même groupe qu’une des solutions de la population. Lorsqu’une solution est générée, elle remplace une solution de la population appartenant au même groupe et possédant une moins bonne évaluation par la fonction objectif, si une telle solution existe. Si aucune solution n’appartient au même groupe mais que la solution générée possède une meilleure évaluation que la moins bonne solution de la population, elle la remplace. Sinon, aucun remplacement n’a lieu. Ce mécanisme permet d’assurer une certaine diversité au sein de la population, qui favorise l’exploration de l’espace de recherche en particulier en début de recherche. La valeur du paramètre  $D_{cut}$  diminue progressivement au cours de l’exécution, afin de favoriser l’intensification lorsque des solutions prometteuses ont été trouvées.

**Optimisation à l’aide d’un modèle de substitution** Il est possible d’intégrer un modèle d’apprentissage artificiel au sein d’une procédure d’optimisation boîte-noire, en tant que modèle d’approximation (ou de *substitution*) de la fonction objectif [QUEIPO et al. 2005 ; VU et al. 2017]. Cela est notamment pertinent si la fonction objectif est une fonction boîte-noire possédant un coût de calcul élevé. Ce genre de fonction est typiquement une simulation informatique d’un processus complexe, permettant par exemple la conception d’éléments en aéronautique [VAIDYANATHAN et al. 2003]. Le but est alors de minimiser le coût total de l’optimisation, avec l’idée que la connaissance intégrée dans le modèle de substitution permettra de sélectionner des solutions plus pertinentes et de limiter les appels à la fonction objectif coûteuse.

Dans ce cadre, les procédures d’optimisation et d’apprentissage sont effectuées conjointement au sein d’une procédure itérative. Le modèle de substitution permet de sélectionner des solutions prometteuses à faible coût. La valeur de fonction objectif de ces solutions est calculée régulièrement et ces données sont intégrées au sein du jeu de données d’entraînement du modèle de substitution. Plusieurs types d’algorithmes d’apprentissage artificiel et d’optimisation peuvent être intégrés au sein de ce cadre général de résolution, nommé optimisation à l’aide d’un modèle de substitution ou optimisation basée sur un modèle de substitution. Nous en effectuons une présentation plus complète au sein du Chapitre 4. Précisons que ces algorithmes sont typiquement définis pour des problèmes d’optimisation continue.

## Évaluation de l'efficacité de la recherche

Lorsque plusieurs méthodes d'optimisation peuvent être définies pour résoudre un problème, il est souvent souhaitable de pouvoir les comparer afin de sélectionner la meilleure approche. La façon la plus directe est de comparer les valeurs de fonction objectif des solutions obtenues. La méthode obtenant les meilleures solutions peut être considérée la meilleure, en particulier si ces performances sont observées sur plusieurs variations (*instances*) d'un problème générique, ainsi que sur plusieurs exécutions indépendantes si les méthodes incluent des composants stochastiques. L'inconvénient de cette approche de comparaison est qu'elle ne prend pas en compte l'efficacité des méthodes, c'est-à-dire le rapport entre coût de calcul et la qualité des solutions obtenues.

Pour prendre en compte l'efficacité de la recherche, il est possible de prendre en compte le temps de calcul. Cela peut être effectué simplement en fixant un budget de temps de calcul. Il est alors possible de comparer la qualité des solutions obtenues pour un budget de temps donné. L'inconvénient est que cela nécessite d'effectuer toutes les expériences dans un environnement de calcul homogène, et que les résultats sont liés à un environnement de calcul en particulier. Cela peut être un obstacle à la reproductibilité des résultats.

Plutôt que de considérer le temps de calcul, il peut être pertinent de quantifier le coût de l'optimisation en fonction du nombre d'appels à la fonction objectif. On peut ainsi évaluer la qualité des solutions en fonction d'un budget d'appels à la fonction objectif, ce qui permet d'obtenir des résultats a priori reproductibles indépendamment de l'environnement de calcul.

Pour compléter et préciser cette mesure, il est possible de mesurer le coût de calcul nécessaire pour l'obtention d'une valeur cible de la fonction objectif. La métrique nommée « *expected runtime* » en anglais (abrégée ERT) définit une mesure de ce coût [HANSEN et al. 2021]. Nous la traduisons par « espérance du coût de l'exécution ». Dans cette traduction, nous choisissons de parler de coût et non de temps de calcul car cette mesure peut de façon générique représenter le coût exprimé sous forme de temps d'exécution ou de nombre d'appels à la fonction objectif. La mesure d'ERT permet de comparer numériquement les performances de plusieurs méthodes sur des valeurs cibles de la fonction objectif plus ou moins élevées. Elle est définie pour agréger les résultats de plusieurs exécutions d'une même méthode. Elle correspond au coût dépensé pour obtenir une solution de valeur de fonction objectif au moins égale à une cible  $c$ , divisé par le nombre d'exécutions ayant effectivement atteint la cible  $c$ . Son calcul peut être formalisé de la manière suivante.

$$\text{ERT}(X, c) = \frac{\sum_{x \in X} \text{coût\_min}(x, c)}{\sum_{x \in X} \text{cible\_atteinte}(x, c)} \quad (1.9)$$

Dans cette équation,  $X$  correspond à l’ensemble d’exécutions devant être évalué et  $c$  correspond à la valeur cible de fonction objectif. La fonction `coût_min` représente le coût dépensé par l’exécution  $x$  pour obtenir la première solution atteignant une valeur de fonction objectif au moins égale à  $c$ . Si  $x$  n’atteint jamais la cible, son coût total est pris en compte. La fonction `cible_atteinte` renvoie 1 si la cible  $c$  a été atteinte par l’exécution  $x$  et 0 sinon. Cette mesure représente le coût total dépensé dans le but d’obtenir la cible, divisé par le nombre de fois que la cible a été atteinte. Si la cible n’est atteinte pour aucune exécution (et seulement dans ce cas), alors le calcul implique une division par zéro et nous considérons que sa valeur n’est pas définie. Si la cible est atteinte pour toutes les exécutions, alors ce calcul est équivalent au coût moyen pour obtenir la cible. Dans les autres cas, la valeur d’ERT agrège au sein d’une unique valeur le coût dépensé ainsi que le nombre de fois que la cible a été atteinte.

Il est important de préciser que la valeur d’ERT est sensible au budget alloué aux exécutions ainsi qu’au nombre d’exécutions indépendantes menées. Considérons un cas de figure dans lequel nous menons  $n$  exécutions indépendantes d’une même expérience stochastique, en utilisant pour chaque exécution un budget de temps ou de nombre d’appels à la fonction objectif  $b$ . Dans le pire cas (hors cas indéfini), la cible a été atteinte par une unique exécution avec un coût correspondant au budget  $b$ . Le numérateur de l’équation (1.9) vaut alors  $n \times b$ , tandis que le dénominateur vaut 1. La mesure d’ERT pour cette expérience a donc pour valeur  $n \times b$ . Or, cette valeur peut varier fortement en fonction de la façon dont  $b$  et  $n$  ont été sélectionnés. Par conséquent, pour que les résultats soient comparables entre plusieurs expériences, il est important que ces paramètres soient identiques.

### 1.3.2 Optimisation moléculaire par perturbation d’une représentation

Pour l’optimisation moléculaire ou plus généralement la génération de molécules à l’aide de méthodes issues du domaine de l’intelligence artificielle, nous distinguons trois catégories de méthodes selon une catégorisation subjective basée sur la façon dont les molécules sont générées. La première de ces catégories correspond aux méthodes basées sur la perturbation d’une représentation explicite des molécules, que nous présentons dans

cette section.

## Algorithmes évolutionnaires

De nombreuses méthodes de cette catégorie sont des algorithmes évolutionnaires, qui sont utilisés pour la génération de molécules depuis la fin des années 1990 [DEVI et al. 2015]. De nombreux algorithmes évolutionnaires pour l'optimisation de propriétés moléculaires ont été proposés ces dernières années. Le plus populaire d'entre eux est GB-GA [JENSEN 2019]. GB-GA est basé sur un ensemble d'opérateurs de mutation appliquant des perturbations locales au graphe moléculaire, ainsi que sur un opérateur de recombinaison construisant une ou plusieurs molécules à partir de sous-graphes moléculaires issus des parents. Les opérateurs de mutation sont associés à une probabilité d'application qui est calculée selon la fréquence d'apparition des caractéristiques correspondantes dans un sous-ensemble de la base de données ZINC [IRWIN et al. 2020]. L'objectif de cette pondération est de favoriser l'apparition de caractéristiques moléculaires réalistes. La principale critique effectuée à l'encontre des algorithmes évolutionnaires pour la chimie est en effet la production de solutions peu réalistes et donc a priori peu synthétisables [SCHNEIDER et al. 2009]. Nous pouvons également mentionner l'algorithme MolFinder [KWON et LEE 2021]. Il s'agit d'un algorithme de type CSA, qui peut être considéré comme un algorithme évolutionnaire mais qui intègre également des mécanismes permettant de contrôler le taux d'exploration au cours de la recherche selon une mesure de la diversité au sein de la population.

Afin de produire des molécules plus susceptibles d'être synthétisées, des auteurs ont proposé des algorithmes basés sur des opérateurs de mutation correspondant à l'ajout de fragments moléculaires connus. Nous pouvons citer notamment l'algorithme CReM [POLISHCHUK 2020]. Ce dernier est basé sur une base de fragments (sous-graphes moléculaires) issus de la base de données ChEMBL [GAULTON et al. 2017]. Chaque fragment est associé avec le *contexte* auquel il est lié, qui correspond également à un sous-graphe moléculaire. Cela résulte en une base associant un ensemble de contextes avec un ensemble de fragments, a priori « interchangeables » pour un contexte donné. CReM n'est pas présenté comme un algorithme évolutionnaire en soi mais définit une procédure d'optimisation basée sur une population et sur l'application aléatoire des opérateurs de mutation qui peut être considérée de la sorte. Nous pouvons également mentionner l'algorithme évolutionnaire LEADD, dont les mutations sont basées sur un ensemble de fragments également issus de ChEMBL [KERSTJENS et DE WINTER 2022].

D’autres algorithmes évolutionnaires pour la génération de molécules sont basés sur une représentation texte des molécules. L’algorithme ChemGE représente les molécules comme un vecteur de nombres entiers, et définit une procédure basée sur une grammaire non contextuelle pour convertir cette représentation en représentation textuelle SMILES [YOSHIKAWA et al. 2018]. Cette procédure permet de garantir la validité syntaxique des SMILES (les solutions correspondent à un parcours d’un graphe moléculaire), mais pas la validité sémantique (les règles de valence ne sont pas nécessairement respectées). Pour pallier ce problème, [NIGAM et al. 2020] proposent un algorithme évolutionnaire basé sur la représentation SELFIES qui garantit la validité sémantique des solutions.

### Apprentissage par renforcement

L’approche basée sur la perturbation d’une représentation peut également être utilisée en dehors du cadre des métaheuristiques. En particulier, plusieurs travaux sont basés sur l’association d’un agent d’apprentissage par renforcement avec un ensemble de perturbations du graphe moléculaire comparables à celles de GB-GA, par exemple. L’apprentissage par renforcement correspond à une classe de méthodes qui considèrent un agent évoluant dans un environnement, avec lequel il interagit à travers l’application d’actions. L’agent obtient à l’issue de ses actions une *récompense* numérique qu’il doit apprendre à maximiser à travers le choix des actions [SUTTON et BARTO 2018]. Dans le cas présent, l’environnement correspond à la représentation explicite des molécules et les actions correspondent à l’ensemble des opérateurs de perturbation. Plusieurs modèles de ce type basés sur des modèles d’apprentissage profond ont été proposés pour la génération de molécules. Nous pouvons mentionner MolDQN [ZHOU et al. 2019] ainsi que les travaux de [ZHANG et al. 2019]. Notons que ces travaux incluent des opérateurs de perturbation destructifs, ce qui implique que les choix effectués ne sont pas nécessairement définitifs. En cela, nous considérons que ces méthodes se rapprochent plus des approches d’optimisation par voisinage que des approches de construction itérative de solutions dont nous présenterons des exemples en section 1.3.4.

### 1.3.3 Optimisation d’un espace moléculaire latent continu

Ces dernières années, de nombreuses méthodes de génération moléculaire ont été proposées en tirant partie des nouvelles possibilités offertes par les modèles d’apprentissage profond. Parmi ces possibilités, nous nous intéressons dans cette section à l’apprentis-

sage d'un espace moléculaire continu rendu possible par les auto-encodeurs variationnels (voir section 1.2), abrégés VAE. L'espace latent des VAE correspond en effet à un espace moléculaire continu, dans lequel des méthodes d'optimisation continue peuvent être appliquées.

[GÓMEZ-BOMBARELLI et al. 2018] ont proposé une approche de ce type. Le VAE est entraîné à l'encodage et au décodage de SMILES issus de QM9 ou ZINC. Les auteurs observent que les molécules du jeu de données d'entraînement et les molécules générées aléatoirement par le VAE possèdent des propriétés similaires. Cela montre que le modèle permet effectivement d'apprendre « un » espace moléculaire ou plutôt un sous-ensemble de l'espace moléculaire correspondant à la distribution des données d'entraînement. Les auteurs associent ce modèle avec un autre modèle d'apprentissage profond dont le rôle est de prédire les valeurs d'une propriété moléculaire. Ce dernier modèle étant dérivable, la propriété peut être optimisée au sein de l'espace latent du VAE. Plus exactement, le modèle de substitution de la propriété est optimisé au sein de cet espace.

Une autre approche nommée MSO est très semblable, à la différence principale que l'espace latent de l'auto-encodeur est optimisé à l'aide d'une méthode d'optimisation par essaim particulaire [WINTER et al. 2019]. Une autre différence est que l'auto-encodeur n'est pas entraîné à reconstruire la représentation moléculaire donnée en entrée mais à convertir une représentation en entrée dans une seconde représentation en sortie. La motivation est que cela devrait favoriser l'apprentissage d'un espace correspondant à une abstraction sensée de l'espace moléculaire.

Ces méthodes d'optimisation basées sur l'apprentissage d'un espace latent continu tirent profit des avancées effectuées dans le domaine de l'apprentissage profond et rendent possible l'utilisation de méthodes d'un pan important du domaine de l'optimisation. On peut considérer l'espace latent comme une projection d'un espace moléculaire discret dans un espace continu possédant de bonnes propriétés dans un contexte d'optimisation. Cependant, il n'existe pas de garantie que la totalité des points de l'espace moléculaire soit représentée dans cet espace, et il est même raisonnable de s'attendre à ce que ce ne soit pas le cas. L'espace latent est finalement dépendant des données d'entraînement qui ont donc une grande importance dans cette approche.

### 1.3.4 Apprentissage d'un générateur moléculaire

Une dernière approche selon notre classification consiste à utiliser un modèle d'apprentissage profond pour générer des molécules, en dehors du cas particulier qui consiste



à effectuer l’optimisation dans l’espace latent appris par le modèle. La plupart de ces approches génèrent les molécules de façon séquentielle, par construction itérative. Le modèle de génération permet alors de sélectionner la prochaine étape de construction pour un état donné de la molécule.

## Représentations

Certaines méthodes de génération sont basées sur une représentation graphe des molécules. [LI et al. 2018] ainsi que [YOU et al. 2018] proposent des méthodes de construction itérative de graphes moléculaires basées sur des réseaux de neurones graphe. Les actions déclenchées après sélection par le modèle d’apprentissage profond correspondent à l’ajout d’atomes et de liaisons. MolGAN est un modèle de type GAN qui génère une représentation probabiliste d’un graphe, dont est tiré un graphe moléculaire [DE CAO et KIPF 2018].

De nombreux modèles sont basés sur des réseaux de neurones récurrents, conçus notamment à partir de cellules LSTM. Ces modèles sont la plupart du temps associés à une représentation des molécules sous forme de SMILES, pour laquelle ils sont très adaptés. Nous pouvons mentionner les travaux de [OLIVECRONA et al. 2017; SEGLER et al. 2018; YUAN et al. 2020], qui correspondent à la définition d’un modèle permettant de générer les molécules séquentiellement, caractère par caractère. Nous pouvons également mentionner ORGAN, un modèle de type GAN basé sur une représentation SMILES [GUIMARAES et al. 2018]. [YANG et al. 2017] proposent ChemTS, une méthode basée sur la technique de *Monte-Carlo tree search* [BROWNE et al. 2012]. Un réseau de neurones récurrent est utilisé au sein de ChemTS pour effectuer la phase de simulation, qui consiste à générer la partie terminale d’un parcours de l’arbre représentant l’ensemble des SMILES. Il est intéressant de préciser que ChemTS a été utilisé dans le cadre d’une application pratique pour la recherche de molécules satisfaisant des propriétés électroniques. Certaines de ces molécules ont été synthétisées pour la confirmation expérimentale des propriétés [SUMITA et al. 2018].

La représentation moléculaire sous forme d’un nuage d’atomes en 3 dimensions est plus complexe et est très rarement employée dans les méthodes de génération. Nous pouvons toutefois mentionner les travaux de [GEBAUER et al. 2019], qui proposent un modèle génératif de construction itérative d’une molécule par placement des atomes dans un espace géométrique.

## Génération de molécules satisfaisant des propriétés moléculaires

Nous avons cité un ensemble de méthodes de génération que nous avons classées selon la représentation moléculaire utilisée, mais nous n'avons pour le moment pas décrit les techniques qu'elles emploient pour générer des molécules satisfaisant une propriété moléculaire donnée.

Un certain nombre de ces méthodes sont basées sur une approche d'apprentissage par renforcement dont l'agent est le modèle génératif lui-même. Des techniques basées sur le gradient pour l'apprentissage d'une *politique* telles que l'algorithme REINFORCE [WILLIAMS 1992] sont employées. Cela concerne entre autres les travaux de [OLIVECRONA et al. 2017] et de [YOU et al. 2018]. Dans ce contexte, la métrique de récompense est définie notamment à partir des valeurs de la propriété cible, parfois dans le cadre d'un compromis avec un second objectif consistant à respecter la distribution d'un jeu de données de référence [OLIVECRONA et al. 2017].

D'autres approches choisissent d'utiliser une approche d'apprentissage par transfert. Dans ce contexte, cela signifie qu'après un entraînement sur un jeu de données généraliste, le modèle génératif est entraîné sur un jeu de données restreint dont les molécules possèdent les propriétés moléculaires recherchées. [YUAN et al. 2020] entraînent ainsi leur modèle génératif sur une base de données pour la chimie pharmaceutique, puis effectuent un second entraînement sur des molécules possédant des propriétés électroniques. [GEBAUER et al. 2019] utilisent une approche similaire, en entraînant leur modèle sur une partie du jeu de données QM9, puis en effectuant un second entraînement correspondant à un « réglage fin » (*fine-tuning*) des paramètres sur un sous-ensemble des données possédant des valeurs précises de propriété cible. [SEGLER et al. 2018] utilisent une méthode très similaire pour la génération de molécules possédant des activités biologiques.

Une autre approche consiste à effectuer un conditionnement de l'apprentissage, ce qui correspond à l'étiquetage des données d'entraînement selon une valeur de propriété cible réelle ou binaire. Cette variable peut permettre de contrôler les caractéristiques des solutions générées a posteriori, sans nécessiter un ré-entraînement. Cette approche est notamment utilisée par [LI et al. 2018].

Comme les méthodes consistant à apprendre une représentation continue de l'espace moléculaire, les méthodes présentées dans cette section tirent profit des avancées récentes dans le domaine de l'apprentissage profond. Elles possèdent toutefois un inconvénient générique, qui est celui de la dépendance à un jeu de données moléculaires. Il est raisonnable d'attendre que cette dépendance crée implicitement un biais vers la génération

de molécules proches du jeu de données de référence. Or, il n'existe pas de jeu de données « universel » représentatif de l'ensemble de l'espace moléculaire. Cela pourrait être un frein à la génération de molécules dans des zones pour le moment inconnues ou peu documentées de l'espace de recherche.

## 1.4 Propriétés moléculaires et réalisme des molécules

Un problème d'optimisation est défini par l'espace de recherche des solutions et par la fonction objectif. Pour l'optimisation de propriétés moléculaires, l'espace de recherche correspond logiquement à l'espace moléculaire, et la fonction objectif est logiquement exprimée à partir d'une propriété moléculaire cible, voire correspond directement à cette propriété. Lorsque l'objectif recherché est la maximisation ou la minimisation d'une propriété, il suffit en effet de définir la fonction objectif comme étant la propriété cible, ou éventuellement son opposé pour inverser son sens.

Dans des cas plus réalistes, les chimistes souhaitent obtenir des molécules possédant des valeurs cibles d'une propriété, ou des valeurs au-delà ou en deçà d'un seuil. Il est dans ce cas possible d'appliquer une transformation à la propriété pour définir la fonction objectif. Une fonction gaussienne centrée sur la cible peut être utilisée si l'objectif correspond à l'obtention de solutions possédant une valeur donnée de propriété. Une fonction sigmoïde permet d'exprimer un seuil sur les valeurs d'une propriété. Dans des cas réalistes, les chimistes souhaitent également que les molécules proposées puissent être synthétisées en pratique.

Dans la suite de cette section, nous nous intéressons d'abord au concept de synthétisabilité des molécules. Nous proposons la notion de réalisme des molécules, qui y est liée. Nous présentons ensuite plusieurs métriques de la littérature définies pour proposer une estimation quantitative de l'accessibilité synthétique des molécules. Finalement, nous présentons plusieurs propriétés moléculaires dont nous serons amenés à étudier l'optimisation au sein de ce mémoire.

### Réalisme et estimation de l'accessibilité synthétique des molécules

**Synthétisabilité et réalisme des molécules** Les utilisateurs des méthodes de génération moléculaire attendent typiquement que les molécules générées soient susceptibles d'être synthétisées expérimentalement, afin de résoudre des problèmes réalistes. Or, la synthétisabilité est une notion dont l'évaluation systématique est très complexe. L'exper-

tise des chimistes leur permet de juger si une molécule semble a priori synthétisable ou si au contraire elle possède des caractéristiques qui la rendent complexe voire impossible à synthétiser. Cependant, cette expertise est difficile à formaliser. Cela s'explique par le fait qu'il s'agit d'une notion liée à une part d'intuition des chimistes, issue de leur expérience. De plus, il est important de préciser que cette notion n'est pas universelle. Elle est en effet dépendante du domaine de la chimie de prédilection du ou de la chimiste qui effectue le jugement. Au sein seulement de la chimie moléculaire organique, les molécules rencontrées en chimie pharmaceutique et en chimie des matériaux moléculaires peuvent être très différentes. Or, il s'agit de domaines relativement hermétiques. Parfois, les chimistes attendent également qu'une telle mesure prenne en compte des éléments liés à leur domaine qui ne sont pas directement liés à la possibilité de synthèse. De plus, la notion de synthétisabilité n'est pas absolue dans le temps. La recherche de routes de synthèse pour des structures qui n'ont jamais été synthétisées fait partie des thèmes de recherche en chimie, et cette connaissance est donc en constante évolution.

Pour ces raisons, nous préférons souvent parler de *réalisme* des molécules au sein de ce mémoire. Il s'agit d'une notion plus qualitative que quantitative et qui selon nous fait mieux apparaître la réalité subjective du concept. Le réalisme des molécules ne représente pas seulement si une molécule est susceptible d'être synthétisée, mais plus généralement s'il est raisonnable d'attendre qu'elle soit stable dans des conditions normales de température et de pression. Ce concept peut également prendre en compte des éléments liés à un domaine de la chimie en particulier.

Malgré la difficulté de formaliser la notion de synthétisabilité, il existe un ensemble de métriques dans la littérature qui sont définies afin d'en proposer une estimation quantitative. L'existence de ces métriques est justifiée par le fait qu'en pratique, il est parfois nécessaire de trier ou de catégoriser des molécules selon une estimation de leur synthétisabilité. Nous présentons plusieurs de ces métriques dans les paragraphes suivants.

**SAScore** La métrique de SAScore a été proposée par [ERTL et SCHUFFENHAUER 2009]. SA est l'abréviation de « *synthetic accessibility* », qui signifie « accessibilité synthétique ». Le SAScore est basé sur l'analyse structurelle d'un ensemble de molécules issues de la base PubChem (voir la section 1.1.4). Le SAScore récompense la présence de fragments fréquemment rencontrés dans la base de référence, et pénalise les fragments rares. Il pénalise également d'autres caractéristiques chimiques connues pour affecter négativement la synthétisabilité. Il s'agit d'une métrique définie telle que la meilleure valeur possible est 1

et la pire est 10. Les auteurs proposent un seuil en deçà duquel une molécule est considérée comme probablement synthétisable selon le SAScore, qu’ils fixent à 6.0. [VORŠILÁK et al. 2020] proposent d’abaisser ce seuil à 4.4.

**CLScore** La métrique de CLScore a été proposée par [BÜHLMANN et REYMOND 2020]. CL est l’abréviation de « *ChEMBL-likeness* », ce qui signifie « ressemblance à ChEMBL ». Il s’agit en effet d’une métrique définie pour évaluer la ressemblance d’une molécule aux molécules appartenant à un sous-ensemble contenant environ 457 000 molécules de la base de données ChEMBL. Ce sous-ensemble est choisi pour contenir des molécules possédant des activités sur des protéines, et donc susceptibles de pouvoir être utilisées en tant que médicaments. Le CLScore est défini à partir de l’étude de la représentation des *shingles* de rayon 3 (voir la section 1.1.3) au sein de ChEMBL. Formellement, il est défini selon l’équation (1.10).  $N$  correspond au nombre de *shingles* au sein de la molécule évaluée.  $m$  correspond au nombre de *shingles* qu’elle partage avec le sous-ensemble de ChEMBL.  $(f_S)_i$  correspond au nombre d’occurrences du *shingle*  $i$  au sein du sous-ensemble de ChEMBL. Notons que seuls les *shingles* apparaissant au moins 100 fois au sein de ChEMBL sont considérés.

$$\text{CLScore} = \frac{\sum_{i=1}^m \log_{10}(f_S)_i}{N} \quad (1.10)$$

Il ne s’agit pas à proprement parler d’une métrique définie pour évaluer la synthétisabilité, mais les auteurs partent de l’hypothèse que les molécules de ChEMBL sont représentatives de la chimie pharmaceutique et donc qu’elles sont synthétisables. De plus, ils montrent qu’il existe une corrélation entre les valeurs du SAScore et les valeurs du CLScore. Les auteurs proposent un seuil des valeurs de CLScore fixé à 3.3. Les valeurs associées aux molécules synthétisables sont supérieures à ce seuil.

**Rétro-synthèse** La rétro-synthèse constitue une façon plus précise et plus complexe d’évaluer la synthétisabilité des molécules. Cela consiste à chercher une route de synthèse, c’est-à-dire une séquence de réactions chimiques permettant d’effectuer la synthèse à partir d’une base de molécules disponibles. Le coût de la rétro-synthèse peut être très élevé car l’association de l’ensemble de réactions et de l’ensemble de molécules forment une très grande combinatoire. [GENHEDEN et al. 2020] proposent un outil de rétro-synthèse nommé AiZynthFinder qui est basé sur une recherche par *Monte-Carlo tree search* [BROWNE et al. 2012] assistée par un réseau de neurones. Cette approche permet de réduire le coût de

calcul nécessaire. Selon les auteurs, il faut toutefois compter 10 secondes en moyenne pour l'évaluation d'une molécule. Dans un contexte d'optimisation moléculaire, cela représente un coût très élevé. Par conséquent, nous n'utiliserons pas d'approche de ce genre au sein de nos travaux.

## Propriétés moléculaires

Nous présentons désormais plusieurs propriétés moléculaires dont nous serons amenés à étudier l'optimisation au sein de ce mémoire. Deux de ces propriétés peuvent être calculées à faible coût. Elles proviennent du domaine de la chimie pharmaceutique bien qu'elles ne correspondent pas tout à fait à des objectifs réalistes pour la découverte de médicaments. Elles sont communément utilisées pour évaluer les performances des méthodes de génération moléculaire. Les deux autres propriétés sont étudiées pour la chimie des matériaux moléculaires, et dépendent de calculs coûteux. Elles ne correspondent pas en soi à des objectifs réalistes, mais peuvent potentiellement être intégrées dans la formulation d'un problème réaliste.

**plogP** Le plogP est une propriété moléculaire définie à partir d'une seconde propriété nommée logP. Le logP correspond à une échelle de solubilité entre l'eau et l'octanol. Une valeur de logP négative correspond à une meilleure solubilité dans l'eau, et une valeur positive correspond au contraire à une meilleure solubilité dans l'octanol. Il s'agit d'une mesure expérimentale qui peut être estimée automatiquement à partir d'une méthode qui prend en compte les contributions atomiques [WILDMAN et CRIPPEN 1999]. Il s'agit d'une mesure importante pour la chimie pharmaceutique, car les médicaments ont typiquement une valeur de logP au sein d'un intervalle précis.

La métrique plogP a été définie dans un contexte d'optimisation moléculaire [GÓMEZ-BOMBARELLI et al. 2016]. plogP signifie « penalized logP », c'est-à-dire « logP pénalisé », et est défini selon l'équation (1.11). Il correspond donc à la maximisation des valeurs de logP, avec une pénalité fixée par le SAScore pour favoriser la génération de molécules synthétisables, et une seconde pénalité pour limiter la présence de cycles de grandes tailles. Cette dernière est définie comme  $\text{penalite\_cycles}(x) = \max(\text{taille\_max}(x) - 6, 0)$ , avec  $\text{taille\_max}(x)$  la taille du cycle de plus grande taille au sein de la molécule  $x$ . Souvent, une variante normalisée du plogP est utilisée. Cette dernière correspond à la valeur de l'équation (1.11) après normalisation de  $\text{logP}(x)$ ,  $\text{SAScore}(x)$  et  $\text{penalite\_cycles}(x)$  telle qu'ils possèdent une moyenne nulle et un écart-type unitaire sur un sous-ensemble de

ZINC contenant 250 000 molécules.

$$\text{plogP}(x) = \log\text{P}(x) - \text{SAScore}(x) - \text{penalité\_cycles}(x) \quad (1.11)$$

**QED** La métrique de QED est issue de la chimie pharmaceutique. QED correspond à l’abréviation de « *quantitative estimate of drug-likeness* », que l’on peut traduire par « estimation quantitative de la ressemblance aux médicaments » [BICKERTON et al. 2012]. Cette métrique est définie à partir de l’étude de 8 propriétés moléculaires typiquement associées aux médicaments, à partir d’un ensemble de médicaments issus de la base de données ChEMBL. Parmi ces propriétés, on retrouve le nombre de cycles aromatiques et la valeur de logP. La valeur de QED est définie entre 0 et 1. Pour correspondre à une valeur de QED élevée, une molécule doit posséder pour les 8 propriétés une valeur au centre de la distribution observée sur les médicaments de ChEMBL. Les auteurs montrent que des médicaments connus possèdent des valeurs de QED élevées. Précisons que cela ne signifie pas qu’une molécule possédant une valeur élevée de QED est un médicament.

**Énergies HOMO et LUMO** Nous étudierons également l’optimisation des valeurs d’énergie HOMO et LUMO. Il s’agit de propriétés étudiées en chimie des matériaux moléculaires organiques, et dont l’estimation dépend nécessairement de calculs en chimie quantique comme la DFT. L’énergie HOMO correspond à l’énergie de l’électron de plus haute énergie au sein de la molécule. L’énergie LUMO est une valeur virtuelle qui correspond à l’énergie d’un électron qui viendrait s’ajouter au sein de la molécule. Nous choisissons ce couple de propriétés car elles sont souvent étudiées de façon concomitante, au sein d’une propriété nommée « *HOMO/LUMO gap* », qui correspond à la différence absolue entre ces deux propriétés. Cette dernière propriété peut permettre de caractériser des problèmes réalistes, notamment pour la recherche d’un matériau moléculaire organique photovoltaïque. Dans le cadre de nos travaux, nous mènerons simplement des optimisations indépendantes des énergies HOMO et LUMO, afin d’évaluer nos approches dans le cadre de la chimie des matériaux moléculaires organiques.

# OPTIMISATION ÉVOLUTIONNAIRE DE PROPRIÉTÉS MOLÉCULAIRES

---

## Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>77</b>
<b>2.2</b>	<b>EvoMol : un algorithme pour l'optimisation de propriétés moléculaires</b>	<b>78</b>
2.2.1	Conception de notre approche	78
2.2.2	Espace de recherche	80
2.2.3	Parcours de l'espace de recherche	83
2.2.4	Algorithme	87
<b>2.3</b>	<b>Évaluation</b>	<b>92</b>
2.3.1	Optimisation de propriétés peu coûteuses	92
2.3.2	Test sur le <i>benchmark</i> GuacaMol	97
2.3.3	Optimisation de propriétés électroniques coûteuses	105
<b>2.4</b>	<b>Génération de solutions réalistes</b>	<b>114</b>
2.4.1	Optimisation des métriques d'accessibilité synthétique	114
2.4.2	Contraintes sur l'espace de recherche	116
2.4.3	Effet des contraintes pour l'optimisation de la QED	118
2.4.4	Effet des contraintes sur le <i>benchmark</i> GuacaMol	126
<b>2.5</b>	<b>Conclusion et perspectives</b>	<b>128</b>

---

Ce chapitre fait l'objet de la publication suivante.

[LEGUY et al. 2020]





Dans ce chapitre, nous proposons et étudions un algorithme évolutionnaire pour l'optimisation de propriétés moléculaires. Notre méthode est conçue pour permettre l'optimisation de propriétés moléculaires variées, dans le cadre très large de la chimie organique. Pour cela, nous prenons soin d'éviter d'intégrer des connaissances trop spécifiques à un domaine de la chimie organique en particulier dans la version de base de notre méthode. Il est cependant possible d'intégrer un biais volontaire afin de spécifier l'espace de recherche accessible selon la volonté et les connaissances de l'utilisateur. Nous montrons que notre approche est performante pour l'optimisation de nombreuses propriétés moléculaires. Nous montrons l'interprétabilité de notre approche à travers une visualisation a posteriori de la recherche sous la forme d'un arbre d'exploration. Finalement, nous étudions plusieurs approches pour favoriser le réalisme des solutions générées.

## 2.1 Introduction

De façon générale, l'objectif des chercheurs en chimie moléculaire consiste à rechercher des molécules qui satisfont des propriétés moléculaires désirées, ainsi qu'un ensemble de contraintes supplémentaires telles que l'accessibilité synthétique ou encore des contraintes de non-toxicité ou de stabilité. La découverte de nouvelles cibles de synthèse est un processus très lent et coûteux. Ces dernières années, de nombreux travaux proposent d'utiliser des méthodes issues du domaine de l'intelligence artificielle afin d'automatiser et d'accélérer une partie de ce processus [LIU et al. 2017; YANG et al. 2019].

Depuis la fin des années 1990, de nombreux travaux sont basés sur des algorithmes évolutionnaires pour ces problèmes de génération moléculaire [DEVI et al. 2015]. Les scientifiques identifient rapidement que cette approche est susceptible de générer des structures complexes et peu réalistes. Pour cette raison, ces algorithmes sont basés majoritairement sur des opérateurs de perturbation définis au niveau des fragments moléculaires et non des atomes.

Récemment, de nombreux travaux proposent des méthodes de génération moléculaire basées sur des modèles d'apprentissage profond [ELTON et al. 2019]. Leur développement a été permis par la mise à disposition récente de jeux de données de grande taille en chimie moléculaire [GAULTON et al. 2017; IRWIN et al. 2020; RAMAKRISHNAN et al. 2014]. L'intérêt pour ces approches est justifié en partie par le fait qu'elles peuvent apprendre et reproduire la distribution des propriétés d'un jeu de données, et qu'elles tendent à générer des molécules plus réalistes [BROWN et al. 2019].

En parallèle, une nouvelle vague d’algorithmes évolutionnaires ont été proposés ces dernières années. Ceux-ci s’avèrent au moins aussi performants que les approches basées sur l’apprentissage profond pour des problèmes d’optimisation de propriétés moléculaires [BROWN et al. 2019]. Parmi ces travaux, certains mettent également à profit des jeux de données afin de promouvoir le réalisme des solutions générées. CReM [POLISHCHUK 2020] et LEADD [KERSTJENS et DE WINTER 2022] utilisent des fragments moléculaires extraits dynamiquement de ChEMBL [GAULTON et al. 2017]. La fréquence d’application des opérateurs de mutation de GB-GA [JENSEN 2019] est obtenue par analyse statistique des caractéristiques correspondantes dans ZINC [IRWIN et al. 2020].

Dans ce chapitre, nous proposons un algorithme évolutionnaire pour l’optimisation de propriétés moléculaires. Nous faisons ce choix pour la qualité espérée des performances d’optimisation, mais également car nous souhaitons proposer une méthode générique pouvant être adaptée à faible coût à des problèmes d’optimisation très différents issus de différents domaines de la chimie organique. En outre, nous montrons que nos choix de conception permettent la génération de visualisations interprétables pour les utilisateurs chimistes. Finalement, nous étudions différentes approches pour favoriser le réalisme des solutions générées ; sous la forme d’un terme inséré dans la fonction objectif ou d’une contrainte binaire restreignant l’espace de recherche accessible.

## 2.2 EvoMol : un algorithme pour l’optimisation de propriétés moléculaires

### 2.2.1 Conception de notre approche

Nous concevons notre approche dans l’objectif qu’elle puisse répondre à plusieurs caractéristiques qui nous semblent importantes. En particulier, nous attendons de notre approche qu’elle permette de traiter des problèmes divers, qu’il existe des mécanismes permettant de contrôler les biais intégrés à la recherche, et qu’elle soit au maximum interprétable. Ces caractéristiques justifient le choix de concevoir un algorithme évolutionnaire, mais également certains choix de conception que nous détaillerons dans la suite de ce chapitre.

**Généricité** En premier lieu, nous souhaitons proposer une méthode générique au sens qu’elle doit pouvoir être appliquée efficacement à des problèmes issus de différents do-

maines de la chimie et plus précisément de la chimie organique. Nous nous intéresserons en particulier à des problèmes issus du domaine de la chimie pharmaceutique ainsi qu'à des problèmes issus du domaine de la chimie des matériaux moléculaires organiques. La chimie pharmaceutique est le domaine d'application le plus étudié dans la littérature des méthodes de génération moléculaire. La chimie des matériaux moléculaires organiques correspond à notre domaine d'application cible. Cette généralité est traduite par la limitation des hypothèses sur l'espace de recherche dans la version de base de notre algorithme. Cela signifie qu'au maximum, nous évitons de faire des choix de conception adaptés à un domaine de la chimie en particulier, et également que nous évitons que notre approche soit dépendante d'un jeu de données moléculaires dans sa version de base. Cet objectif de généralité est également traduit par la conception et la mise à disposition d'une implémentation de notre méthode très facilement adaptable à tout problème d'optimisation moléculaire exprimé sous la forme d'une fonction objectif et d'une spécification de l'espace de recherche.

**Contrôle des biais** Pour résoudre un problème d'optimisation moléculaire en particulier, il est souvent bénéfique d'intégrer à la méthode de résolution une connaissance métier apportée explicitement par un expert ou implicitement par un jeu de données moléculaires. Cette connaissance peut par exemple correspondre à une spécification de l'espace de recherche (taille des molécules, types d'atomes, etc.) ou à une sélection pertinente d'opérateurs de perturbation. Cette connaissance peut également être extraite d'un jeu de données, par exemple en favorisant la génération de caractéristiques moléculaires représentées dans un jeu de données en particulier. Dans le cadre de notre approche, nous cherchons à pouvoir intégrer ces biais de manière explicite et contrôlée. Ceux-ci peuvent être intégrés notamment par la spécification des paramètres de recherche (spécification de l'espace de recherche et des opérateurs de perturbation), au sein de la fonction objectif ou encore sous la forme d'une population initiale.

**Interprétabilité** Nous souhaitons également proposer une approche permettant aux utilisateurs du domaine d'application d'étudier les chemins parcourus dans l'espace moléculaire lors de la recherche. Nous espérons en particulier que cela permette d'étudier les relations entre structures et propriétés moléculaires. Nous espérons qu'en plus des molécules proposées à la fin de la procédure d'optimisation, les chimistes aient la possibilité de tirer des connaissances de la recherche et d'en avoir une meilleure compréhension. Nous

proposerons en particulier une visualisation de la recherche a posteriori sous la forme d'un arbre d'exploration.

## 2.2.2 Espace de recherche

### Représentation sous forme de graphe moléculaire

Nous choisissons de représenter les molécules comme des graphes moléculaires. Cela va nous permettre de définir dans la section suivante de ce chapitre un voisinage intuitif et facilement compréhensible pour des chimistes, basé sur des transformations locales du graphe moléculaire. Comme nous l'avons exposé au sein du chapitre précédent, nous considérons des graphes moléculaires avec représentation implicite des atomes d'hydrogène. Cela va nous permettre de simplifier la définition du voisinage moléculaire, puisque cela revient à manipuler uniquement les atomes lourds. Ces derniers sont liés automatiquement à des atomes d'hydrogène lorsque la somme de leurs liaisons n'atteint pas la valence attendue.

Pour qu'un graphe moléculaire soit considéré comme valide, deux conditions doivent être respectées. D'abord, les règles de valence doivent être respectées. Cela correspond au concept de validité sémantique décrit dans le Chapitre 1. Puisque les atomes d'hydrogène sont considérés implicitement, la valence de tous les atomes est toujours atteinte et un atome ne peut « manquer » d'une ou de plusieurs liaisons. Il doit toutefois être vérifié qu'aucun atome ne forme plus de liaisons que possible selon son type lorsque les graphes moléculaires sont manipulés. Ensuite, il doit également être vérifié que le graphe moléculaire est connexe, c'est-à-dire qu'il ne représente pas plusieurs molécules distinctes en son sein. Si la valence d'un graphe moléculaire est respectée et qu'il est connexe, alors il est valide.

L'espace de recherche que nous considérons est donc celui des graphes moléculaires valides. Par mesure de simplicité, nous serons amenés à parler simplement d'espace des graphes moléculaires. Nous détaillons les mécanismes permettant d'assurer la validité des graphes moléculaires dans la section 2.2.3 qui décrit les opérateurs de perturbation du graphe moléculaire (ou actions sur le graphe moléculaire) que nous proposons. L'espace de recherche des graphes moléculaires est paramétré par la taille des molécules ainsi que par les types d'atomes considérés. Nous exprimons la taille des molécules par rapport au nombre d'atomes lourds qui sont contenus.

## Espace de recherche sous contraintes

Nous définissons au sein de notre méthode d'optimisation évolutionnaire un ensemble de mécanismes permettant de restreindre l'espace de recherche accessible. Ces mécanismes sont les suivants.

**Restriction de l'espace de recherche selon des contraintes** Notre algorithme offre d'abord la possibilité d'intégrer une contrainte binaire sur les solutions, afin de restreindre l'espace de recherche accessible. Nous définissons cette contrainte sous la forme d'une fonction  $f_{\text{cont}}$  assignant une valeur booléenne à tout graphe moléculaire. Lors de la recherche de solutions améliorantes par l'application des opérateurs de perturbation, les solutions mutantes invalides selon  $f_{\text{cont}}$  seront ignorées. Ces contraintes nous permettront notamment de définir un filtrage de l'espace de recherche dans l'objectif d'améliorer le réalisme des solutions en section 2.4. Par défaut toutes les solutions sont accessibles, c'est-à-dire que  $f_{\text{cont}}$  est défini comme  $f_{\text{cont}}(x) = \text{Vrai}$  pour toute solution  $x$ .

**Liste taboue** Nous intégrons à notre algorithme la possibilité d'intégrer préalablement à la recherche une liste de solutions  $L_{\text{tabou}}$  ne pouvant pas être générées. Cette liste est nommée *taboue* en référence aux méthodes de recherche taboue. Dans le cadre de nos travaux, nous utiliserons ce paramètre au sein du Chapitre 4.

**Gel du graphe moléculaire** Nous intégrons également à notre algorithme la possibilité de filtrer les perturbations du graphe moléculaire applicables en fonction d'une étiquette marquant certains atomes du graphe moléculaire. L'objectif est de définir la possibilité de « geler » certains atomes d'une molécule, de sorte qu'ils ne puissent pas être modifiés ni supprimés par la procédure d'optimisation. On souhaite qu'il soit tout de même possible de brancher de nouveaux atomes aux atomes gelés. Nous définissons pour cela une contrainte supplémentaire limitant la validité des perturbations du graphe moléculaire en fonction d'une liste  $L_{\text{gel}}$  des atomes gelés. Nous décrivons le fonctionnement de cette contrainte dans la section suivante, après avoir défini les actions sur le graphe moléculaire qui correspondent à ces perturbations. Nous illustrons une application de ce filtre en section 2.3.3.

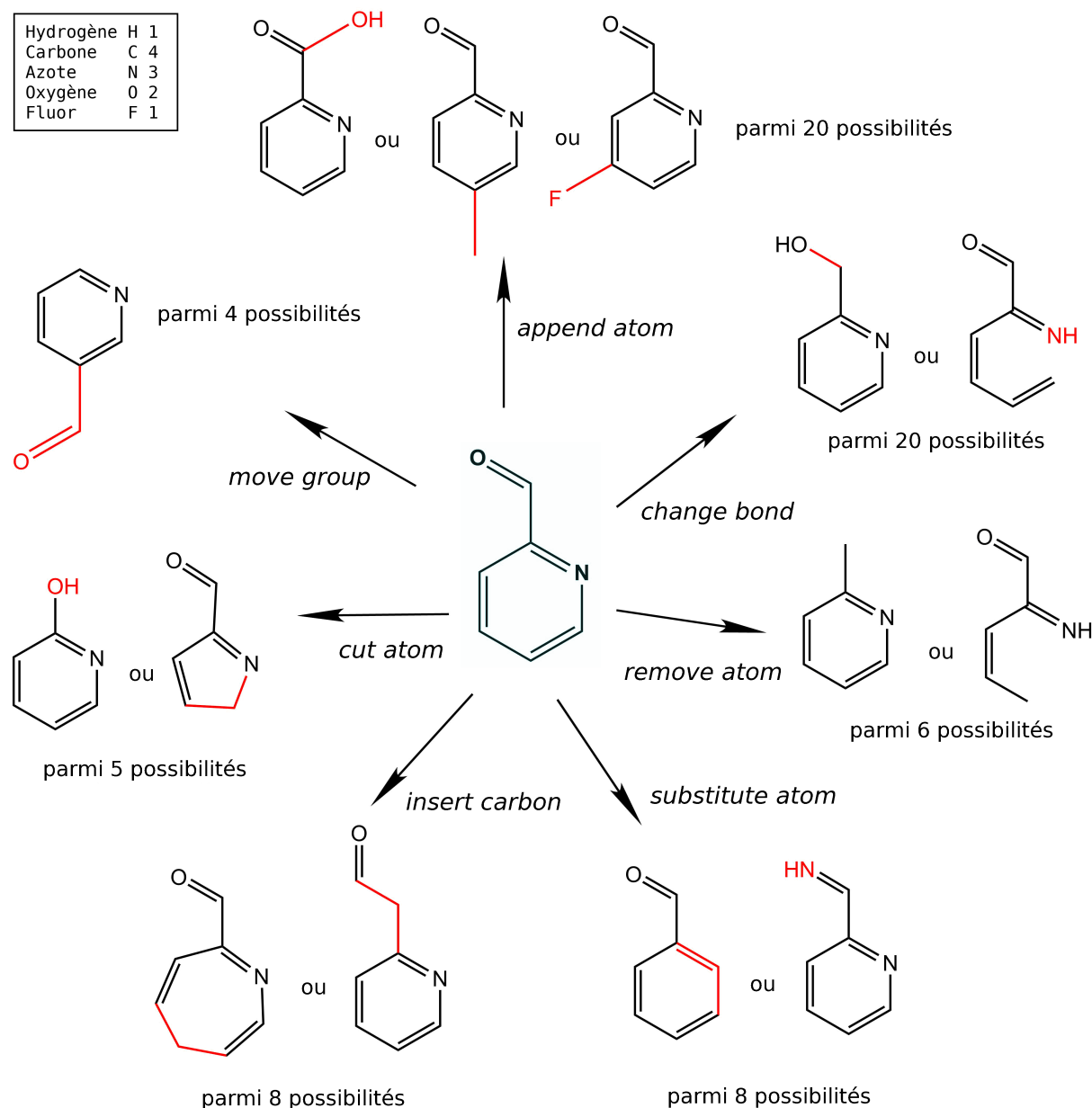


FIGURE 2.1 – Application des actions sur le graphe moléculaire à la molécule de 2-Formylpyridine (au centre), en utilisant un espace de recherche défini à partir de l'ensemble d'atomes lourds {C, N, O, F}. Seules les applications d'actions correspondant à un graphe moléculaire valide sont considérées. La coloration est utilisée pour faire apparaître les différences et non les types des atomes. Pour faciliter l'interprétation, les types d'atomes ainsi que leurs symboles et leurs valences sont rappelés en partie supérieure gauche.

## 2.2.3 Parcours de l'espace de recherche

### Actions sur le graphe moléculaire

L'opérateur de mutation que nous définissons est basé sur l'application successive d'une ou de plusieurs actions sur le graphe moléculaire. Ces actions permettent de définir un voisinage moléculaire fin et intuitif pour des chimistes. Il peut en effet se rapporter aux transformations qui ont lieu lors des réactions chimiques, bien que les actions ne correspondent pas strictement à des réactions chimiques.

**Actions primaires** Nous définissons un ensemble d'actions sur le graphe moléculaire, que nous classons en deux catégories. La première est celle des actions que nous considérons « primaires », c'est-à-dire dont le comportement est très simple et qui définissent un voisinage très local. Ces actions sont décrites ci-dessous, et sont également illustrées en Figure 2.1. Pour rappel, nous considérons une représentation implicite des atomes d'hydrogène au sein des graphes moléculaires. Ainsi, les actions sur le graphe moléculaire peuvent entraîner l'ajout ou la suppression implicite d'atomes d'hydrogène.

- *append atom* : ajout d'un atome au graphe moléculaire, lié par une liaison simple à un atome existant. Dans la Figure 2.1, un atome de type C, N, O ou F peut être ajouté à n'importe quel atome du graphe moléculaire dont la valence n'est pas déjà atteinte par des liaisons explicites. Cela correspond à la totalité des atomes de carbone, à l'exception de l'atome du cycle qui est lié au groupe contenant un atome de carbone et un atome d'oxygène.
- *remove atom* : suppression d'un atome du graphe moléculaire, ainsi que des liaisons auxquelles il est associé. Cette action peut être appliquée uniquement aux atomes dont la suppression ne mène pas à la séparation du graphe moléculaire en plusieurs composantes distinctes. Ainsi en Figure 2.1, tous les atomes de la molécule centrale peuvent être supprimés à l'exception de l'atome de carbone qui ne fait pas partie du cycle et de l'atome de carbone du cycle auquel il est lié.
- *change bond* : changement du type d'une liaison. Nous considérons ici l'absence de liaison comme un type de liaison, que l'on peut considérer de type nul. Ainsi, cet opérateur permet de créer et supprimer des liaisons. Toute liaison de type {nul, simple, double, triple} peut être transformée en une liaison d'un autre type, à condition que la valence des atomes impliqués dans cette liaison demeure respectée et que la suppression d'une liaison n'implique pas la séparation du graphe



moléculaire en plusieurs composantes distinctes.

**Actions secondaires** La seconde catégorie d’actions sur le graphe moléculaire est celle des actions que l’on considère « secondaires ». Celles-ci ne permettent pas d’atteindre des portions supplémentaires de l’espace moléculaire, mais définissent un voisinage plus complexe qui permet d’effectuer des altérations plus importantes mais toujours compréhensibles et intuitives. Ces actions sont décrites ci-dessous, et sont également illustrées en Figure 2.1.

- *substitute atom* : changement du type d’un atome du graphe moléculaire. Tout atome lourd peut être transformé en un atome lourd d’un autre type, à condition que sa valence soit toujours respectée. En Figure 2.1, on observe par exemple que l’atome d’azote formant déjà trois liaisons explicites peut être transformé en atome de carbone, de valence 4. Il ne pourrait en revanche pas être transformé en atome d’oxygène, de valence 2.
- *insert carbon* : insertion d’un atome de carbone entre deux atomes qui partageaient précédemment une liaison simple, double ou triple. La liaison initiale est détruite et est remplacée par deux liaisons simples. L’intérêt de cette action est qu’elle permet d’agrandir simplement des chaînes d’atomes ou des cycles. Il s’agit d’une opération relativement complexe qui sans cet opérateur nécessiterait l’application successive d’un certain nombre d’actions primaires.
- *cut atom* : suppression d’un atome d’une chaîne tout en reconstruisant les liaisons nécessaires pour conserver la chaîne. L’atome supprimé doit être lié à exactement deux atomes, qui ne doivent eux-mêmes pas être liés entre eux. Une liaison simple est créée entre ces derniers à l’issue de la suppression. Nous n’avons pas retenu la possibilité de créer une liaison double ou triple car le type de la liaison pourra être transformé ultérieurement par l’application de l’action *change bond*. Cette action permet de raccourcir une chaîne d’atomes ou un cycle.
- *move group* : déplacement d’un groupe d’atomes, en conservant le type de la liaison déplacée. Toute liaison ne faisant pas partie d’un cycle permet de définir deux groupes d’atomes, à ses deux extrémités. Cette liaison est coupée et chaque groupe peut être lié à tout atome de l’autre groupe dont les liaisons explicites et la valence le permettent. Pour la molécule présentée en Figure 2.1, deux paires de groupes peuvent être extraites. Ces groupes sont représentés en Figure 2.2. Le groupe **a** peut être lié au groupe **b** uniquement par la liaison qui préexistait. Il en est de

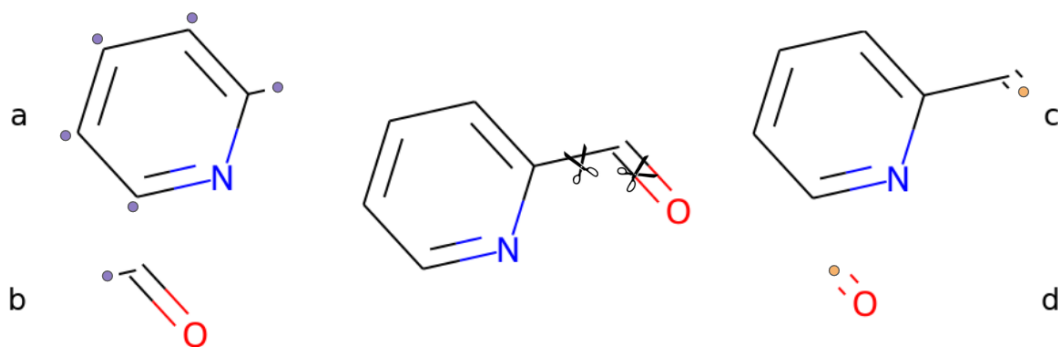


FIGURE 2.2 – Au centre, dessin moléculaire de la molécule de 2-Formylpyridine avec deux paires de ciseaux pour mettre en évidence les groupes pouvant être déplacés selon l'action *move group*. À gauche et à droite, les deux paires de groupes (a, b) et (c, d) en question. Des pastilles de couleur font apparaître les branchements possibles pour ces groupes.

même pour le groupe **c** par rapport au groupe **d** ainsi que pour le groupe **d** par rapport au groupe **c**. En revanche, le groupe **b** peut être lié à 4 atomes différents du groupe **a** si l'on ignore la position de la liaison qui préexistait. Cela correspond aux 4 voisins de la molécule de 2-Formylpyridine selon l'action *move group*.

## Validité des solutions

**Vérification de la validité des graphes moléculaires** Il est possible de filtrer a priori l'ensemble des actions sur le graphe moléculaire applicables à une solution. La contrainte de valence peut être vérifiée simplement en filtrant toutes les mutations entraînant le dépassement de la valence d'un atome du graphe (la somme des liaisons explicites dépasse la valence maximale de l'atome en question). Il n'est pas nécessaire de vérifier qu'il existe un nombre minimal de liaisons puisque des liaisons implicites sont formées avec des atomes d'hydrogène de sorte que la valence des atomes est toujours atteinte. Toutes les actions sont susceptibles d'enfreindre la contrainte de valence, à l'exception de *remove atom* qui ne crée aucune liaison ni ne change le type d'un atome, ainsi que *insert carbon* et *cut atom* qui remplacent des liaisons quelconques par des liaisons simples.

La contrainte de connexité peut être vérifiée en filtrant toutes les perturbations entraînant la suppression d'une liaison qui est un pont du graphe moléculaire, c'est-à-dire dont la suppression entraîne l'augmentation du nombre de composantes connexes. Il existe des algorithmes efficaces pour la détection des ponts, en complexité linéaire [SCHMIDT 2013]. Dans notre implémentation, nous utilisons pour cela l'implémentation de la bibliothèque

NetworkX [HAGBERG et al. 2008]. Les actions susceptibles d’altérer la connexité du graphe moléculaire sont *remove atom* et *change bond*.

### Contraintes sur l’espace de recherche

**Gel du graphe moléculaire** La contrainte de gel du graphe moléculaire implique de définir un filtre spécifique des actions applicables. Pour rappel, nous laissons la possibilité de « geler » un ensemble d’atomes du graphe moléculaire de sorte qu’ils ne puissent pas être modifiés ni supprimés, mais qu’ils puissent toutefois former de nouvelles liaisons s’ils disposent d’électrons non appariés à des atomes lourds. Ce filtre est défini de la façon suivante, avec  $L_{\text{gel}}$  la liste des atomes gelés. Les actions *remove atom*, *substitute atom* et *cut atom* ne peuvent être appliquées que sur des atomes n’appartenant pas à  $L_{\text{gel}}$ . De plus, les actions *change bond*, *insert carbon* et *move group* ne peuvent être appliquées que si la liaison sur laquelle elles sont appliquées implique au moins un atome n’appartenant pas à  $L_{\text{gel}}$ .

**Gestion des charges** Les actions sur le graphe moléculaire que nous avons définies ne permettent pas de créer des charges sur les atomes des molécules, car cela serait susceptible d’entraîner la génération d’une forte proportion de molécules très peu réalistes. Les atomes chargés tendent en effet à se retrouver uniquement dans certains groupes d’atomes typiques, tels que le groupe nitro évoqué dans la section 1.1.2 de ce mémoire. Cependant, nous définissons une procédure supplémentaire permettant d’appliquer les actions précédemment définies sur des molécules contenant des charges. Dans ce contexte, les actions sur le graphe moléculaire ne peuvent pas s’appliquer sur les liaisons impliquant des atomes chargés, à l’exception de l’opérateur de suppression des atomes. Ainsi, des molécules contenant des charges peuvent être comprises dans la population initiale de notre algorithme évolutionnaire. Les charges seront alors conservées ou supprimées pendant la recherche.

### Opérateur de mutation

Nous définissons une mutation comme étant l’application successive d’une ou plusieurs actions sur le graphe moléculaire. La possibilité d’effectuer des mutations constituées de plusieurs actions est donnée afin de favoriser la capacité de la procédure d’optimisation à échapper à des minimums locaux, si aucun améliorant n’existe dans un voisinage très

proche. Lorsqu'une mutation doit être appliquée, sa profondeur (le nombre d'actions successives) est tirée aléatoirement entre 1 et un paramètre définissant la profondeur maximale. En fonction de la profondeur de la mutation, les actions sont successivement tirées aléatoirement puis appliquées. Le tirage aléatoire de chaque action est effectué de la façon suivante. Le type de l'action (*append atom*, *remove atom*, etc.) est d'abord tiré selon une loi uniforme. Puis, l'action qui sera appliquée est tirée aléatoirement parmi la liste des perturbations valides selon les contraintes de connexité, de valence et de gel du graphe moléculaire pour le type d'action en question selon l'état de l'individu muté.

Nous ne définissons en revanche pas d'opérateur de recombinaison. L'utilisation exclusive de l'opérateur de mutation qui est basé sur les actions sur le graphe moléculaire correspond à une exploration de l'espace de recherche par voisinage direct. Nous souhaitons étudier les performances d'optimisation de notre approche dans ce contexte. De plus, l'absence d'opérateur de recombinaison facilite la définition de visualisations interprétables de l'espace de recherche exploré, sous la forme d'un arbre d'exploration (voir les Figures 2.4 et 2.10).

## 2.2.4 Algorithme

Nous décrivons désormais le fonctionnement de notre algorithme évolutionnaire. Précisons que nous préférons parler d'un algorithme évolutionnaire plutôt que d'un algorithme génétique en raison de choix que nous avons effectués lors de sa conception. En particulier, nous choisissons une représentation des solutions sous forme de graphe moléculaire, ce qui diffère de la représentation binaire classiquement attendue d'un algorithme génétique. De plus, notre algorithme n'utilise pas d'opérateur de recombinaison, opérateur qui est typiquement attendu d'un algorithme génétique. Pour ces raisons, nous utilisons le terme d'algorithme évolutionnaire, qui est plus générique.

Le fonctionnement de notre algorithme évolutionnaire est détaillé en Algorithme 1. Dans un premier temps, les paramètres de la recherche doivent être spécifiés. L'espace de recherche est caractérisé à travers le choix des actions sur le graphe moléculaire utilisées au sein de l'opérateur de mutation, de l'ensemble des atomes lourds considérés et de la taille maximale des solutions en nombre d'atomes lourds. De plus, des contraintes binaires définies par la fonction  $f_{\text{cont}}$  et la liste  $L_{\text{tabou}}$  peuvent être définies afin de restreindre l'espace des solutions accessibles. Le problème à résoudre est caractérisé sous la forme d'une fonction objectif  $f_{\text{obj}}$ , et les paramètres internes de l'algorithme sont spécifiés. Nous considérons par défaut que la fonction objectif doit être maximisée, sauf mention contraire.

---

**Algorithme 1** Algorithme évolutionnaire pour l'optimisation de propriétés moléculaires

---

**entrée:**  $S$  l'espace de recherche (types d'atomes, taille max. des molécules),  
 $S_{\text{cont}}$  l'espace  $S$  contraint par  $f_{\text{cont}}$ ,  $L_{\text{tabou}}$  et  $L_{\text{gel}}$ ,  
 $f_{\text{obj}}$  la fonction objectif,  
 $TaillePopMax$  la taille maximale de la population,  
 $TailleLot$  le nombre d'individus devant être remplacés à chaque étape  
init. de la population  $pop$  de taille  $\leq TaillePopMax$   
**tant que** le critère d'arrêt n'est pas atteint **faire**  
  tri de la population pour obtenir la pile  $P$  des individus à muter à l'étape courante  
  sélection de la liste  $L$  d'individus à remplacer à l'étape courante, de taille  $TailleLot$   
  **pour** chaque individu  $ind$  de la liste  $L$  **faire**  
    **répéter**  
      **si**  $P$  n'est pas vide **alors**  
         $c \leftarrow \text{sommet}(P)$   
        recherche améliorant  $mut \in S_{\text{cont}}$  de  $ind$  selon  $f_{\text{obj}}$  par mutation de  $c$   
        **si** un améliorant  $mut$  est trouvé **alors**  
          remplacer  $ind$  par  $mut$  dans la population  
        **fin si**  
      dépiler( $P$ )  
    **fin si**  
  **jusqu'à** ce qu'un améliorant soit trouvé ou que la pile  $P$  soit vide  
  **fin pour**  
**fin tant que**

---

Pour rappel, il suffit de considérer l'opposé de la fonction objectif pour transformer un problème de minimisation en problème de maximisation. La dernière étape nécessaire avant le démarrage de la recherche consiste à sélectionner la population initiale. Cette sélection peut être une façon d'intégrer de la connaissance sur le problème d'optimisation, si des bonnes solutions ou des structures moléculaires pertinentes sont connues. La taille effective de la population initiale peut être inférieure au paramètre définissant la taille maximale de la population. Dans ce cas, la population initiale comprend également un ensemble d'individus fictifs non définis ayant une valeur de la fonction objectif égale à  $-\infty$ ; ces derniers seront ainsi remplacés en priorité dans les premières étapes de l'algorithme.

La procédure d'optimisation consiste ensuite à effectuer successivement des étapes de remplacement d'un ensemble d'individus de la population, jusqu'à ce qu'un critère d'arrêt soit atteint. Dans nos travaux, nous utiliserons comme critère d'arrêt une limite sur le nombre d'étapes d'optimisation ou sur le nombre d'appels à la fonction objectif. À chaque étape, des individus sont sélectionnés pour être remplacés par des améliorants obtenus par mutation d'une pile d'individus. Ainsi, une liste  $L$  de *TailleLot* individus susceptibles d'être remplacés est sélectionnée en début d'étape. Les individus sélectionnés sont les *TailleLot* individus possédant les valeurs de fonction objectif les plus faibles. Si la taille effective de la population est inférieure au paramètre de taille maximale, cette liste contient des individus non définis, et la taille effective de la population augmentera donc en fin d'étape.

Pour remplacer un individu *ind* de  $L$  par un améliorant, on examine les mutations possibles des individus de la population courante. On considère la population courante comme une pile  $P$  triée selon les valeurs décroissantes de la fonction objectif, de sorte que les individus mutés en priorité sont les individus possédant les plus hautes valeurs de fonction objectif. Pour chaque individu que l'on souhaite remplacer, on recherche un améliorant par mutation de l'individu au sommet de la pile. Si un améliorant est trouvé, nous effectuons le remplacement dans la population. Sinon, un améliorant est cherché par mutation de l'élément suivant dans la pile. Si la pile est vide, alors l'étape courante est interrompue, sans que *TailleLot* individus aient pu être remplacés. Dans ce schéma, le nombre d'individus qui seront mutés ne peut être prévu à l'avance.

Lorsque la population initiale contient moins de solutions définies que la valeur du paramètre *TailleLot*, la pile  $P$  ne contient pas suffisamment de solutions candidates à la mutation pour que *TailleLot* individus puissent effectivement être remplacés. Le nombre d'individus remplacés est donc le minimum entre la taille effective de la population et

*TailleLot* (en considérant que des améliorants sont obtenus avec succès). Ainsi, si la population à l'initialisation contient 1 individu défini et que le paramètre *TailleLot* est fixé à 10, elle en contiendra 2 à l'étape suivante, puis 4, puis 8, puis 16, puis 26, 36, 46, etc.

---

**Algorithme 2** Recherche d'une solution améliorante

---

**entrée:**  $S$  l'espace de recherche (types d'atomes, taille max. des molécules),

$S_{\text{cont}}$  l'espace  $S$  contraint par  $f_{\text{cont}}$ ,  $L_{\text{tabou}}$  et  $L_{\text{gel}}$ ,

$f_{\text{obj}}$  la fonction objectif,

$pop$  la liste des individus au sein de la population courante,

$ind$  la solution dont on cherche un améliorant,

$c$  l'individu candidat à la mutation,

*EssaisMax* le nombre d'essais maximal pour chercher un améliorant

**répéter**

mutation de  $c$  avec respect de la contr. définie par  $L_{\text{gel}}$  pour obtenir la sol.  $mut \in S$

**si**  $((f_{\text{obj}}(mut) \geq f_{\text{obj}}(ind)) \text{ et } (f_{\text{cont}}(mut)) \text{ et } (mut \notin L_{\text{tabou}} \cup pop))$  **alors**  
un améliorant  $mut \in S_{\text{cont}}$  a été trouvé

**fin si**

**jusqu'à** ce que *EssaisMax* essais aient été effectués ou qu'un améliorant ait été trouvé

---

La procédure de recherche de solutions améliorantes est formalisée dans l'Algorithme 2. Elle dépend d'un paramètre supplémentaire *EssaisMax* qui définit le nombre maximal de tentatives effectuées pour chercher une solution améliorante d'une solution notée *ind* par la mutation d'un individu noté *c*. La procédure de génération est effectuée en deux temps afin de garantir que les individus générés respectent l'ensemble des contraintes. Cela est dû au fait que le respect d'une partie des contraintes peut être garanti dès la mutation par filtrage des actions sur le graphe moléculaire, tandis que le respect des autres contraintes ne peut être vérifié qu'après l'application de la mutation.

Les contraintes qui sont garanties par filtrage des actions lors de la mutation sont la validité du graphe moléculaire et le gel du graphe moléculaire. La validité du graphe moléculaire qui correspond au respect de la valence et au respect de la contrainte de connexité (voir la section 2.2.3) est ici sous-entendue. Nous faisons en revanche apparaître de manière explicite dans l'algorithme l'application de la contrainte de gel du graphe moléculaire. Après qu'une solution appartenant à l'espace des graphes moléculaires valides et respectant la contrainte de gel a été générée, nous vérifions qu'il s'agit d'un améliorant de la solution qui doit être remplacée, qu'elle respecte la contrainte  $f_{\text{cont}}$  et qu'elle n'appartient pas à la liste taboue ni à la population courante. Cette dernière contrainte est définie pour garantir que la recherche ne converge pas vers une unique solution et pour garantir

l'unicité des individus de la population. Si une solution mutante ne respecte pas l'une de ces conditions, elle est ignorée et une nouvelle tentative est effectuée si le nombre maximal de tentatives n'est pas atteint.

**Implémentation** Une implémentation en langage Python de notre algorithme est mise à disposition sous licence libre sur GitHub<sup>1</sup>. Elle dépend notamment de la bibliothèque RDKit, pour représenter et traiter les molécules [LANDRUM 2010]. Notre implémentation est pensée pour être utilisable simplement par des utilisateurs du domaine d'application. Une interface sous forme de dictionnaire permet de décrire la fonction objectif et de spécifier l'ensemble des paramètres de l'algorithme. Cette interface est documentée sur la page GitHub. À titre d'exemple, nous reportons ci-dessous le code permettant de reproduire l'expérience de maximisation des valeurs de QED avec une population de taille 1000 que nous présentons en section 2.3.1. Le programme peut être lancé avec une utilisation minimale de code. Il est également possible d'utiliser en tant que fonction objectif une fonction Python quelconque évaluant une molécule.

```
1 from evomol import run_model
2
3 run_model({
4     "obj_function": "qed",           # Fonction objectif
5     "optimization_parameters": {
6         "pop_max_size": 1000,       # Parametre TaillePopMax
7         "max_steps": 1500,          # Critere d'arret (etapes)
8         "mutation_max_depth": 2,    # Profondeur max. des mutations
9         "k_to_replace": 10          # Parametre TailleLot
10    },
11    "action_space_parameters": {
12        "atoms": "C,N,O,F,P,S,Cl,Br", # Types des atomes lourds
13        "max_heavy_atoms": 38         # Taille max. des solutions
14    }
15 })
```

---

1. <https://github.com/jules-leguy/EvoMol>



## 2.3 Évaluation

### 2.3.1 Optimisation de propriétés peu coûteuses

Nous proposons d'étudier les performances d'EvoMol pour la maximisation de plusieurs propriétés moléculaires communément optimisées dans la littérature, à savoir le logP pénalisé (plogP), sa variante normalisée, et la QED (voir la section 1.4 de ce mémoire). Ces trois propriétés peuvent être calculées selon un coût très faible. Nous utilisons l'optimisation des valeurs de QED en tant qu'expérience de référence à travers les différents chapitres de ce mémoire. Nous comparons les résultats obtenus à plusieurs méthodes de l'état de l'art, présentées en section 1.3. Parmi ces méthodes, ChemGE [YOSHIKAWA et al. 2018] et l'approche proposée par [NIGAM et al. 2020] sont des algorithmes évolutionnaires basés sur une représentation des molécules sous forme de texte. GB-GA est un algorithme évolutionnaire basé sur une représentation sous forme de graphe moléculaire. Il s'agit de la méthode la plus proche d'EvoMol. Une différence importante est que GB-GA dispose d'un opérateur de recombinaison [JENSEN 2019]. MSO est un autre algorithme métaheuristique, d'optimisation par essaim particulière [WINTER et al. 2019]. MolDQN [ZHOU et al. 2019], GCPN [YOU et al. 2018] et les travaux proposés par [ZHANG et al. 2019] sont des algorithmes d'apprentissage par renforcement basés sur des modèles d'apprentissage profond.

#### Conditions expérimentales

Nous étudions les performances d'optimisation en faisant varier certains des paramètres de l'algorithme. En particulier, nous choisissons d'effectuer deux jeux d'expériences distincts utilisant une taille de population (*TaillePopMax*) de 1 et 1000. Dans le cas où la population ne contient qu'un individu, notre algorithme se comporte comme un *hill-climber* avec sélection du premier améliorant, c'est-à-dire qu'une unique solution est améliorée par voisinage local et qu'aucune stratégie n'est utilisée pour échapper aux minimums locaux. La recherche s'effectue dans un espace relativement large de la chimie organique, avec des solutions pouvant contenir des atomes lourds de type C, N, O, F, P, S, Cl et Br. Les types d'atomes ayant une influence importante sur les valeurs du plogP (voir résultats), nous effectuons également un second jeu d'expériences avec un espace moléculaire dont les atomes lourds sont restreints à l'ensemble {C, N, O, F}. La population initiale de toutes les expériences est initialisée avec une unique molécule de méthane

Méthode		QED	plogP	plogP normalisé
ChemGE*				<i>5.88</i>
GB-GA				7.40
GCPN		0.948		7.98
MolDQN		0.948	11.84	
[ZHANG et al. 2019]		0.954	12.96	
MSO*		0.948	<i>26.8</i>	
[NIGAM et al. 2020]*				<i>20.72</i>
EvoMol	pop. 1	0.922	14.49	11.19
	pop. 1000	0.948	18.06	13.79
EvoMol {C, N, O, F}	pop. 1	0.902	13.88	11.19
	pop. 1000	0.948	13.88	11.19

TABLE 2.1 – Meilleures valeurs obtenues pour l’optimisation de la QED, du plogP et du plogP normalisé. Les valeurs reportées pour les méthodes de l’état de l’art sont tirées des articles originaux. Les résultats pour EvoMol correspondent à la moyenne du maximum obtenu sur 10 exécutions indépendantes. \*Absence de limite sur la taille maximale des molécules ou limite élevée, les valeurs des propriétés dépendant du logP sont reportées en italique à titre indicatif.

(c’est-à-dire contenant pour seul atome explicite un atome de carbone). Cela permet d’intégrer une connaissance minimale sur le problème devant être résolu. Afin d’obtenir des résultats comparables à la majorité des travaux de l’état de l’art, l’espace moléculaire est restreint aux molécules contenant jusqu’à 38 atomes lourds. Toutes les actions sur les graphes moléculaires sont utilisées pour définir l’opérateur de mutation. Le paramètre de profondeur maximale de mutation est fixé à 2. Cela signifie qu’une mutation consiste à appliquer aléatoirement une ou deux actions sur le graphe moléculaire. Le nombre d’individus remplacés à chaque étape (*TailleLot*) est fixé à 1 lorsque la population ne contient qu’un unique individu, et à 10 lorsqu’elle en contient 1000. Au plus 50 tentatives (*Essais-Max*) sont effectuées pour la recherche d’un améliorant. La procédure d’optimisation est stoppée après 1500 étapes, et toutes les expériences sont effectuées 10 fois.

## Résultats

Les résultats sont présentés en Table 2.1. Les résultats pour les méthodes de la littérature sont tirés des articles originaux. Toutes les méthodes présentées définissent une taille maximale comparable des molécules (38 atomes lourds), à l’exception des méthodes marquées d’une étoile. Ces dernières sont évaluées sans limite de taille des molécules ou avec une limite élevée. La taille des molécules ayant une influence importante sur la valeur

du plogP, cela doit être pris en compte pour analyser les résultats. Les scores de ces trois approches pour cette propriété sont donc reportés à titre indicatif. Précisément, ChemGE et MSO ne définissent pas de limite, tandis que [NIGAM et al. 2020] utilisent une limite sur la taille de la représentation SMILES de la molécule, qui ne peut dépasser 81 caractères. Cette limite peut mener à la génération de molécules contenant jusqu’à 81 atomes lourds.

Dans les conditions d’un *hill-climber*, EvoMol permet d’obtenir des solutions de scores élevés, et surpasse même les méthodes de la littérature comparables pour l’optimisation du plogP. Dans la plupart des cas, l’utilisation d’une population de taille supérieure permet comme attendu d’obtenir de meilleures solutions. Pour l’optimisation du plogP, il apparaît (voir Figure 2.3) que les meilleures solutions sont des chaînes de carbone (alcane) quand l’espace de recherche est restreint aux molécules contenant des atomes lourds parmi l’ensemble {C, N, O, F}. Il s’agit d’un résultat cohérent puisque ces molécules sont en effet très solubles dans l’octanol et ont une pénalité de synthétisabilité faible puisque les alcanes sont considérés comme une matière première. Lorsque l’espace de recherche est étendu, l’insertion d’atomes de S, P, Cl et Br en bout de chaîne permet d’augmenter le score. Ce comportement est plus inattendu chimiquement, et est lié à la façon dont la valeur de logP est estimée, c’est-à-dire comme une somme de contributions atomiques favorisant grandement les atomes de soufre, de brome et de phosphore [WILDMAN et CRIPPEN 1999]. L’observation de ces meilleures solutions permet facilement de comprendre pourquoi deux des trois méthodes n’appliquant pas de limite sur la taille des molécules ou appliquant une limite supérieure obtiennent des valeurs de plogP plus élevées. Les solutions qu’elles obtiennent sont des chaînes de carbone et/ou de soufre ou d’autres hétéroatomes contenant simplement plus d’atomes.

Concernant l’optimisation de la QED, notre approche permet d’obtenir des résultats très comparables à la littérature, à l’exception d’une méthode ([ZHANG et al. 2019]) qui obtient une valeur supérieure, dont la solution associée n’est pas donnée dans l’article présentant les résultats. L’analyse des solutions obtenues (voir Figure 2.3) montre que l’optimisation de cette propriété peut mener vers des solutions très peu réalistes, contenant une proportion importante d’hétéroatomes. Ces solutions possèdent une concentration importante de caractéristiques typiques des médicaments (atomes donneurs et accepteurs de liaisons hydrogène, cycles aromatiques, liaisons permettant une rotation), mais ne forment pas un médicament réaliste. L’évaluation de cette métrique dépend de RDKit, qui effectue une interprétation de l’aromaticité basée sur la règle de Hückel (voir la section 1.1.1 de ce mémoire). Or, il s’agit d’un cas limite de cette règle, et les chimistes qui analyseraient les

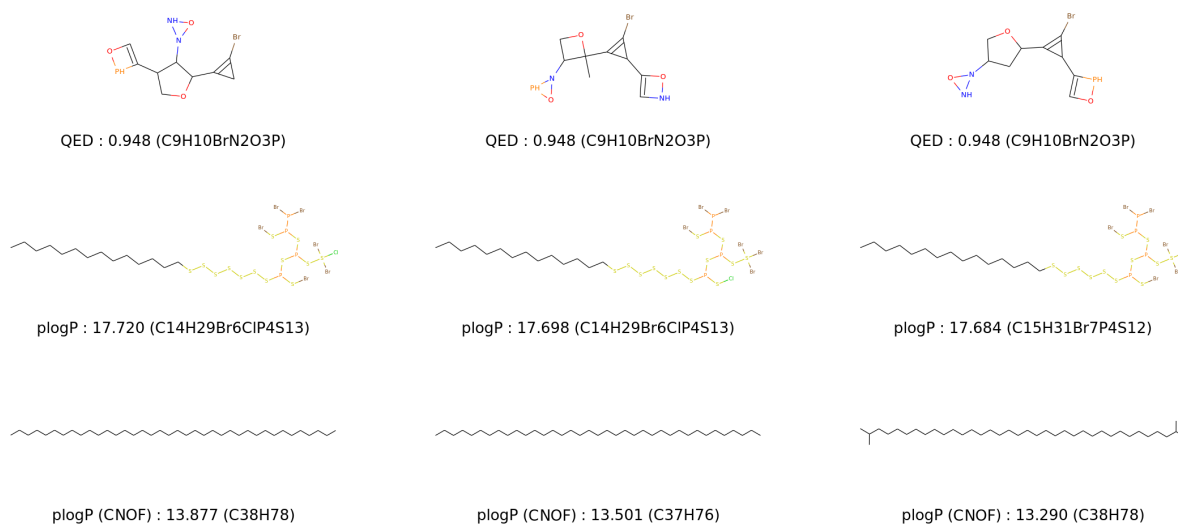


FIGURE 2.3 – Trois meilleures solutions obtenues pour l’optimisation la QED et du plogP avec une population de taille 1000. Pour chaque ligne, les solutions sont tirées d’une unique exécution tirée aléatoirement parmi les 10 exécutions de chaque expérience. La légende des molécules indique l’expérience représentée et la valeur de propriété cible, ainsi que la formule brute.

solutions présentées considéreraient certainement qu’il ne s’agit pas réellement de cycles aromatiques.

### Visualisation de l’exploration

Nous proposons en Figure 2.4 une visualisation de l’exploration qui a lieu lors de l’optimisation des valeurs de QED. Chaque point représente une solution générée et insérée dans la population lors de la recherche. Chaque arête représente une mutation. Le point de départ (méthane, de formule brute  $\text{CH}_4$ ) est mis en évidence par une flèche en gras. Pour mieux comprendre la recherche, nous colorons les nœuds selon les valeurs de QED. En particulier, il est intéressant de remarquer qu’il existe des solutions ayant donné lieu à un grand nombre de mutants. Ces solutions sont repérables par les arcs de cercles qu’elles forment par accumulation d’arêtes et de nœuds. Leur présence est liée au fonctionnement de notre algorithme, qui considère en priorité les meilleures solutions dans la pile  $P$  des individus candidats à la mutation. Les bonnes solutions sont susceptibles d’appartenir à la tête de la pile pendant plusieurs étapes et sont donc plus susceptibles de générer une quantité importante de descendants. Ces solutions sont progressivement abandonnées à

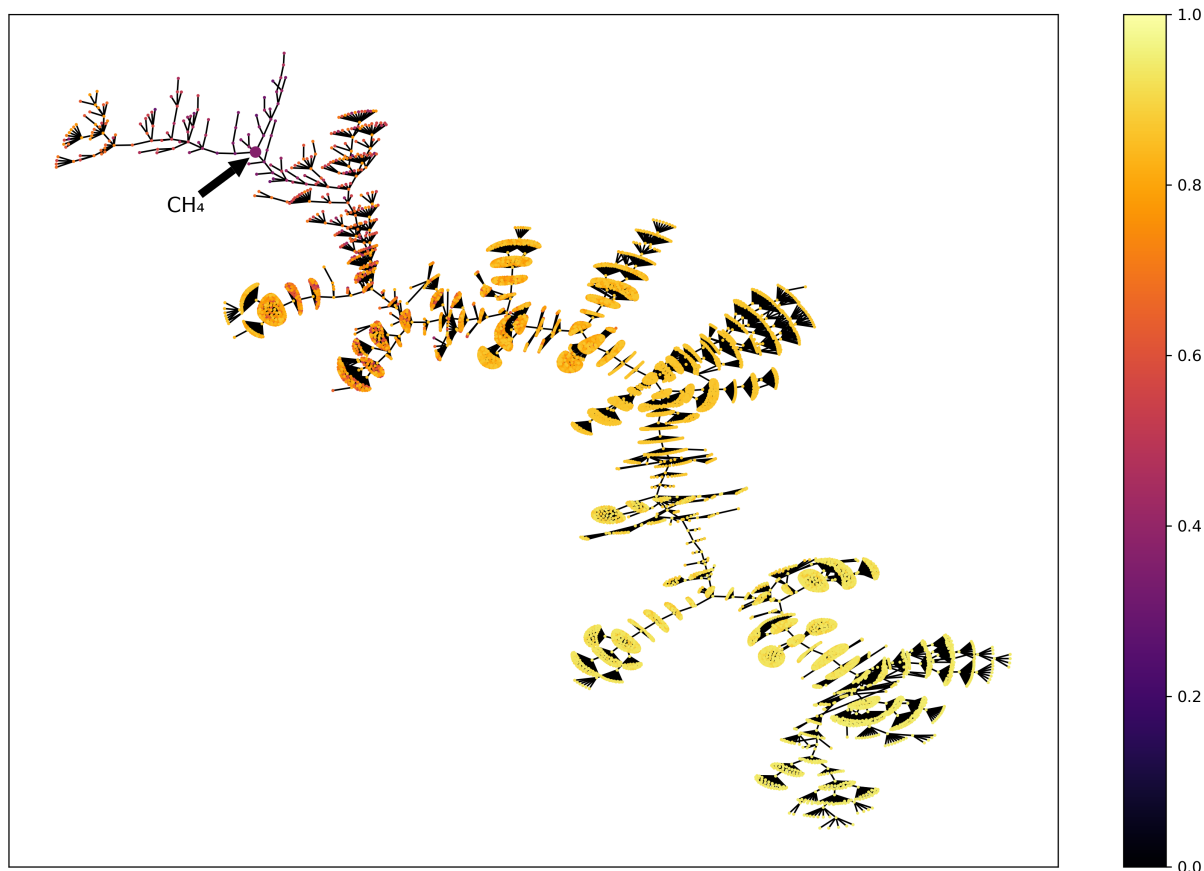


FIGURE 2.4 – Arbre d’exploration pour la maximisation de la valeur de QED. Une arête représente la mutation liant deux solutions. Toutes les solutions qui ont fait partie de la population sont représentées. Le point de départ (méthane) est indiqué à l’aide d’une flèche. Les solutions sont colorées selon leur valeur de QED.

mesure que de meilleures solutions sont obtenues.

L’arbre d’exploration permet une certaine interprétabilité de la recherche effectuée par notre algorithme, en mettant en évidence l’intensification effectuée dans le voisinage des meilleures solutions connues à un moment donné de la recherche. Dans le Chapitre 3, nous utiliserons cette représentation pour étudier l’effet de l’introduction d’une diversité chimique au sein de la population, ce qui favorise l’exploration de l’espace de recherche au détriment de l’intensification. Précisons que cette représentation est rendue possible par le fait qu’il existe entre deux solutions de la population un lien uniquement basé sur une mutation. Si un opérateur de recombinaison était utilisé, il deviendrait très compliqué de suivre les liens de parenté à l’échelle de la population.

### 2.3.2 Test sur le *benchmark* GuacaMol

Afin de comparer plus précisément notre algorithme avec les autres approches de la littérature, nous choisissons de l'évaluer selon le *benchmark* GuacaMol [BROWN et al. 2019]. Il s'agit d'un *benchmark* de référence de la littérature qui a été proposé pour uniformiser l'évaluation des méthodes de génération moléculaire. En plus des méthodes évaluées lors de sa proposition, GuacaMol a été utilisé comme référence pour l'évaluation de nombreuses méthodes de génération moléculaire qui ont été proposées récemment. Deux ensembles de tâches d'optimisation sont proposés par GuacaMol. Le premier permet d'évaluer la capacité des méthodes de génération à reproduire la distribution d'un jeu de données de référence selon différentes métriques. Il s'agit d'un test conçu avant tout pour les méthodes de génération basées sur un modèle d'apprentissage, dont la phase d'apprentissage consiste à reproduire les solutions du jeu d'entraînement. Il serait difficile de l'utiliser pour un algorithme évolutionnaire, puisque la fonction objectif évalue des individus plutôt que l'ensemble de la population et qu'il ne serait pas trivial de formaliser une telle fonction pour ce problème. Par conséquent, nous étudierons plutôt le second ensemble de tâches d'optimisation permettant d'évaluer la capacité des méthodes à générer des solutions satisfaisant un but.

20 tâches d'optimisation sont définies. Trois tâches consistent à redécouvrir une molécule cible à l'aide d'une fonction de similarité. Trois autres tâches consistent à redécouvrir un ensemble de solutions similaires à une cible, à l'aide d'un seuil appliqué à la fonction de similarité. Deux autres tâches consistent à retrouver un ensemble d'isomères à partir d'une formule brute et d'une fonction évaluant la proximité avec la formule brute cible. Deux isomères sont deux graphes moléculaires différents possédant la même formule brute. Les douze tâches restantes combinent un ensemble d'objectifs, et sont définies comme une moyenne de plusieurs propriétés souvent contradictoires. Deux de ces tâches consistent à trouver une molécule « médiane », maximisant la similarité à deux molécules différentes. Sept autres tâches notées *MPO* (*multi-property objective*) consistent à trouver une solution similaire à un médicament connu, tout en faisant varier une propriété de ce médicament. Les trois dernières tâches combinent la nécessité de la présence ou de l'absence de structures moléculaires prédéfinies avec des objectifs de similarité ou de valeurs de propriété. La distance de Tanimoto sur des empreintes moléculaires est utilisée pour former les objectifs de similarité (voir la section 1.1.3 de ce mémoire). Une fonction gaussienne centrée sur la cible permet de définir les objectifs relatifs à l'obtention d'une valeur cible pour une propriété moléculaire donnée. Chaque tâche permet d'obtenir un

score entre 0 et 1.

Au sein de l'article présentant GuacaMol, plusieurs approches d'optimisation sont comparées [BROWN et al. 2019]. Il s'agit notamment de l'algorithme évolutionnaire GB-GA, ainsi que d'une approche basée sur un réseau de neurones récurrent pour la génération de molécules sous la forme de SMILES [SEGLER et al. 2018]. Cette méthode, nommée SMILES LSTM, consiste à entraîner le modèle de génération sur un jeu de données de référence, puis à effectuer itérativement un ré-entraînement du modèle sur les solutions générées correspondant le plus aux valeurs de propriété cible. Cela peut être vu comme une approche d'apprentissage par transfert mais également comme un algorithme de *hill-climbing* appliqué à un modèle d'apprentissage profond génératif. Nous nous intéressons également aux résultats publiés ultérieurement pour deux métaheuristiques définies pour l'optimisation de propriétés moléculaires, qui ont également été évaluées sur ce *benchmark*. Il s'agit d'abord de l'algorithme MSO que nous avons déjà étudié dans ce chapitre pour l'optimisation des valeurs de QED et de plogP [WINTER et al. 2019]. Pour rappel, MSO est un algorithme d'optimisation par essaim particulaire. Il s'agit également de la méthode MolFinder, qui suit un algorithme de type CSA (*conformational space annealing*) [KWON et LEE 2021]. Il s'agit d'un algorithme proche d'un algorithme évolutionnaire mais qui intègre également des mécanismes permettant de contrôler le taux d'exploration de l'espace de recherche selon une mesure de la diversité au sein de la population. En section 1.3 de ce mémoire, nous avons présenté ces algorithmes de manière plus précise.

**Conditions expérimentales** GuacaMol fournit un jeu de données de référence basé sur ChEMBL, qui est utilisé comme jeu de données d'entraînement pour les méthodes dépendant d'un modèle d'apprentissage ou comme base pour la population initiale des algorithmes évolutionnaires évalués. Nous suivons la même méthodologie que GB-GA, l'algorithme évolutionnaire évalué dans l'article présentant GuacaMol, c'est-à-dire que pour chaque tâche la population initiale est composée des 100 molécules du jeu de données possédant les scores les plus élevés. Nous effectuons une expérience de référence utilisant les mêmes paramètres que pour les expériences précédentes basées sur une population de taille 1000. Pour rappel, le paramètre de profondeur maximale de mutation est fixé à 2. Cela signifie qu'une mutation consiste à appliquer aléatoirement une ou deux actions sur le graphe moléculaire. Le nombre d'individus remplacés à chaque étape (*TailleLot*) est fixé à 10. Au plus 50 tentatives (*EssaisMax*) sont effectuées pour la recherche d'un améliorant. La procédure d'optimisation est stoppée après 1500 étapes. Nous considérons

ici l'espace de recherche des graphes moléculaires contenant jusqu'à 50 atomes lourds parmi {C, N, O, F, P, S, Cl, Br}. Dans les résultats, cette expérience sera nommée « Tous op. mut ». Nous effectuons deux expériences supplémentaires. La première a pour but d'étudier l'impact des actions secondaires sur le graphe moléculaire, et consiste à limiter l'opérateur de mutation aux actions primaires. La seconde a pour but d'étudier l'apport de la connaissance apportée par la population initiale, et consiste à utiliser en tant que population initiale la molécule de méthane uniquement. Pour ces expériences, les paramètres sont laissés à l'identique, à l'exception de (i) la mutation qui correspond à l'application successive de jusqu'à 3 opérateurs afin d'améliorer la capacité à échapper aux minimums locaux, et (ii) le nombre d'étapes d'optimisation qui est porté à 3000 puisqu'il est nécessaire d'appliquer plus de mutations pour arriver au même état. Toutes les expériences sont effectuées 10 fois.



Tâche	État de l'art				EvoMol			
	SMILES LSTM	GB-GA	CReM	MSO	MolFinder	Tous op. mut. op. mut. primaires	Op. mut. methane	Init. meilleure exécution
Celecoxib redisc.	1.000	1.000	1.000	1.000	1.000	0.978	0.714	0.923
Troglitazone redisc.	1.000	1.000	1.000	1.000	1.000	1.000	0.936	0.676
Thiofixene redisc..	1.000	1.000	1.000	1.000	1.000	0.876	0.852	0.695
Aripiprazole sim.	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.964
Albuterol sim.	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.878
Mestranol sim.	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
C <sub>11</sub> H <sub>24</sub>	0.993	0.971	0.966	0.997	1.000	1.000	1.000	1.000
C <sub>9</sub> H <sub>10</sub> N <sub>2</sub> O <sub>2</sub> PF <sub>2</sub> Cl	0.879	0.982	0.940	1.000	1.000	0.998	1.000	1.000
Median molecules 1	0.438	0.406	0.371	0.437	0.412	<b>0.455</b>	0.446	<b>0.455</b>
Median molecules 2	0.422	0.432	0.434	0.395	<b>0.454</b>	0.417	0.411	0.286
Osimertinib MPO	0.907	0.953	<b>0.995</b>	0.966	0.945	0.955	0.959	0.911
Fexonadine MPO	0.959	0.998	<b>1.000</b>	<b>1.000</b>	0.999	<b>1.000</b>	0.966	0.981
Ranolazine MPO	0.855	0.920	<b>0.969</b>	0.931	0.947	0.966	0.943	0.967
Perindopril MPO	0.808	0.792	0.815	0.834	0.816	<b>0.845</b>	0.809	0.789
Amlodipine MPO	0.894	0.894	0.902	0.900	<b>0.924</b>	0.867	0.874	0.796
Sitagliptin MPO	0.545	0.891	0.763	0.868	<b>0.948</b>	0.915	0.943	0.946
Zaleplon MPO	0.669	0.754	0.770	0.764	0.695	0.791	0.791	<b>0.793</b>
Valsartan SMARTS	0.978	0.990	0.994	0.994	<b>0.999</b>	0.998	<b>0.999</b>	0.000
deco hop	0.996	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.607
scaffold hop	0.998	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.948	<b>1.000</b>	0.989	0.655
total	17.340	17.983	17.919	18.086	18.087	18.060	17.632	15.298
total (MPO)	5.637	6.202	6.214	6.263	6.274	6.339	6.286	6.160

TABLE 2.2 – Résultats sur le *benchmark* GuacaMol. Les résultats pour CReM, MSO et MolFinder sont reportés depuis leurs articles respectifs. Les résultats dans les trois premières colonnes *EvoMol* correspondent au score moyen obtenu sur 10 exécutions dans différentes conditions expérimentales. La colonne *Meilleure exécution* représente les scores obtenus lors de l'exécution possédant le meilleur score total. La coloration permet de mettre en évidence les groupes de tâches.

Tout d’abord, on peut observer que le score total de notre expérience de référence (colonne *Tous op. mut.*) est supérieur à celui de SMILES LSTM, GB-GA et CReM, et est très proche des scores de MSO et MolFinder. Cela montre que notre méthode est très compétitive avec l’état de l’art. La différence principale entre EvoMol et GB-GA est l’utilisation d’un opérateur de recombinaison au sein de GB-GA. Les résultats indiquent ici que l’opérateur de recombinaison n’est pas indispensable pour les tâches évaluées. De plus, la colonne *Meilleure exécution* montre que des résultats encore supérieurs sont susceptibles d’être obtenus. Par ailleurs, il est intéressant de remarquer que notre méthode est généralement plus performante que les autres approches sur les problèmes d’isomérisation (lignes 7 et 8). Il s’agit fondamentalement de problèmes d’intensification, pour lesquels notre approche est très efficace grâce à la localité des opérateurs de mutation. Notre approche est également la plus efficace pour les problèmes MPO, selon la dernière ligne du tableau qui représente les totaux obtenus pour ce groupe de tâches. Il s’agit de problèmes définis comme un assemblage de propriétés contradictoires, dont les solutions sont nécessairement inconnues contrairement aux tâches de redécouverte et de similarité. L’efficacité d’EvoMol pour ces problèmes peut s’expliquer à nouveau par l’utilisation d’opérateurs de mutation proches du niveau atomique, mais également par la limitation des hypothèses sur l’espace de recherche. Ces caractéristiques permettent une grande liberté d’exploration de l’espace de recherche.

La Table 2.2 nous permet également d’étudier l’influence de la connaissance intégrée sous la forme de la population initiale. L’expérience avec méthane comme seul individu de départ obtient un score total inférieur, bien que les scores ne soient pas impactés pour certaines tâches. Il s’agit notamment du cas des tâches 7 et 8 qui correspondent à la recherche d’isomères, et pour lesquelles le score obtenu est 1. Il est intéressant de noter qu’un score nul est obtenu pour la tâche *Valsartan SMARTS*. Cela s’explique par le fait que l’objectif est défini comme une moyenne géométrique (basée sur le produit des valeurs et non la somme) dont un des termes est un objectif binaire représentant la présence d’une structure moléculaire. Cette structure n’étant pas présente dans la population initiale, le terme correspondant à sa présence est nul. Par conséquent, la valeur de fonction objectif est nulle également. Cette structure est très peu susceptible d’apparaître car elle doit être trouvée par exploration aléatoire de l’espace de recherche. Cela souligne l’importance du soin qui doit être apporté à la définition de la fonction objectif, qui doit permettre de guider la recherche vers des solutions pertinentes. La Figure 2.5 représentant la distribution des scores obtenus pour les tâches de MPO montre que l’expérience utilisant le méthane

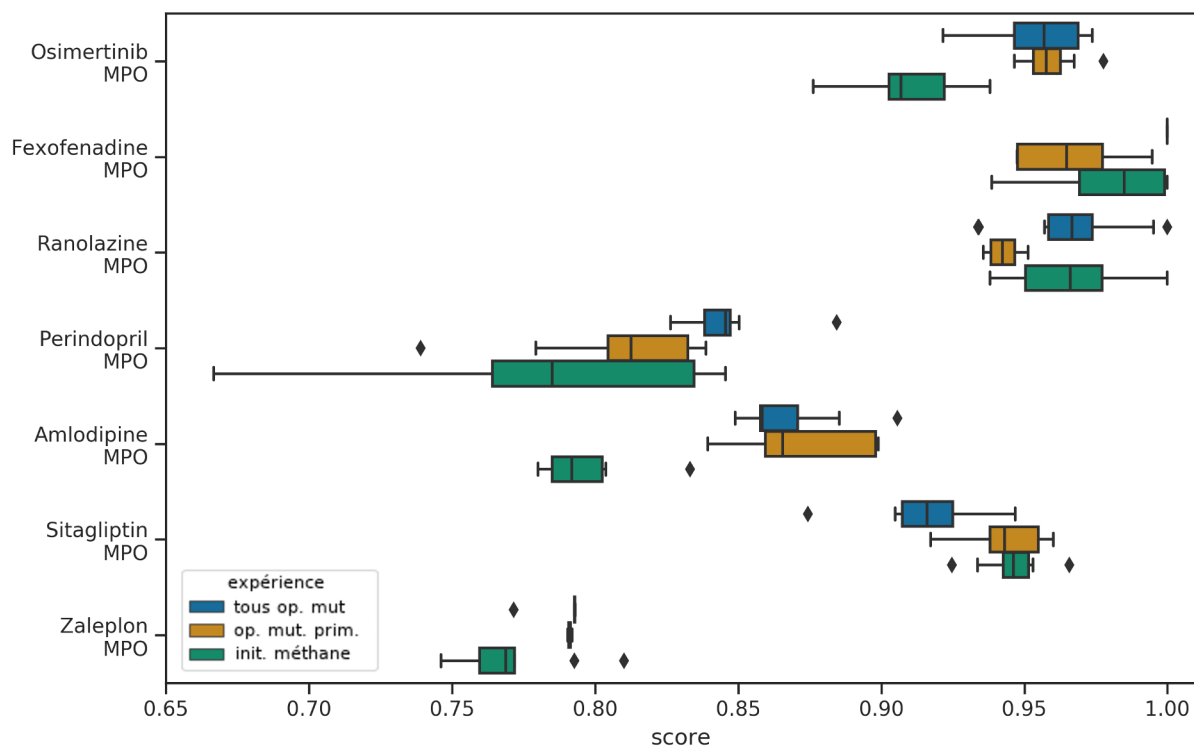


FIGURE 2.5 – Diagramme en boîte de la distribution des scores obtenus par EvoMol sur les tâches MPO du *benchmark* GuacaMol. Pour chaque tâche, la première ligne correspond à l'expérience de référence utilisant tous les opérateurs de mutation (bleu), la seconde ligne correspond à l'expérience n'utilisant que les opérateurs de mutation primaires (orange), et la troisième ligne correspond à l'expérience utilisant le méthane en tant que population initiale (vert).

comme population initiale est celle causant la plus grande variabilité des résultats. Nous pouvons toutefois noter que des scores comparables aux autres expériences sont obtenus pour une grande partie des tâches. Cela laisse penser que l'absence de population initiale n'empêche pas strictement l'obtention de bonnes solutions mais rend simplement ces problèmes plus difficiles, dans le sens que les chemins dans l'espace des graphes moléculaires composés de solutions améliorantes sont plus longs et sont rares. La distribution des scores pour l'ensemble des propriétés est visible en Annexe A.1.

Les résultats de l'expérience utilisant seulement les opérateurs de mutation primaires (Table 2.2) montrent que les performances sont moins impactées que par l'absence de population initiale. Les opérateurs de mutation secondaires ne permettent pas d'accéder à des zones de l'espace moléculaire supplémentaires qui ne seraient pas accessibles par

Tâche	Tous op. mut.		Op. mut. primaires		Init. methane	
	Succès	ERT ( $\times 10^6$ )	Succès	ERT ( $\times 10^6$ )	Succès	ERT ( $\times 10^6$ )
Celecoxib	9	0.7	-	-	8	3.9
Troglitazone	9	0.4	6	16.3	1	36.9
Thiotixene	4	3.6	3	60.4	-	-

TABLE 2.3 – Impact des conditions expérimentales sur l’efficacité de l’optimisation pour les tâches de redécouverte. La colonne succès correspond au nombre d’exécutions qui ont permis d’obtenir exactement la molécule recherchée (sur un total de 10). La colonne ERT correspond à l’espérance du coût de l’exécution en nombre d’appels à la fonction objectif pour obtenir la valeur cible de fonction objectif (1.0). Les tirets correspondent aux valeurs non définies d’ERT (absence de succès).

les opérateurs de mutation primaires. Cependant, ils définissent des sauts plus complexes dans l’espace de recherche. Pour l’optimisation des tâches de GuacaMol, les fonctions objectif semblent dans l’ensemble suffisamment bien définies pour donner la possibilité de guider itérativement vers ces transformations plus complexes à partir d’opérateurs élémentaires. Les pertes de score les plus importantes causées par l’absence des opérateurs secondaires concernent les tâches de redécouverte. La Table 2.3 représente les valeurs d’espérance du coût de l’exécution en nombre d’appels à la fonction objectif pour les tâches de redécouverte. Pour rappel, il s’agit d’une mesure définie dans la section 1.3.1 qui permet de représenter l’efficacité de la recherche pour obtenir une valeur cible de fonction objectif. La valeur cible est ici fixée à la valeur maximale 1 qui correspond à la redécouverte de la molécule. Les résultats présentés dans cette table mettent en évidence que pour ces tâches, l’utilisation de l’ensemble des opérateurs est très bénéfique puisque cela permet de trouver la molécule cible plus régulièrement et avec une valeur d’ERT nettement plus faible. On observe même que les opérateurs secondaires semblent indispensables pour obtenir la molécule de celecoxib. Pour rappel, cela ne signifierait pas qu’il existe des structures moléculaires que les opérateurs primaires ne permettent pas d’atteindre en général, mais plutôt qu’il n’existe pas de chemins de solutions améliorantes vers la molécule de celecoxib en utilisant uniquement des opérateurs primaires. Pour les expériences dans lesquelles le méthane est utilisé en tant que population initiale, on observe également un effet négatif sur l’efficacité de la recherche. La population initiale tirée de ChEMBL semble même indispensable pour obtenir la molécule de thiotixene. Ces résultats montrent que le choix des opérateurs de mutation et les structures moléculaires présentes à l’état initial ont une grande influence sur la capacité de la procédure d’optimisation à maximiser l’objectif de similarité. La valeur optimale de l’objectif de similarité forme un îlot qui peut être difficile

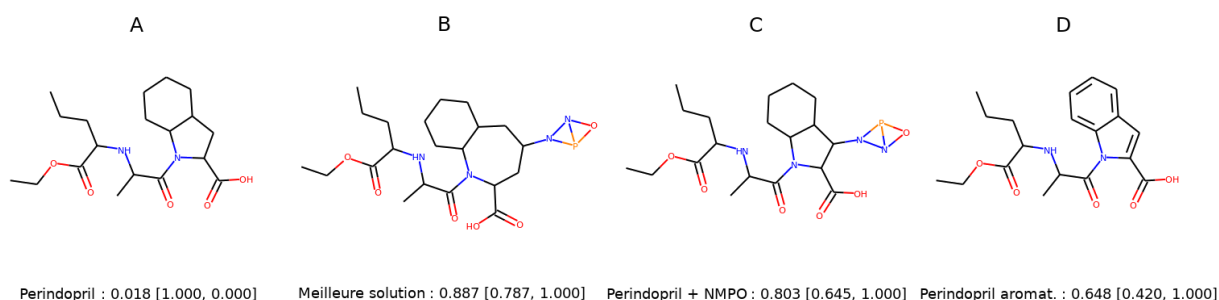


FIGURE 2.6 – Étude de la tâche Perindopril MPO. A : Molécule de perindopril. B : Meilleure solution obtenue par EvoMol. C et D : solutions hypothétiques (perindopril avec double cycle aromatique NNPO et variante doublement aromatique du perindopril). Scores reportés : score tâche Perindopril MPO [score de similarité, score sur le nombre de cycles aromatiques]

voire impossible d’atteindre sans des opérateurs de mutation adaptés ou des structures moléculaires déjà présentes dans la population initiale.

Nous décidons d’étudier le comportement de l’objectif de similarité à travers plusieurs solutions à la tâche Perindopril MPO. Celle-ci consiste à maximiser un score formé comme la moyenne géométrique de la similarité à la molécule de perindopril et d’une fonction Gaussienne appliquée sur le nombre de cycles aromatiques et centrée sur 2 (avec une variance de 0.5). Pour rappel, l’aromaticité est une propriété de certains cycles qui peut être détectée simplement par la règle de Hückel mais dont la définition exacte fait débat (voir la section 1.1.1). Ainsi, on cherche une solution très similaire au perindopril possédant 2 cycles aromatiques (le perindopril n’en possède aucun). La Figure 2.6 montre plusieurs solutions à ce problème et leurs scores. Tout d’abord, la molécule de perindopril (A) est comme attendu une solution médiocre car elle ne possède aucun cycle aromatique, bien qu’elle soit évidemment parfaitement similaire à elle-même. La molécule B, issue de l’optimisation par EvoMol possède le meilleur score que l’on connaisse. Il est intéressant de remarquer que l’aromaticité n’a pas été intégrée au sein des deux cycles déjà existants mais dans un nouveau groupe de 4 atomes (à droite), qui permet en fait de concentrer cette propriété en affectant peu les environnements chimiques dans le reste de la molécule. Ce groupe est considéré comme contenant deux cycles aromatiques selon la règle de Hückel. Notons cependant que des chimistes ne le considéreraient certainement pas de cette façon, tant cette combinaison et concentration d’hétéroatomes est éloignée de la chimie connue. Il est également intéressant de remarquer que pour maximiser le score de similarité, des

atomes de carbone ont été ajoutés au cycle contenant un atome d'azote. Cela permet de maximiser les environnements chimiques locaux qui ne changent pas. En effet, la solution C qui pourrait être considérée intuitivement comme plus proche du perindopril possède en réalité un score de similarité inférieur. Finalement, la solution D correspond à la façon la plus intuitive de résoudre le problème posé, c'est-à-dire en transformant les deux cycles du perindopril en cycles aromatiques. Il est d'ailleurs probable qu'il s'agisse de la solution imaginée par les créateurs de GuacaMol en définissant ce problème. Or, le score de similarité est relativement faible. Cette étude montre qu'une fonction objectif basée sur un score de similarité défini à partir de caractéristiques ECFP est susceptible de provoquer des comportements peu intuitifs. D'autre part, cela montre que des méthodes d'optimisation peuvent exploiter des failles difficiles à anticiper dans la définition des propriétés, comme les groupes d'hétéroatomes aromatiques.

### 2.3.3 Optimisation de propriétés électroniques coûteuses

Nous souhaitons maintenant étudier notre approche pour l'optimisation de propriétés du domaine de la chimie des matériaux moléculaires organiques. Nous choisissons d'étudier l'optimisation des énergies HOMO et LUMO, dont l'estimation dépend de calculs coûteux en DFT (voir la section 1.4 de ce mémoire). Nous étudions ici ces deux propriétés séparément, dans le cadre d'une preuve de concept de l'utilisation de notre algorithme évolutionnaire pour l'optimisation de propriétés électroniques coûteuses. Nous menons une maximisation de l'énergie HOMO et une minimisation de l'énergie LUMO.

**Expériences** Pour ces expériences, nous considérons l'espace de recherche des graphes moléculaires contenant jusqu'à 9 atomes lourds parmi {C, N, O, F}. Cet espace de recherche est plus restreint que pour les expériences effectuées précédemment, car cela permet de limiter le coût de calcul des simulations DFT (voir la section 1.1 de ce mémoire). Les expériences sont menées selon les mêmes paramètres que pour l'optimisation des valeurs de QED, à l'exception de la taille de la population (*TaillePopMax*) qui est fixée à 20 et le nombre d'individus remplacés à chaque étape (*TailleLot*) qui est fixé à 2. Comme les expériences précédentes, la molécule de méthane est utilisée comme point de départ de la recherche évolutionnaire, le paramètre de profondeur maximale de mutation est fixé à 2 et au plus 50 tentatives (*EssaisMax*) sont effectuées pour la recherche d'un améliorant. La géométrie initiale des optimisations DFT est obtenue par optimisation en mécanique moléculaire selon l'implémentation du champ de force MMFF94 du programme OpenBabel

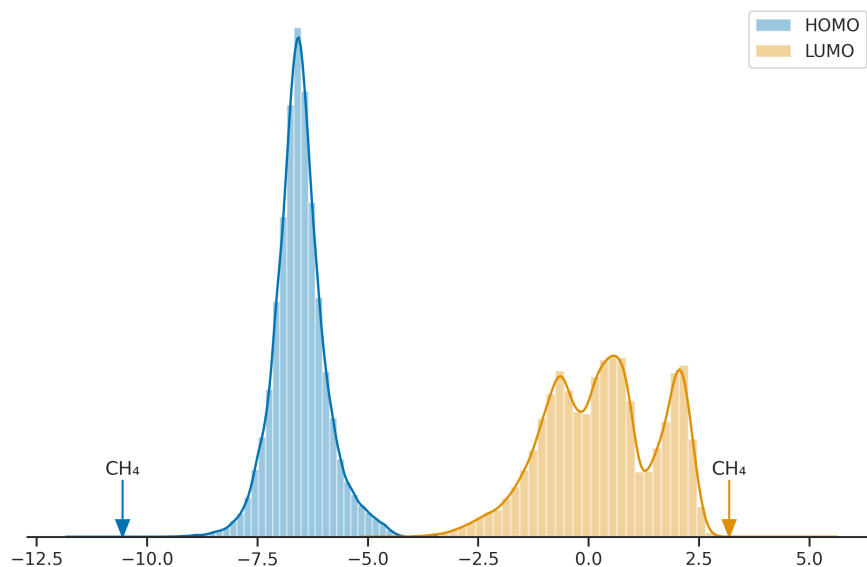


FIGURE 2.7 – Distribution des énergies HOMO et LUMO en eV dans le jeu de données QM9. Les valeurs de propriétés du méthane ( $\text{CH}_4$ ) sont indiquées par deux flèches.

[YOSHIKAWA et HUTCHISON 2019].

Afin d’obtenir des valeurs de propriétés cibles de référence, nous étudions les valeurs d’énergies HOMO et LUMO dans le jeu de données QM9. Pour rappel, QM9 correspond à une énumération partielle de l’espace moléculaire des molécules de petites tailles. Les molécules de QM9 correspondent au même espace de recherche que nos expériences d’optimisation, c’est-à-dire les molécules contenant jusqu’à 9 atomes lourds parmi  $\{\text{C}, \text{N}, \text{O}, \text{F}\}$ . Les distributions des valeurs d’énergies HOMO et LUMO sont représentées en Figure 2.7. On y observe que les valeurs d’énergie HOMO sont relativement resserrées au sein d’une distribution approximativement gaussienne, centrée autour de -7 eV. La distribution des valeurs d’énergie LUMO est plus étendue et plus complexe. Ses valeurs sont principalement comprises entre -2.5 eV et 2.5 eV. Il est intéressant de noter que la molécule de méthane possède des valeurs particulièrement éloignées de notre objectif pour les deux propriétés : une énergie HOMO (que l’on cherche à maximiser) très faible et une énergie LUMO (que l’on cherche à minimiser) très élevée. Il s’agit donc d’une mauvaise solution pour les deux problèmes d’optimisation, ce qui en fait un point de départ intéressant de la procédure d’optimisation.

Nous avons observé lors de l’optimisation des valeurs de QED que notre algorithme est susceptible de produire des solutions peu réalistes, possédant notamment des cycles de

petites tailles composés principalement d'hétéroatomes. Nous proposons ici d'intégrer au sein de la fonction objectif un terme conçu pour favoriser le réalisme des solutions. Nous utilisons ici le CLScore [BÜHLMANN et REYMOND 2020], qui évalue la proximité d'une molécule avec le jeu de données ChEMBL (voir la section 1.4 également). Le CLScore possède des valeurs élevées pour les molécules dont les sous-graphes moléculaires (*shingles*) sont très représentés dans ChEMBL. ChEMBL contient des molécules que l'on sait synthétisables. Pour cette raison, il est raisonnable d'espérer que des molécules semblables à ChEMBL soient susceptibles d'être synthétisées. Précisons toutefois que ChEMBL est un jeu de données plutôt orienté pour la chimie pharmaceutique. Chercher à maximiser le CLScore est donc susceptible de pousser la recherche vers un sous-ensemble de l'espace moléculaire incompatible avec l'optimisation de propriétés électroniques. Nous choisissons donc de définir un seuil sur les valeurs de CLScore, plutôt que de chercher à les maximiser. Pour cela nous utilisons une fonction sigmoïde qui est centrée sur la valeur 1.5 grâce à une transformation linéaire de la valeur de CLScore, avec un paramètre de largeur  $\lambda$  fixé à 10 (voir l'équation 2.1). La valeur de  $\lambda$  élevée permet de resserrer la zone centrale de la fonction sigmoïde. Cela permet de bloquer les valeurs de CLScore inférieures à 1 et de pénaliser les valeurs comprises entre 1 et 2. Ce paramétrage de la fonction sigmoïde implique que la valeur 0.999 est atteinte quand la valeur de CLScore est au moins égale à 2.19. Pour rappel, les auteurs du CLScore suggèrent un seuil à une valeur de 3.3. Nous fixons le nôtre à une valeur inférieure afin d'effectuer un compromis entre réalisme des solutions et espace de recherche accessible pour l'optimisation des énergies HOMO et LUMO.

Nous combinons l'objectif de propriété électronique avec l'objectif de réalisme comme un produit de fonctions sigmoïdes, afin d'obtenir un score compris entre 0 et 1. La fonction sigmoïde guidant la recherche pour la minimisation (resp. maximisation) de l'énergie LUMO (resp. HOMO) est centrée au milieu de la distribution, c'est-à-dire à 0 eV (resp. -7 eV). Les deux fonctions sont définies en équation (2.1). Notons que l'opposé de la valeur d'énergie LUMO est considéré afin de changer le sens de la fonction sigmoïde. La valeur du paramètre  $\lambda$  fixée à 1 permet ici d'obtenir une variation plus lente des valeurs de la fonction sigmoïde afin de couvrir l'étendue des valeurs d'énergie HOMO et LUMO.

Pour résumer, nous effectuons deux expériences d'optimisation pour chacune des deux propriétés. La première consiste à optimiser directement la valeur de propriété (minimisation de l'énergie LUMO et maximisation de l'énergie HOMO). La deuxième expérience consiste à maximiser la fonction objectif qui agrège les objectifs de propriété électronique et de réalisme des solutions, c'est-à-dire  $f_{\text{LUMO}}(x) \times f_{\text{CLScore}}(x)$  et  $f_{\text{HOMO}}(x) \times f_{\text{CLScore}}(x)$ .



Contrairement aux expériences pour l’optimisation des valeurs de QED et des tâches du *benchmark* GuacaMol, ces quatre expériences sont effectuées une seule fois en raison de leur coût.

$$\begin{aligned} f_{\text{CLscore}}(x) &= \frac{1}{1 + e^{-10(\text{CLScore}(x)+1.5)}} \\ f_{\text{LUMO}}(x) &= \frac{1}{1 + e^{-1(-\text{LUMO}(x))}} \\ f_{\text{HOMO}}(x) &= \frac{1}{1 + e^{-1(\text{HOMO}(x)-7)}} \end{aligned} \tag{2.1}$$

**Résultats** La Figure 2.8 représente les meilleures solutions contenues dans QM9 (A) et obtenues par notre approche (B, C) pour la minimisation de l’énergie LUMO. Nous observons d’abord que les solutions obtenues sans objectif de réalisme (B) semblent extrêmement peu stables et peu réalistes, puisqu’elles sont composées exclusivement d’hétéroatomes. Notons par ailleurs que leurs valeurs de CLScore sont nulles, ce qui indique que les sous-graphes moléculaires n’existent pas dans ChEMBL ou sont très rares. L’objectif de réalisme semble être relativement efficace (C), puisque les solutions semblent plus crédibles bien que possédant une géométrie contrainte, donc a priori peu stable. Nous pouvons en particulier noter que la deuxième molécule de la ligne C (le cyclopropanetrione) a déjà été détectée expérimentalement [XIN et al. 2019]. Il est donc intéressant de remarquer que notre expérience très simple a mené vers l’obtention d’une cible connue en chimie des matériaux moléculaires organiques. Finalement, nous observons que les meilleurs résultats sont obtenus par l’expérience basée sur l’objectif de réalisme. Cela signifie que notre approche a pu permettre d’obtenir de meilleures solutions que le jeu de données QM9, qui est défini sur un sous-ensemble du même espace de recherche. Ces résultats ont été obtenus par voisinage successif à partir du méthane, une molécule très simple et possédant une énergie LUMO très élevée. Il est intéressant de noter que l’utilisation de l’objectif de réalisme a permis d’éviter une zone de l’espace de recherche qui semble contenir un minimum local et dont il semble difficile de sortir pour l’expérience de la ligne B.

Pour la maximisation de l’énergie HOMO, les résultats sont présentés en Figure 2.9. On observe que les trois premières solutions issues de QM9 (A) présentent des structures originales qui semblent a priori peu réalistes. Il s’agit en fait de dérivés de fenestranes, une classe de molécules connues pour leur instabilité [VENEPALLI et AGOSTA 1987]. EvoMol permet d’obtenir des solutions possédant des valeurs d’énergie HOMO très élevées

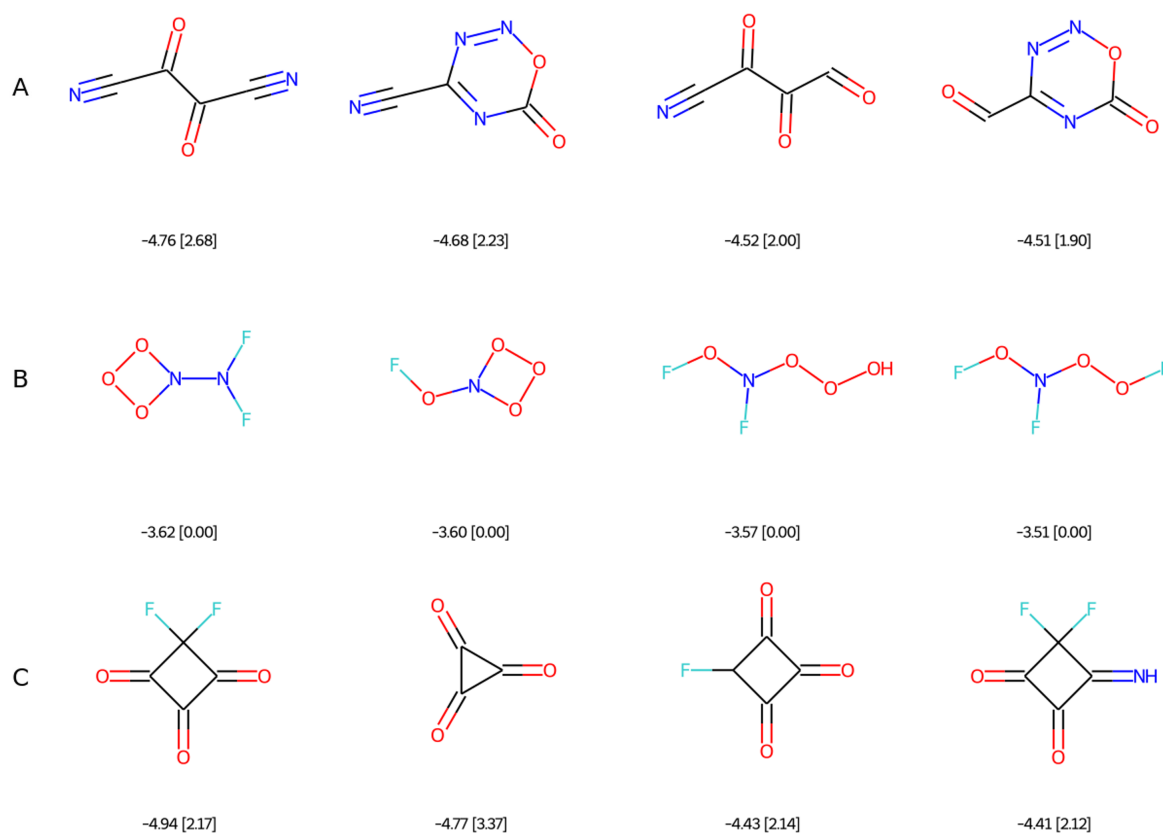


FIGURE 2.8 – Meilleures solutions pour la minimisation de l'énergie LUMO. A : Dans le jeu de données QM9, B : Par optimisation évolutionnaire, C : Par optimisation évolutionnaire avec objectif de réalisme. Les valeurs de propriétés (LUMO en eV [CLScore]) sont indiquées sous chaque molécule.

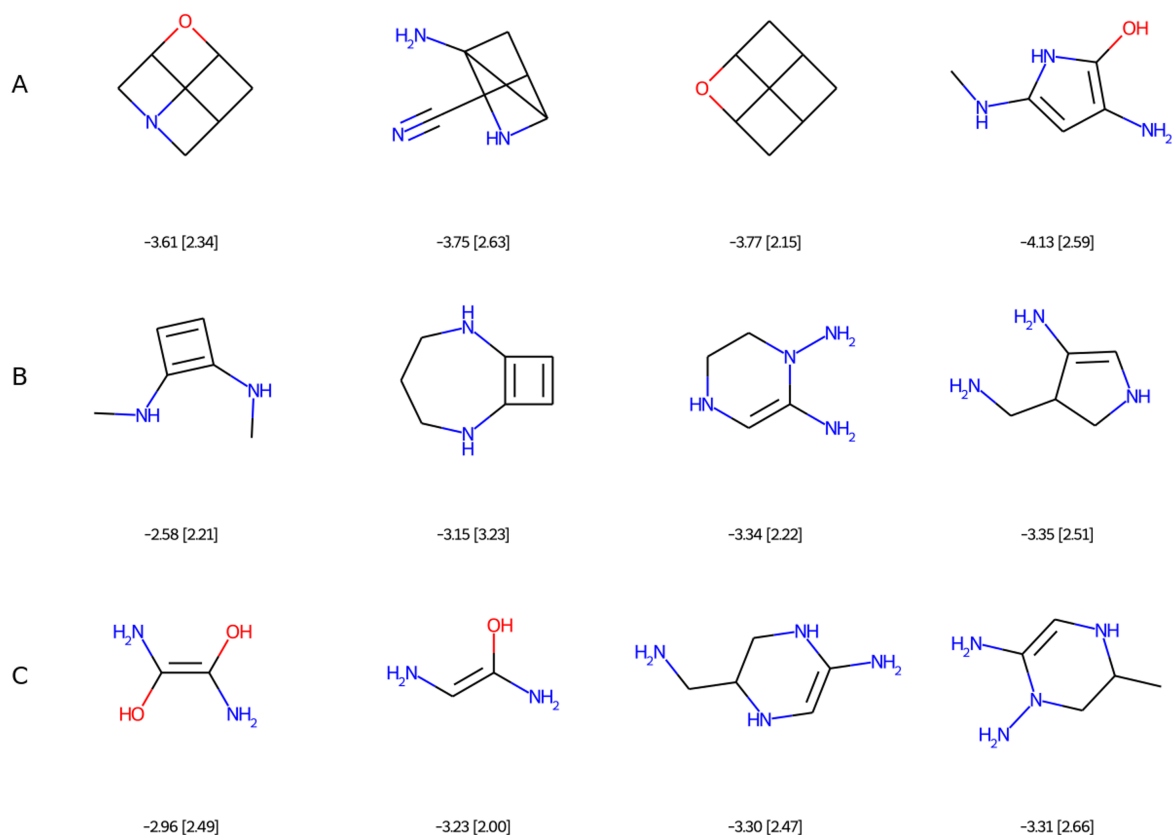


FIGURE 2.9 – Meilleures solutions pour la maximisation de l'énergie HOMO. A : Dans le jeu de données QM9, B : Par optimisation évolutionnaire, C : Par optimisation évolutionnaire avec objectif de réalisme. Les valeurs de propriétés (HOMO en eV [CLScore]) sont indiquées sous chaque molécule.

(B, C), plus élevées que les solutions contenues dans QM9. Comme il peut être attendu chimiquement, ces solutions sont des dérivés d'amines.

Pour résumer, nos expériences montrent que notre algorithme peut effectivement être utilisé pour l'optimisation de propriétés électroniques. Il permet d'obtenir de meilleures solutions que les molécules de QM9, un jeu de données de référence pour l'apprentissage de propriétés électroniques, dont l'apprentissage des énergies HOMO et LUMO [GLAVATSKIKH et al. 2019].

**Expérience complémentaire** Nous effectuons maintenant une expérience complémentaire, afin d'évaluer notre approche sur un problème plus réaliste pour la recherche de

cibles en chimie des matériaux moléculaires organiques, et de démontrer sa flexibilité et son interprétabilité. Nous supposons que le cœur d'une molécule prometteuse a été identifié par une personne experte du domaine. Nous cherchons à déterminer le branchement d'atomes sur ce cœur qui permettra de répondre au mieux au problème d'optimisation. Il s'agit d'un scénario représentatif de la façon dont travaillent les chimistes du domaine des matériaux moléculaires organiques, qui sont amenés régulièrement à utiliser leur expertise pour améliorer par voisinage une molécule connue.

Nous basons notre expérience sur un cœur furane, une molécule aromatique composée de quatre atomes de carbone, d'un atome d'oxygène et de quatre atomes d'hydrogène implicites. Cette molécule est similaire à des molécules utilisées en pratique dans le domaine des matériaux moléculaires organiques, et en particulier à la molécule de thiophène qui correspond à la molécule de furane dans laquelle l'atome d'oxygène est substitué par un atome de soufre. Nous choisissons l'étude du furane et non du thiophène car son coût de calcul est inférieur, l'atome de soufre contenant 8 électrons supplémentaires par rapport à l'atome d'oxygène. La molécule de furane est utilisée comme unique individu de la population initiale. Ses atomes sont marqués comme non mutables, et nous appliquons les contraintes de gel des atomes non mutables décrites en section 2.2.2. Ainsi, les seules mutations possibles correspondent au branchement d'atomes sur la molécule de furane. Nous menons une minimisation de l'énergie LUMO dans ce contexte, en utilisant des paramètres identiques aux expériences précédentes à l'exception du nombre d'étapes d'optimisation qui est restreint à 20 afin de faciliter la visualisation graphique des résultats. Pour rappel, la taille de la population (*TaillePopMax*) est fixée à 20, le nombre d'individus remplacés à chaque étape (*TailleLot*) est fixé à 2, le paramètre de profondeur maximale de mutation est fixé à 2 et au plus 50 tentatives (*EssaisMax*) sont effectuées pour la recherche d'un améliorant. Nous considérons à nouveau l'espace de recherche des graphes moléculaires contenant jusqu'à 9 atomes lourds parmi {C, N, O, F}. Puisque la molécule de furane contient 4 atomes de carbone et un atome d'oxygène, jusqu'à 4 atomes explicites peuvent être ajoutés pour former les solutions. Nous utilisons l'objectif basé sur deux fonctions sigmoïdes combinant l'objectif d'énergie LUMO et l'objectif de réalisme basé sur la valeur de CLScore.

Les résultats sont présentés en Figure 2.10, qui correspond à l'arbre d'exploration de l'espace de recherche. Chaque nœud de l'arbre est une solution intégrée dans la population. Chaque arête lie deux solutions liées par une mutation, et est évaluée par le type de l'action ou des actions sur le graphe moléculaire qui la composent. On observe que cette expérience

permet d'obtenir des solutions très réalistes, qui contiennent des groupes chimiques attendus par rapport au problème d'optimisation. Il s'agit en particulier des groupes peroxy (deux atomes d'oxygène consécutifs) et des groupes fluorure d'oxygène (atome de fluor lié à un atome d'oxygène). Il est intéressant de remarquer que cette visualisation permet d'identifier les groupes de solutions n'ayant pas donné lieu à des améliorations ou ayant donné lieu à des améliorations à l'échelle de la population qui n'étaient pas suffisamment importantes et qui ont donc été abandonnées par la recherche. C'est le cas par exemple des groupes amines (un atome d'azote connecté à des atomes de carbone et d'hydrogène) en haut à gauche de l'arbre. Nous menons également une expérience équivalente pour la maximisation de l'énergie HOMO, dont l'arbre d'exploration est présenté en Annexe A.3.

La visualisation sous la forme d'un arbre d'exploration permet aux utilisateurs du domaine d'application d'étudier le chemin parcouru dans l'espace de recherche pour obtenir les solutions résultantes. De façon au moins aussi intéressante, cela permet d'identifier les caractéristiques qui ont été testées et qui n'ont pas été retenues par la procédure d'optimisation car elles ne permettaient pas une amélioration suffisante de la fonction objectif. Cela permet finalement aux utilisateurs d'étudier la relation entre structures et propriétés moléculaires. Il s'agit en réalité d'un enjeu de recherche à part entière dans le domaine d'application, car cela permet de former l'expertise des chimistes sur l'effet des fonctions chimiques sur les propriétés électroniques en fonction du contexte.



## 2.4 Génération de solutions réalistes

Nous avons remarqué lorsque nous avons étudié l’optimisation des valeurs de QED et d’énergie LUMO en particulier que notre approche est susceptible de générer des solutions peu réalistes. Il s’agit d’une caractéristique connue des algorithmes évolutionnaires pour l’optimisation moléculaire. Dans la section précédente, nous avons proposé une première approche pour améliorer le réalisme des solutions générées, qui consiste à intégrer un terme au sein de la fonction objectif pour favoriser la similarité à un jeu de molécules synthétisables. Dans cette section, nous étudions de façon plus approfondie le problème de la génération de solutions réalistes dans le cadre de l’optimisation de propriétés moléculaires. D’abord, nous mettons en évidence des limites de l’approche qui consiste à intégrer une métrique estimant la synthétisabilité des solutions au sein d’une fonction objectif. Ensuite, nous proposons une approche basée sur la définition de contraintes binaires sur l’espace de recherche, et nous étudions plusieurs contraintes possibles.

Durant les quinze dernières années, plusieurs métriques ont été proposées pour estimer quantitativement l’accessibilité synthétique des molécules (voir la section 1.4 de ce mémoire). Dans le cadre des travaux que nous menons ici, nous choisissons de nous baser sur les métriques de SAScore et de CLScore. Pour rappel, le SAScore est basé sur une mesure de similarité à des molécules de la base de données généraliste PubChem [ERTL et SCHUFFENHAUER 2009]. Le CLScore est l’approche que nous avons utilisée dans la section précédente, et est basé sur une mesure de similarité à des molécules de la base de données ChEMBL, orientée pour la chimie pharmaceutique.

### 2.4.1 Optimisation des métriques d’accessibilité synthétique

Le SAScore et le CLScore étant des métriques quantitatives, il est possible de les considérer comme un objectif d’optimisation comme nous l’avons fait dans la section précédente pour le CLScore. Nous menons une expérience qui consiste à optimiser ces deux métriques (minimisation du SAScore, maximisation du CLScore), dans les mêmes conditions que pour l’optimisation des valeurs de QED que nous avons effectuée en section 2.3.1, avec une population de taille 1000. Il s’agit d’un problème virtuel, qui consiste à rechercher les molécules « les plus facilement synthétisables » selon les métriques.

Les meilleures solutions obtenues sont représentées en Figure 2.11. Les deux métriques mènent vers l’obtention d’hydrocarbures. Il s’agit d’un résultat cohérent puisque l’atome de carbone est l’atome le plus représenté dans la chimie organique, et donc également dans

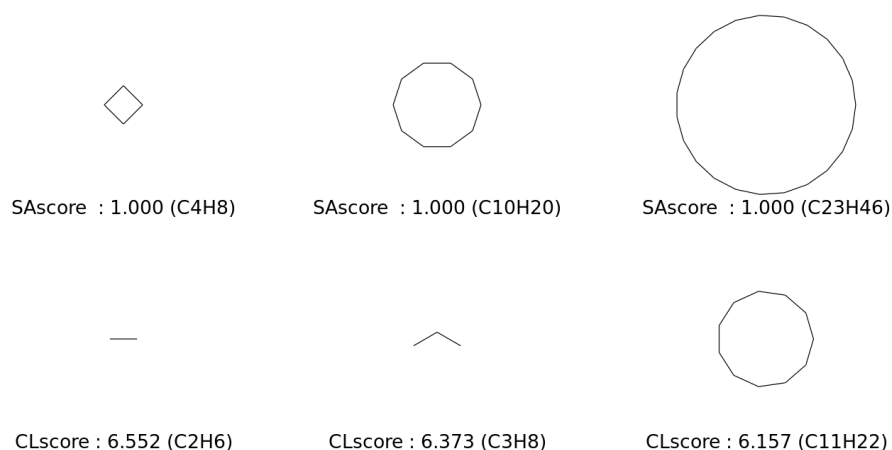


FIGURE 2.11 – Trois meilleures solutions obtenues pour l’optimisation du SAScore et du CLScore à l’aide d’EvoMol. La légende des solutions comprend la valeur de fonction cible ainsi que la formule brute.

les jeux de données sur lesquels sont basées ces mesures. En tant que mesures d’accessibilité synthétique, il s’agit également d’un résultat cohérent puisque les hydrocarbures sont considérés comme une matière première, dont le coût de synthèse est donc inexistant.

**Limites** Pour concevoir une fonction objectif prenant en compte le réalisme des solutions, nous pourrions associer l’objectif correspondant à la propriété moléculaire optimisée avec l’objectif quantitatif favorisant le réalisme, comme nous l’avons fait au sein de la section précédente pour l’optimisation de propriétés électroniques. Cependant, il est facile d’imaginer que dans de nombreux cas ces objectifs risquent d’être contradictoires, les optimums du second objectif étant des structures moléculaires très simples comme des hydrocarbures.

Il est possible comme nous l’avons fait précédemment d’appliquer une transformation sur l’objectif quantitatif de réalisme, basée par exemple sur une fonction sigmoïde afin d’appliquer un seuil qui favorise des valeurs « raisonnables » sans guider la recherche vers des valeurs « extrêmes ». Cela implique un réglage de la fonction de transformation qui nécessite une connaissance du comportement de l’objectif de réalisme, mais cela nécessite surtout de pouvoir pondérer numériquement l’objectif de propriété moléculaire et l’objectif de réalisme. Or, ce dernier point en particulier est une limite importante puisque ces deux objectifs ne représentent a priori pas des quantités comparables. La pondération sera donc dépendante de l’étendue de valeurs des différents objectifs et donc dépendante du problème d’optimisation. De plus, il n’existe a priori pas de procédure pour définir le



réglage, qui devra donc souvent être effectué par essais et erreurs.

Pour les expériences que nous avons réalisées dans la section précédente pour l’optimisation de propriétés électroniques, nous avons appliqué une transformation sur les deux objectifs afin qu’ils soient définis par une valeur comprise entre 0 et 1. Nous avons appliqué un produit entre ces deux quantités et nous avons ainsi considéré qu’elles étaient de même importance. Une alternative serait de considérer le problème comme un problème d’optimisation multi-objectif au sens strict, ce qui nécessiterait d’intégrer au sein d’EvoMol une procédure de sélection des améliorants basée sur plusieurs objectifs. Cette alternative laisserait aux utilisateurs chimistes le choix du meilleur compromis entre les deux objectifs à l’issue de l’optimisation. Pour la suite, nous choisirons cependant une autre approche basée sur des contraintes appliquées à l’espace de recherche.

## 2.4.2 Contraintes sur l’espace de recherche

Plutôt que de chercher un compromis entre la propriété moléculaire cible et une quantification numérique de l’accessibilité synthétique des molécules, nous pensons qu’il est raisonnable de chercher la meilleure valeur de propriété cible dans un espace de recherche restreint aux solutions réalistes. Cela revient à appliquer un filtre sur l’espace de recherche accessible, ce qui peut également être considéré comme une optimisation sous contrainte. Cela nécessite de pouvoir évaluer de façon binaire le réalisme des solutions. Cette classification peut être basée sur les mesures quantitatives d’accessibilité synthétique, mais nous allons voir qu’il est également possible de définir des estimateurs du réalisme ne dépendant pas d’une évaluation numérique. Un intérêt de cette approche est qu’elle permet d’extraire l’objectif de réalisme de la fonction objectif du problème d’optimisation. Cette dernière reste ainsi consacrée au guidage de la recherche vers des solutions possédant les propriétés moléculaires attendues.

### Contrainte de réalisme basée sur un score d’accessibilité synthétique

Nous proposons d’étudier trois types de contraintes sur l’espace de recherche. D’abord, nous définissons deux contraintes basées sur les estimateurs quantitatifs de l’accessibilité synthétique (SAScore et CLScore). Pour transformer ces mesures en contraintes binaires, nous appliquons un seuil qui sépare les valeurs qui qualifient les molécules considérées comme réalistes et les molécules considérées comme non réalistes. Pour le SAScore, nous choisissons le seuil proposé par [VORŠILÁK et al. 2020], à savoir que les molécules pos-

sédant une valeur de SAScore inférieure ou égale à 4.4 sont considérées réalistes. Pour le CLScore, nous utilisons le seuil proposé dans la publication originale, à savoir que les molécules possédant une valeur de CLScore supérieure ou égale à 3.3 sont considérées réalistes.

### **Contrainte de réalisme basée sur un filtrage par liste noire**

Le deuxième type de contrainte que nous étudions est un filtrage binaire selon une liste noire de caractéristiques. Une molécule est considérée réaliste seulement si elle ne contient aucune caractéristique appartenant à une liste prédéfinie. Nous utilisons pour cela la bibliothèque `rd_filters` [WALTERS 2018], selon le paramétrage proposé par les auteurs de l'article GuacaMol [BROWN et al. 2019] (nous nommons pour cette raison cette contrainte « contrainte GuacaMol »). `rd_filters` implémente un ensemble de règles permettant d'identifier des caractéristiques connues comme indésirables pour la chimie pharmaceutique. Cela inclut par exemple le filtrage de fonctions chimiques connues pour leur toxicité sur des organismes biologiques. Notons que ces règles sont spécifiques à la chimie pharmaceutique et pourraient ne pas être pertinentes pour d'autres domaines de la chimie.

### **Contrainte de réalisme basée sur un filtrage par liste blanche**

Finalement, la dernière contrainte que nous étudions est basée sur une liste blanche de caractéristiques. Une molécule est considérée réaliste seulement si toutes ses caractéristiques appartiennent à une liste prédéfinie. Nous utilisons pour cela la bibliothèque `silly_walks`<sup>2</sup> [WALTERS 2020]. Le programme proposé par cette bibliothèque consiste à extraire les caractéristiques ECFP4 de la molécule et à calculer la proportion de ces caractéristiques qui n'apparaissent jamais dans une base de données moléculaires de référence. Pour rappel, les ECFP4 correspondent à des sous-graphes moléculaires de rayon 2 intégrant également les liaisons formées par les atomes aux extrémités du sous-graphe (voir la section 1.1.3 de ce mémoire). Nous utilisons ce programme pour définir une contrainte que nous noterons `sillywalks` dans la suite de ce mémoire. Selon cette contrainte, une molécule est considérée réaliste seulement si l'ensemble de ses caractéristiques ECFP4 appartient à la base de données de référence. Il s'agit de la contrainte la plus stricte, puisqu'elle n'admet aucune caractéristique qui n'est pas déjà connue. Ces caractéristiques étant locales,

---

2. Le nom est inspiré d'un célèbre sketch des Monty Python « *The Ministry of Silly Walks* » figurant des démarches absurdes, comme les molécules que l'on souhaite filtrer.

cela laisse toutefois de très nombreuses possibilités de combinaisons.

Dans la version originale du programme, le jeu de données de référence est un ensemble de 1495 médicaments issus de ChEMBL [GAULTON et al. 2017]. Cela correspond à un total de 9148 caractéristiques ECFP4 différentes. Dans nos travaux, nous choisissons plutôt de considérer l'ensemble des 1 817 795 molécules de ChEMBL afin de définir un espace de recherche plus large et moins spécifique aux médicaments. Rappelons en effet que ChEMBL ne contient pas exclusivement des médicaments mais plus généralement des molécules ayant des activités biologiques. Cela correspond à un total nettement plus élevé de 556 187 caractéristiques ECFP4 différentes. Le programme `silly_walks` calcule la proportion de caractéristiques qui ne font pas partie de la liste blanche. Nous définissons notre contrainte de la manière la plus stricte possible, c'est-à-dire qu'elle pénalise la présence de toute caractéristique n'appartenant pas à la liste.

### 2.4.3 Effet des contraintes pour l'optimisation de la QED

#### Expérimentations

Nous étudions l'effet des contraintes que nous avons définies dans le cadre de l'optimisation des valeurs de QED à l'aide d'EvoMol. La contrainte est intégrée sous la forme d'une fonction binaire à travers le paramètre  $f_{\text{cont}}$  (voir la section 2.2.4 de ce chapitre). Les autres paramètres sont identiques à l'expérience consistant à optimiser les valeurs de QED selon une population de taille 1000 effectuée en section 2.3.1. Nous menons une expérience pour chaque contrainte (valeur de SAScore, valeur de CLScore, contrainte GuacaMol et contrainte `sillywalks`), ainsi qu'une expérience de référence en l'absence de contrainte. Toutes les expériences sont effectuées 25 fois de façon indépendante. Nous choisissons d'effectuer 25 exécutions et non 10 comme nous l'avons fait pour les expériences précédentes car l'analyse du réalisme est en partie qualitative et bénéficie de la présence d'un plus grand nombre d'exemples.

#### Résultats

Les résultats sont présentés en Figures 2.12 à 2.16, qui représentent la meilleure solution obtenue à l'issue des 25 exécutions pour chaque expérience. Si la meilleure évaluation est partagée par plusieurs solutions de la population alors la molécule représentée est sélectionnée aléatoirement parmi ces solutions.

**Absence de contraintes** En l'absence de contrainte (Figure 2.12), on obtient comme on l'attendait des solutions très peu réalistes. Ces solutions sont composées d'un fort taux d'hétéroatomes. On observe des solutions possédant des cycles très contraints ou des cycles imbriqués très peu réalistes (par exemples les solutions D1, E1 ou E4). On observe également des groupes composés exclusivement d'hétéroatomes (par exemple les solutions E2, D3 ou A4). Le SAScore ainsi que le CLScore identifient correctement que ces solutions sont très difficilement synthétisables (le SAScore est quasi-systématiquement supérieur à 5.00 et le CLScore est quasi-systématiquement inférieur à 2.00). En revanche, ces solutions répondent parfaitement au problème d'optimisation, puisque 24 des 25 solutions possèdent une valeur de QED très élevée de 0.948.

**Valeur de SAScore** L'application de la contrainte basée sur les valeurs de SAScore semble améliorer très légèrement le réalisme des solutions (Figure 2.13). On retrouve notamment moins de structures très contraintes. Cette légère amélioration est capturée par le CLScore, dont les valeurs sont légèrement supérieures (le CLScore est souvent supérieur à 2.00). Cependant, on retrouve encore dans ces solutions une forte proportion d'hétéroatomes, et elles demeurent dans l'ensemble très peu réalistes. L'obtention de solutions avec un score élevé n'est pas entravée puisque la totalité des 25 solutions possèdent une valeur de QED de 0.948.

**Valeur de CLScore** Contrairement au SAScore, l'application de la contrainte basée sur la valeur de CLScore possède un impact important sur les résultats. On observe en Figure 2.14 que les solutions semblent dans l'ensemble nettement plus réalistes. Elles contiennent moins d'hétéroatomes, et ces derniers sont principalement des atomes d'azote et d'oxygène. On observe des caractéristiques intéressantes et réalistes comme des conjuguaisons de cycles aromatiques pour les solutions D1, B3 ou B5. Cependant, on observe également des caractéristiques peu réalistes à travers l'ensemble des solutions et notamment un certain nombre de grands cycles composés exclusivement ou quasiment de liaisons simples. L'application de la contrainte entraîne également un effet sur les performances d'optimisation, puisqu'une solution possédant une valeur de QED de 0.948 n'est obtenue qu'au cours de 14 exécutions.

**Contrainte GuacaMol** Les résultats obtenus lors de l'application de la contrainte basée sur une liste noire de caractéristiques sont présentés en Figure 2.15. Contrairement à

l'expérience en l'absence de contrainte, on note une réduction des groupes exotiques composés exclusivement d'hétéroatomes. Cependant, les solutions demeurent dans l'ensemble très peu réalistes. On observe encore une forte proportion d'hétéroatomes. On observe également des cycles très contraints et des imbrications de cycles très peu réalistes, comme les solutions B1, D1, A2 ou encore E4. Concernant les performances d'optimisation, la contrainte possède une influence modérée puisque des solutions possédant une valeur cible de 0.948 sont obtenues lors de 20 exécutions.

**Contrainte sillywalks** Les résultats les plus convaincants sont obtenus par l'application de la contrainte sillywalks (Figure 2.16). Cette contrainte permet d'obtenir des solutions qui semblent vraiment réalistes. Il s'agit notamment des solutions A2, A3 et E5, ainsi que des solutions E3 et C4 qui sont identiques. Les résultats sont toutefois assez comparables aux résultats obtenus avec la contrainte basée sur CLScore, notamment car certaines solutions possèdent également de grands cycles composés exclusivement ou quasi-exclusivement de liaisons simples. De plus, comme pour la contrainte basée sur le CLScore, des solutions possédant une valeur cible de 0.948 sont obtenues lors de 14 des 25 exécutions.

## Discussions

Les contraintes basées sur un seuil appliqué à un score peuvent être intéressantes pour contraindre le réalisme des solutions. Il s'agit en particulier du cas du seuil appliqué au CLScore, qui permet de filtrer une grande quantité de caractéristiques indésirables. En revanche, le seuil appliqué au SAScore mène à l'obtention de résultats décevants. Nous pouvons cependant supposer qu'un seuil inférieur pourrait améliorer les résultats.

Nous avons observé que l'utilisation de la contrainte basée sur une liste noire de caractéristiques (contrainte GuacaMol) mène également à des résultats décevants. Malgré le filtrage, les solutions possèdent de nombreuses caractéristiques indésirables. Nous pouvons raisonnablement supposer qu'il s'agit d'un problème générique au filtrage par liste noire. En effet, l'espace de recherche est de taille massive, et seule une petite partie est connue, stable et synthétisable. Il est probable qu'il ne soit pas possible de lister exhaustivement l'ensemble des caractéristiques non désirées.

En revanche, nous pensons que le filtre basé sur une liste blanche de caractéristiques (contrainte sillywalks) est très intéressant. Ses performances sont assez comparables à celles de la contrainte basée sur le CLScore. Cela est assez facilement compréhensible,

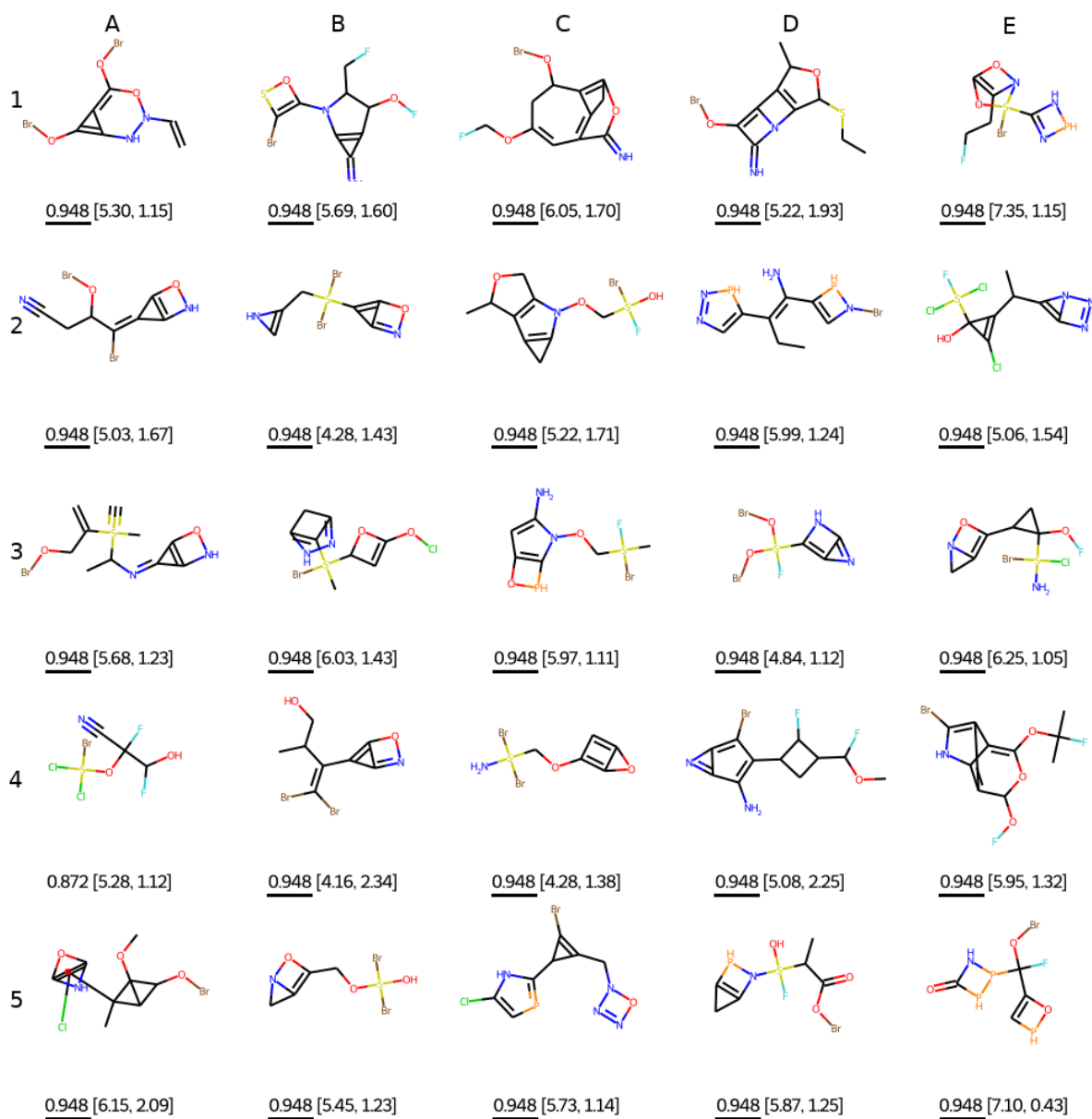


FIGURE 2.12 – Meilleures solutions pour l'optimisation de la QED **sans contrainte** sur l'espace de recherche. Légende : QED [SAScore, CLScore]. Les valeurs de QED égales à 0.948 sont soulignées (**24**).

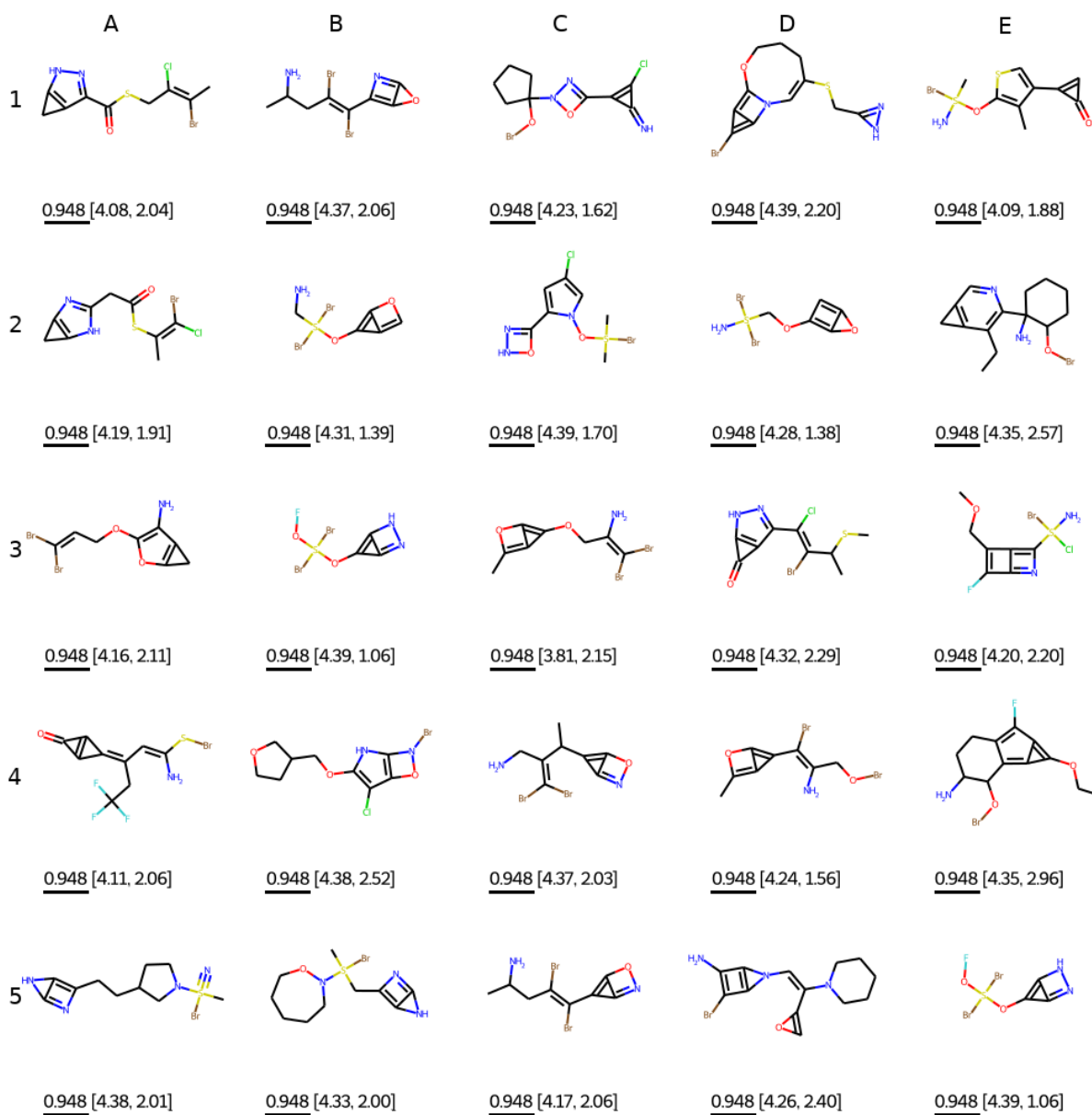


FIGURE 2.13 – Meilleures solutions pour l'optimisation de la QED avec la contrainte correspondant à un **SAScore inférieur ou égal à 4.4**. Légende : QED [SAScore, CLScore]. Les valeurs de QED égales à 0.948 sont soulignées (**25**).

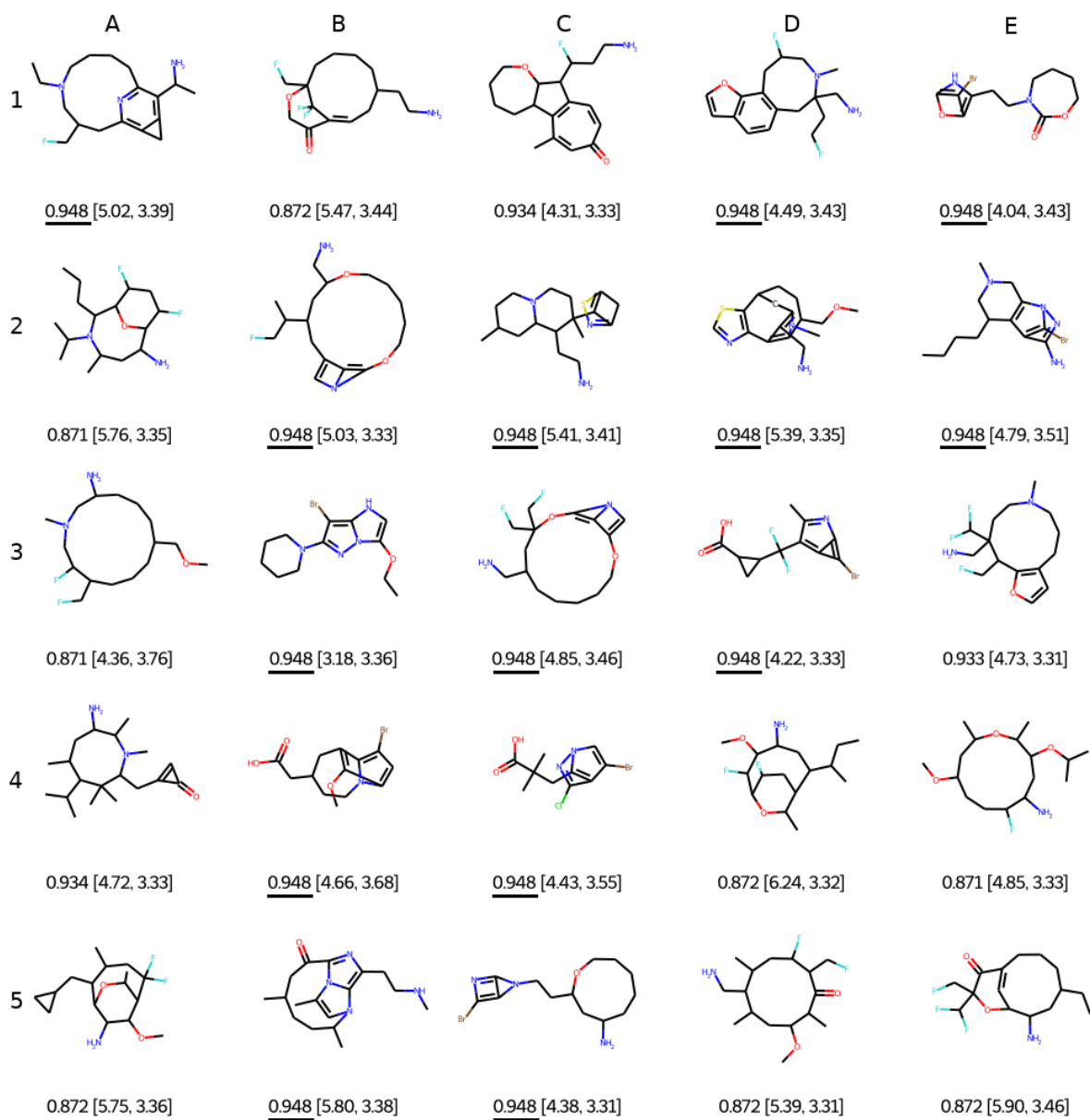


FIGURE 2.14 – Meilleures solutions pour l'optimisation de la QED avec la contrainte correspondant à un **CLScore supérieur ou égal à 3.3**. Légende : QED [SAScore, CLScore]. Les valeurs de QED égales à 0.948 sont soulignées (14).



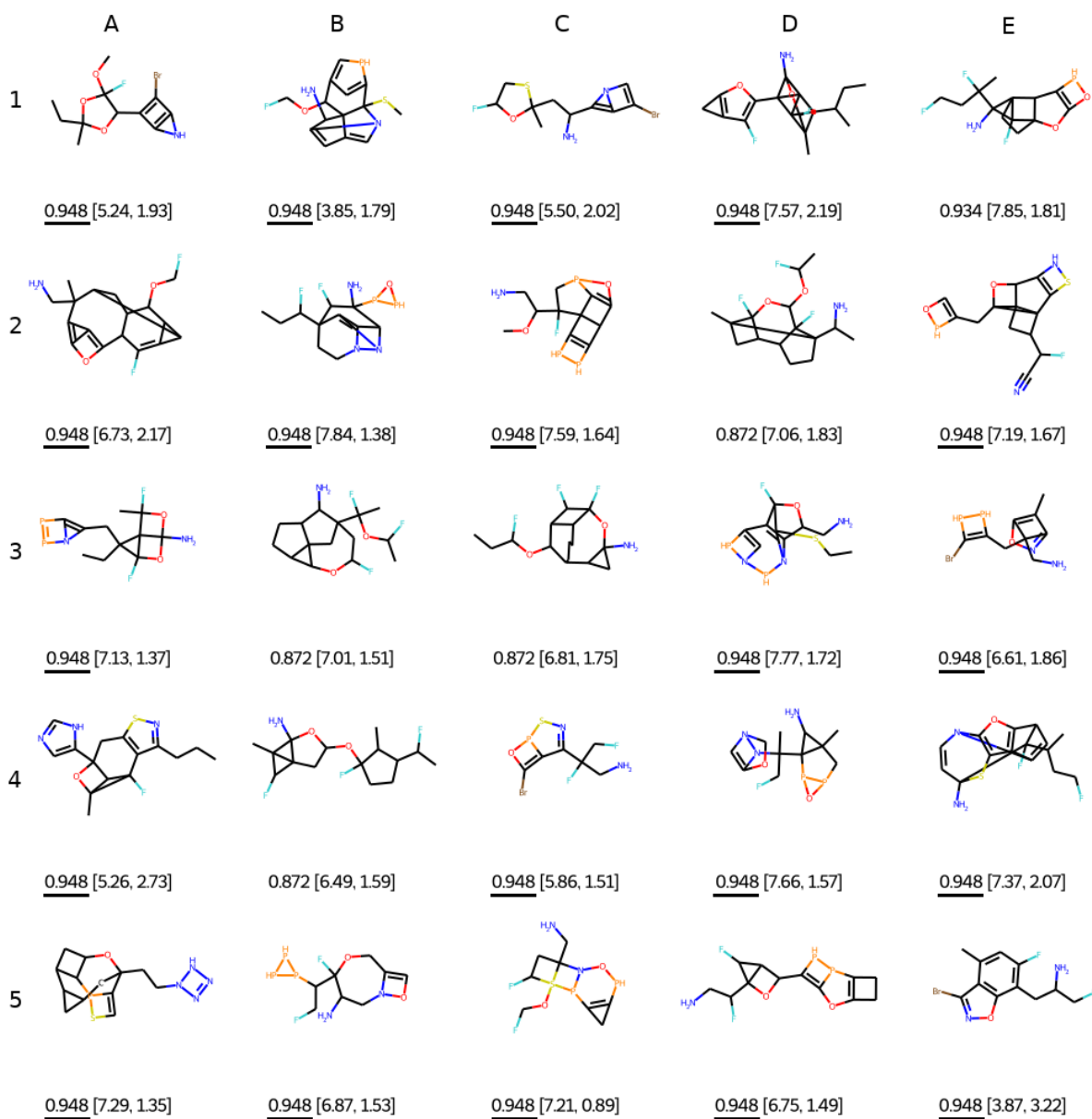


FIGURE 2.15 – Meilleures solutions pour l’optimisation de la QED avec la **contrainte GuacaMol**. Légende : QED [SAScore, CLScore]. Les valeurs de QED égales à 0.948 sont soulignées (20).

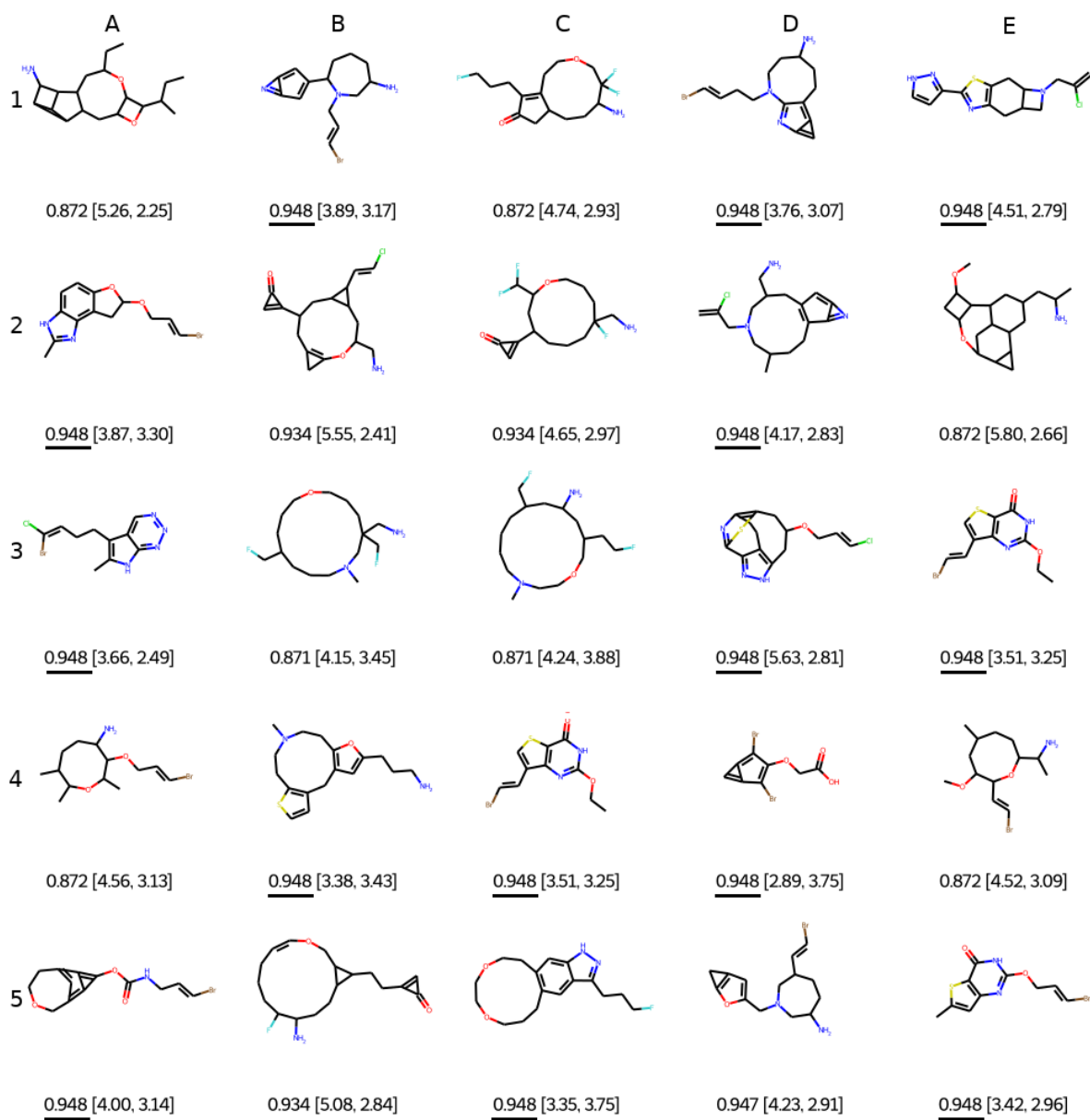


FIGURE 2.16 – Meilleures solutions pour l'optimisation de la QED avec la **contrainte sillywalks**. Légende : QED [SAScore, CLScore]. Les valeurs de QED égales à 0.948 sont soulignées (**14**).

puisque les deux approches sont basées sur des caractéristiques correspondant à des sous-graphes moléculaires extraits de la même base de données moléculaire. Dans la suite de ce mémoire, nous choisissons cependant de nous restreindre à la contrainte sillywalks lorsque nous chercherons à filtrer l'espace de recherche. Ce choix est motivé par le fait que nous nous attendons à ce que cette contrainte permette une exploration moins biaisée de l'espace de recherche. En effet, une valeur élevée de CLScore implique que la distribution des caractéristiques moléculaires se rapporte à celle de ChEMBL. Au contraire, la contrainte sillywalks prend uniquement en compte l'existence des caractéristiques au sein de ChEMBL. Ainsi, nous pouvons nous attendre à obtenir des caractéristiques rares mais que l'on sait existantes ; caractéristiques dont la présence serait pénalisée par le CLScore.

Nous identifions toutefois une limite partagée par la contrainte basée sur le CLScore et la contrainte sillywalks. Ces deux contraintes étant basées exclusivement sur des environnements chimiques locaux, elles ne permettent pas de filtrer des caractéristiques indésirables de grandes tailles, telles que la présence de certains grands cycles contenant des hétéroatomes et dépourvus de liaisons doubles. Finalement, nous observons que l'utilisation de ces contraintes peut avoir une influence non négligeable sur les performances d'optimisation. En effet, nous observons une corrélation négative entre le réalisme des solutions permises par les contraintes et le nombre de solutions possédant des valeurs de fonction objectif élevées. Cela est probablement dû à la présence d'optimums locaux de la fonction objectif dont il est difficile d'échapper dans cet espace de recherche restreint.

#### 2.4.4 Effet des contraintes sur le *benchmark* GuacaMol

Nous souhaitons maintenant étudier l'impact des contraintes sur les performances d'optimisation dans le cadre de tâches d'optimisation plus variées. Nous effectuons pour cela un passage du *benchmark* GuacaMol en appliquant les contraintes sur l'espace de recherche. Les paramètres sont les mêmes que lors des expériences effectuées précédemment, avec l'ensemble des opérateurs de mutation et la population initiale basée sur un sous-ensemble de ChEMBL (voir la section 2.3.2 de ce chapitre). Nous étudions l'effet des deux contraintes binaires basées sur une liste noire (contrainte GuacaMol) et sur une liste blanche (contrainte sillywalks).

Tâche	Expérience référence	Contrainte GuacaMol	Contrainte sillywalks
Celecoxib rediscovery	0.978	<b>1.000</b>	<b>1.000</b>
Troglitazone rediscovery	<b>1.000</b>	0.872	<b>1.000</b>
Thiotixene rediscovery	0.876	0.856	<b>0.958</b>
Aripiprazole similarity	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
Albuterol similarity	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
Mestranol similarity	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
C <sub>11</sub> H <sub>24</sub>	<b>1.000</b>	<b>1.000</b>	0.977
C <sub>9</sub> H <sub>10</sub> N <sub>2</sub> O <sub>2</sub> PF <sub>2</sub> Cl	<b>0.998</b>	0.992	0.925
Median molecules 1	<b>0.455</b>	0.454	0.452
Median molecules 2	<b>0.417</b>	<b>0.417</b>	0.411
Osimertinib MPO	0.955	0.956	<b>0.960</b>
Fexonadine MPO	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
Ranolazine MPO	0.966	<b>0.973</b>	0.961
Perindopril MPO	0.845	<b>0.847</b>	0.806
Amlodipine MPO	0.867	<b>0.904</b>	0.766
Sitagliptin MPO	<b>0.915</b>	0.910	0.766
Zaleplon MPO	<b>0.791</b>	0.781	0.757
Valsartan SMARTS	<b>0.998</b>	0.996	0.997
deco hop	<b>1.000</b>	<b>1.000</b>	0.999
scaffold hop	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
total	<b>18.060</b>	17.959	17.735
total (MPO)	6.339	<b>6.372</b>	6.015
total (rediscovery)	2.853	2.728	<b>2.958</b>

TABLE 2.4 – Résultats sur le *benchmark* GuacaMol avec et sans contrainte favorisant le réalisme des solutions. La première colonne reporte les résultats obtenus dans la section 2.3.2, sans contrainte de réalisme. Les colonnes suivantes correspondent aux résultats obtenus en appliquant respectivement la contrainte GuacaMol basée sur une liste noire, et la contrainte sillywalks basée sur une liste blanche. La coloration permet de mettre en évidence les groupes de tâches.

## Résultats

La Table 2.4 présente les résultats. La première colonne correspond à l'expérience de référence, qui est reportée de la Table 2.2. La première observation que l'on peut effectuer est que l'application des contraintes n'a qu'une influence modérée sur le score total. Celui-ci est en effet de 18,060 en l'absence de contraintes et au minimum de 17,735 lors de l'application de la contrainte sillywalks. Pour comparaison, les méthodes de l'état de l'art obtiennent un score compris entre 17,340 et 18,086 (voir la Table 2.2 en section 2.3.2).

On observe que les contraintes semblent avoir des effets spécifiques en fonction des types de tâches. Le meilleur score obtenu pour les tâches MPO ainsi que le moins bon score obtenu pour les tâches de redécouverte correspond à l'expérience appliquant la contrainte GuacaMol; le moins bon score obtenu pour les tâches MPO ainsi que le meilleur score obtenu pour les tâches de redécouverte correspond à l'expérience utilisant la contrainte sillywalks. Ces différences étant faibles pour certaines, il est possible qu'elles résultent de la variance des résultats. Certains de ces résultats semblent toutefois attendus.

En particulier, il est compréhensible que la contrainte sillywalks pénalise l'obtention de bonnes solutions pour les tâches MPO. Ces dernières consistent en effet à rechercher des solutions maximisant des objectifs contradictoires, qui sont donc plus susceptibles de correspondre à des structures moléculaires peu communes. Réciproquement, on peut facilement comprendre que cette contrainte favorise la redécouverte de molécules connues, d'autant plus que ces dernières font partie du jeu de données ChEMBL qui sert de référence à la contrainte sillywalks. L'utilisation de cette contrainte permet certainement d'éviter un ensemble d'optimums locaux de la fonction objectif qui correspondent à des structures moléculaires exotiques qui sont superflues pour la résolution de ce problème.

## 2.5 Conclusion et perspectives

Dans ce chapitre, nous avons proposé un algorithme évolutionnaire pour l'optimisation de propriétés moléculaires. Notre approche est conçue pour être générique, pour permettre au maximum un contrôle des biais intégrés lors de la recherche et pour offrir une certaine interprétabilité aux utilisateurs du domaine d'application. La généralité et le contrôle des biais sont en fait des caractéristiques très liées. Dans la version de base de notre approche, nous évitons toute hypothèse liée à un domaine spécifique de la chimie. Nous intégrons toutefois un ensemble de paramètres et mécanismes permettant aux utilisateurs du domaine d'application d'intégrer volontairement des biais permettant une résolution plus efficace

d'un problème en particulier, selon leur expertise. L'interprétabilité de notre approche est offerte par la possibilité de générer a posteriori un arbre d'exploration permettant de retracer les chemins parcourus au sein de l'espace de recherche.

Nous avons observé que notre approche possède de très bonnes performances d'optimisation pour diverses propriétés moléculaires. L'étude des expériences que nous avons effectuées montre que notre algorithme est très efficace pour l'intensification, préférentiellement à l'exploration de l'espace de recherche. Cela est favorisé par le fait que notre procédure d'optimisation consiste à remplacer les moins bons individus de la population par la mutation des meilleurs, et que les opérateurs de mutation forment un voisinage très local constitué principalement de perturbations définies au niveau atomique. Bien que nous n'ayons pas observé de contre-performances dues à un manque d'exploration, il est raisonnable d'anticiper que cette prédisposition pour l'intensification peut être préjudiciable pour certains problèmes d'optimisation. Dans le chapitre suivant, nous proposons une approche permettant de favoriser l'exploration de l'espace de recherche par notre algorithme évolutionnaire. Cette approche est basée sur la définition d'un objectif favorisant la diversité moléculaire au sein de la population.

Lors de l'étude des performances de notre approche sur le *benchmark* GuacaMol, nous avons observé qu'elle est susceptible de nécessiter un très grand nombre d'appels à la fonction objectif. Il s'agit d'un résultat prévisible pour un algorithme évolutionnaire, car ces derniers utilisent la fonction objectif de manière « aveugle », et n'en tirent aucune connaissance permettant d'anticiper ses valeurs pour de nouveaux points de l'espace de recherche. Cela peut être préjudiciable dans certains cas et notamment pour l'optimisation de propriétés électroniques, qui dépendent de calculs coûteux. Nous proposerons au Chapitre 4 une méthode d'optimisation basée sur notre algorithme évolutionnaire et sur un modèle de substitution de la fonction objectif. Cette méthode est définie pour permettre une optimisation plus efficace, et notamment en nombre d'appels à la fonction objectif.

Finalement, nous avons proposé une approche pour favoriser le réalisme des molécules générées, basée sur une contrainte restreignant l'espace de recherche accessible. La contrainte dite *sillywalks* basée sur une liste blanche de caractéristiques moléculaires permet en particulier une amélioration nette du réalisme des solutions. Nous observons toutefois que certaines caractéristiques indésirables qui s'expriment à l'échelle des cycles demeurent valides selon cette contrainte. Dans des travaux très récents que nous évoquons dans la conclusion de ce mémoire, notre équipe de recherche propose une nouvelle contrainte qui prend en compte le réalisme des cycles.



# OPTIMISATION DE LA DIVERSITÉ MOLÉCULAIRE

---

## Sommaire

---

<b>3.1</b>	<b>Introduction . . . . .</b>	<b>133</b>
<b>3.2</b>	<b>Quantification de la diversité moléculaire . . . . .</b>	<b>135</b>
3.2.1	Mesures de diversité . . . . .	136
3.2.2	Descripteurs moléculaires . . . . .	138
<b>3.3</b>	<b>Méthode . . . . .</b>	<b>140</b>
<b>3.4</b>	<b>Génération d'un jeu de données avec une forte diversité . . . . .</b>	<b>146</b>
<b>3.5</b>	<b>Optimisation conjointe de la diversité et d'une propriété moléculaire</b>	<b>150</b>
<b>3.6</b>	<b>Conclusion et perspectives . . . . .</b>	<b>156</b>

---

Ce chapitre fait l'objet de la publication suivante.

[LEGUY et al. 2021c]





Dans ce chapitre, nous proposons une méthode pour l’optimisation de la diversité moléculaire au sein d’un jeu de données de molécules. Notre approche est basée sur l’algorithme évolutionnaire que nous avons proposé au Chapitre 2, ainsi que sur une mesure efficace de la contribution à la diversité moléculaire du jeu de données. Nous montrons que cette approche permet effectivement la génération d’un jeu de données moléculaires divers de très grande taille. Nous montrons également qu’il est possible d’optimiser conjointement l’objectif de contribution à la diversité moléculaire et une propriété moléculaire cible, et que cela peut permettre d’optimiser cette propriété cible de manière plus efficace.

## 3.1 Introduction

Des travaux effectués au sein de notre équipe antérieurement à cette thèse ont montré qu’un manque de diversité moléculaire au sein d’un jeu de données de molécules peut affecter négativement les performances de modèles d’apprentissage supervisé de propriétés moléculaires qui l’utilisent en tant que jeu de données d’entraînement [GLAVATSKIKH et al. 2019]. Ces travaux ont été menés sur le jeu de données QM9 [RAMAKRISHNAN et al. 2014], mais il s’agit d’un effet que l’on peut s’attendre à observer pour d’autres jeux de données moléculaires de référence. Ces derniers correspondent en effet à la sélection d’un sous-ensemble spécifique de l’espace moléculaire, qui peut être plus ou moins bien identifié en fonction de la façon dont le jeu de données est construit mais qui ne peut a priori pas être considéré comme représentatif de l’ensemble de l’espace moléculaire.

Dans un contexte d’optimisation de propriétés moléculaires coûteuses, des modèles d’apprentissage prédisant avec une précision suffisante les valeurs de propriété cible pourraient permettre de limiter le coût de l’optimisation. Or, les résultats de nos travaux antérieurs laissent entendre que la qualité de ces modèles d’apprentissage serait conditionnée entre autres par la qualité du jeu de données d’entraînement. Ces résultats motivent les travaux que nous menons dans ce chapitre, qui consistent à proposer une approche pour la génération d’un jeu de données avec une forte diversité moléculaire. Ce jeu de données pourrait être utilisé en tant que jeu de données d’entraînement d’un modèle d’apprentissage de propriétés moléculaires, dont on pourrait espérer qu’il possède une meilleure capacité de généralisation.

Nous proposons de considérer le problème de génération d’un jeu de données moléculaires divers comme un problème d’optimisation moléculaire. Le concept de diversité moléculaire peut être saisi intuitivement, mais il existe différentes façons de le définir.

Indépendamment de sa définition, nous cherchons à maximiser la diversité moléculaire au sein d'un ensemble de solutions à l'aide d'une méthode d'optimisation. Les problèmes d'optimisation sont typiquement définis à l'aide d'une fonction objectif qui évalue la qualité d'une solution de façon indépendante des autres solutions de l'espace de recherche. Or dans le cas présent, la qualité des molécules ne peut pas être évaluée sans considérer l'ensemble du jeu de données. La mesure de diversité est en fait une évaluation du jeu de données, et non des solutions qu'il contient. Afin de traiter ce problème comme un problème d'optimisation moléculaire, nous proposons de considérer une mesure de *contribution* à la diversité moléculaire de l'ensemble du jeu de données. Cette mesure peut être utilisée en tant que fonction objectif d'une méthode d'optimisation, mais nécessite toutefois une procédure d'optimisation adaptée car son calcul est dépendant de l'ensemble des molécules contenues dans le jeu de données.

Pour résoudre le problème d'optimisation de la contribution à la diversité moléculaire, nous proposons d'utiliser l'algorithme évolutionnaire que nous avons conçu au chapitre précédent. Nous choisissons de définir la diversité moléculaire selon la mesure d'entropie de Shannon, appliquée à un descripteur moléculaire [SHANNON 1948]. Il s'agit d'une métrique relativement coûteuse. Nous allons cependant montrer que nous pouvons en proposer une approximation efficace dans le cadre de notre algorithme évolutionnaire. Cette approximation est rendue possible par la considération des descripteurs à l'échelle de la population et par une mise en cache de ces valeurs et un traitement par lot lors de la génération des solutions à chaque étape d'optimisation.

Dans la littérature, il existe des travaux associant un algorithme évolutionnaire avec une procédure permettant de garantir un certain niveau de diversité au sein de la population de solutions. À titre d'exemple, [TSUJIMURA et GEN 1998] proposent d'appliquer des perturbations aléatoires à des individus de la population lorsque la valeur d'une mesure de la diversité au sein de la population est inférieure à un seuil. Pour l'optimisation moléculaire, [NIGAM et al. 2020] proposent un algorithme évolutionnaire pour l'optimisation de propriétés moléculaires, dans lequel un réseau de neurones est utilisé pour assigner un score favorisant la génération de molécules n'appartenant pas à des zones précédemment explorées de l'espace moléculaire. MolFinder est un algorithme de type CSA, dont le fonctionnement est décrit en section 1.3.1 de ce mémoire, et qui intègre donc des mécanismes permettant de garantir un certain niveau de diversité au sein de la population [KWON et LEE 2021]. L'intérêt de ces mécanismes est de favoriser l'exploration de l'espace de recherche lors de la procédure d'optimisation. Or, nous avons observé dans le chapitre

précédent que notre algorithme évolutionnaire possède une tendance forte à l'intensification. Ces travaux suggèrent donc qu'indépendamment de notre objectif de générer un jeu de données possédant une grande diversité, notre approche pourrait permettre d'améliorer les performances d'optimisation de notre algorithme évolutionnaire en offrant un meilleur compromis entre intensification et exploration.

Les travaux que nous présentons au sein de ce chapitre ont été effectués en collaboration avec Marta Glavatkikh, qui est une chimiste qui au moment de la réalisation de ces travaux était post-doctorante au sein de notre équipe de recherche. Nous évoquons dans ce chapitre l'ensemble des travaux que nous avons effectués en collaboration sur ce sujet. Nous présentons un résumé des travaux correspondant à ses contributions. Il s'agit de la génération d'un jeu de données de grande taille possédant une diversité moléculaire importante, et de l'étude de cette diversité. Nous présentons en détail les travaux correspondant à mes contributions personnelles. Il s'agit de la définition d'une méthode efficace pour la maximisation de la diversité moléculaire, et de la démonstration que cette diversité peut permettre l'optimisation plus efficace d'une propriété moléculaire.

Ce chapitre est organisé de la manière suivante. D'abord, nous effectuons un bref état de l'art de mesures de la diversité moléculaire proposées dans la littérature, ainsi que de descripteurs moléculaires qui sont adaptés pour servir de support à ces mesures. Par la suite, nous détaillons notre méthode d'optimisation de la diversité moléculaire ainsi que les approximations qui en permettent un calcul efficace. Nous résumons ensuite les travaux menés par Marta Glavatkikh, qui montrent que notre approche permet effectivement de générer une diversité moléculaire importante. Finalement, nous étudions notre approche dans le cadre de l'optimisation conjointe de l'objectif de diversité et d'une propriété moléculaire.

## 3.2 Quantification de la diversité moléculaire

Nous proposons dans cette section un bref état de l'art dédié à la quantification de la diversité moléculaire au sein de la littérature. Nous nous intéressons à différentes mesures de diversité, puis aux descripteurs moléculaires sur lesquelles elles peuvent s'appuyer. Ces concepts ne sont pas abordés au Chapitre 1 d'état de l'art car ils sont mobilisés exclusivement dans le présent chapitre.

### 3.2.1 Mesures de diversité

**Diversité externe** Certains *benchmarks* proposent d'évaluer la capacité des modèles de génération à reproduire la distribution d'un jeu de données de référence selon une métrique [BROWN et al. 2019; POLYKOVSKIY et al. 2020]. Pour cela, une mesure que l'on peut qualifier de « diversité externe » est définie afin de mesurer la distance entre les données générées et un jeu de référence. Il s'agit d'une métrique qui a du sens particulièrement pour les méthodes de génération moléculaire basées sur un modèle d'apprentissage profond génératif. On cherche alors à montrer que le modèle peut générer des molécules qui suivent la distribution du jeu de données d'entraînement. Dans ce contexte, la diversité externe est une métrique que l'on cherche à minimiser. Parmi ces métriques, nous pouvons mentionner la « *Fréchet ChemNet distance* » (FCD), qui est mesurée dans l'espace latent d'un réseau de neurones entraîné pour la prédiction de propriétés moléculaires [PREUER et al. 2018]. Nous pouvons également mentionner la mesure de distance au voisin le plus proche. Cette dernière est calculée comme la moyenne des distances entre les éléments de l'ensemble évalué et leurs voisins les plus proches dans le jeu de référence. Cela nécessite une mesure de distance entre deux molécules quelconques. La distance de Tanimoto entre les caractéristiques ECFP est communément utilisée pour cela (voir la section 1.1.3 de ce mémoire).

**Diversité interne** Les mesures de diversité externe ne sont pas adaptées aux travaux que nous menons dans ce chapitre, car elles sont exprimées à partir d'un jeu de données de référence externe. Or, nous souhaitons ici maximiser la diversité interne à un unique jeu de données. Dans la littérature, des travaux proposent d'étudier la diversité moléculaire à partir de l'étude des caractéristiques d'un descripteur moléculaire pertinent [LIPKUS et al. 2008]. Nous décrivons dans la suite de ce chapitre plusieurs descripteurs moléculaires qui peuvent être utilisés dans ce contexte. L'analyse statistique de ces descripteurs permet de décrire une forme de diversité moléculaire au sein d'un jeu de données, mais n'en fournit pas une mesure quantitative. Une façon directe d'obtenir une telle mesure est de calculer la distance moyenne entre toutes les paires de molécules, selon une mesure de distance [BENHENDA 2017]. Ce calcul peut être formalisé de la façon suivante, avec  $D$  la mesure de diversité basée sur la distance ainsi définie,  $X$  le jeu de données dont on souhaite évaluer la diversité, et  $d$  une fonction évaluant la distance entre deux points de l'espace de recherche. La notation  $|X|$  représente la cardinalité de l'ensemble  $X$  et donc le nombre de valeurs qu'il contient.

$$D(X) = \frac{1}{|X|^2} \sum_{(x,y) \in X \times X} d(x,y) \quad (3.1)$$

Pour notre application, l'inconvénient majeur de cette approche est qu'il s'agit d'un calcul relativement coûteux. Considérons la contribution à la diversité de l'ajout d'une solution  $x$  dans un jeu de données  $X$ , qui correspond au calcul  $D(X \cup \{x\}) - D(X)$ . Cela implique le calcul des distances entre toutes les paires de solutions de  $X$  et de  $X \cup \{x\}$ . Les résultats intermédiaires du calcul de  $D(X)$  peuvent être réutilisés pour le calcul de  $D(X \cup \{x\})$ , mais il est néanmoins nécessaire de calculer les distances entre  $x$  et tous les éléments de  $X$  à chaque évaluation de la contribution d'une solution  $x$  quelconque à la diversité d'un jeu de données  $X$ .

Dans nos travaux, nous utiliserons une mesure de la diversité moléculaire basée sur la mesure d'entropie de Shannon [SHANNON 1948]. Cette mesure peut être calculée de manière exacte selon l'équation (3.2), pour un jeu de données  $X$  qui correspond ici à une matrice de taille  $M \times N$  représentant pour chaque molécule sa valeur selon un descripteur binaire. Le jeu de données contient  $M$  molécules et le descripteur est de dimension  $N$ . Nous notons  $P_i(X)$  la proportion de molécules de  $X$  pour lesquelles la  $i^{\text{ème}}$  caractéristique du descripteur vaut Vrai.  $\bar{X}$  correspond à la négation de la matrice  $X$ , c'est-à-dire que toutes les valeurs Vrai prennent la valeur Faux et inversement.

$$H_{\text{exact}}(X) = - \sum_{i=1}^N \left( P_i(X) \log P_i(X) + P_i(\bar{X}) \log P_i(\bar{X}) \right) \quad (3.2)$$

Dans l'équation (3.2), l'expression contenue dans la somme vaut 0 si  $P_i(X)$  vaut 0 ou 1, c'est-à-dire si la  $i^{\text{ème}}$  caractéristique du descripteur n'est présente dans aucune solution ou si elle est présente dans toutes les solutions. Il s'agit d'un comportement attendu puisque dans les deux cas il n'existe aucune diversité pour cette caractéristique. Chaque terme de la somme admet sa valeur minimale pour une proportion  $P_i(X)$  égale à 0.5. Cela implique que la valeur maximale de l'entropie du jeu de données est atteinte si toutes les caractéristiques du descripteur sont présentes dans la moitié des solutions. Nous montrerons dans la suite de ce chapitre que la mesure de contribution à la diversité peut être calculée de manière efficace lorsqu'elle est définie à partir de la mesure d'entropie.

### 3.2.2 Descripteurs moléculaires

Les mesures de diversité sont exprimées à partir d'un descripteur moléculaire. En fonction des mesures, des descripteurs très différents peuvent être utilisés. Par exemple, pour la diversité externe, le descripteur moléculaire nécessaire au calcul de la mesure de FCD est l'espace latent d'un modèle d'apprentissage profond. Pour les métriques exprimées à partir d'une mesure de distance moléculaire, les caractéristiques ECFP sont typiquement utilisées en association avec la distance de Tanimoto.

L'estimation de la diversité selon l'entropie de Shannon contraint très peu le choix du descripteur moléculaire. Tout descripteur moléculaire binaire peut ainsi être utilisé. Pour les expériences que nous mènerons dans la suite de ce chapitre, nous utiliserons le descripteur sous forme de vecteur binaire de *shingles* que nous avons présenté en section 1.1.3 de ce mémoire. Pour rappel, les *shingles* correspondent à des sous-graphes moléculaires centrés autour d'un atome et d'un rayon défini par un paramètre  $r$ . Les caractéristiques définies par les *shingles* sont très génériques. Dans un contexte d'estimation de la diversité, ce descripteur peut toutefois souffrir de défauts potentiels. D'abord, il encode des structures locales, et ne peut donc pas représenter des caractéristiques à l'échelle des cycles ou des groupes de cycles selon la valeur de  $r$ , ni à l'échelle de la molécule. Ensuite, la grande généralité des *shingles* implique une explosion combinatoire rapide qui risque de les rendre impraticables pour des rayons supérieurs à 1 ou 2 atomes. Dans les travaux que nous effectuons au sein de ce chapitre, nous étudions également d'autres descripteurs plus spécifiques, qui permettent de représenter des caractéristiques qui nous semblent pertinentes dans un contexte d'estimation de la diversité.

***Scaffolds* génériques et *scaffolds*** Les *scaffolds* génériques sont des descripteurs dont l'objectif est de représenter de manière générique l'ensemble de la structure moléculaire. Ils correspondent en fait au graphe non valué des atomes et des liaisons, c'est-à-dire que ni les types d'atomes ni les types de liaisons ne sont représentés. Ce descripteur pourrait ainsi être encodé sous la forme d'un graphe classique avec un ensemble de sommets et d'arêtes non valués. Par mesure d'homogénéité avec les autres descripteurs basés sur le graphe, nous le représentons sous la forme d'un graphe moléculaire dans lequel tous les atomes sont remplacés par des atomes de carbone, et toutes les liaisons sont remplacées par des liaisons simples. Il s'agit de la façon dont ce descripteur est implémenté au sein de la bibliothèque RDKit [LANDRUM 2010]. À titre d'exemple, la Figure 3.1 présente le *scaffold* générique de la molécule de celecoxib. Il existe une variation de ce descripteur nommée

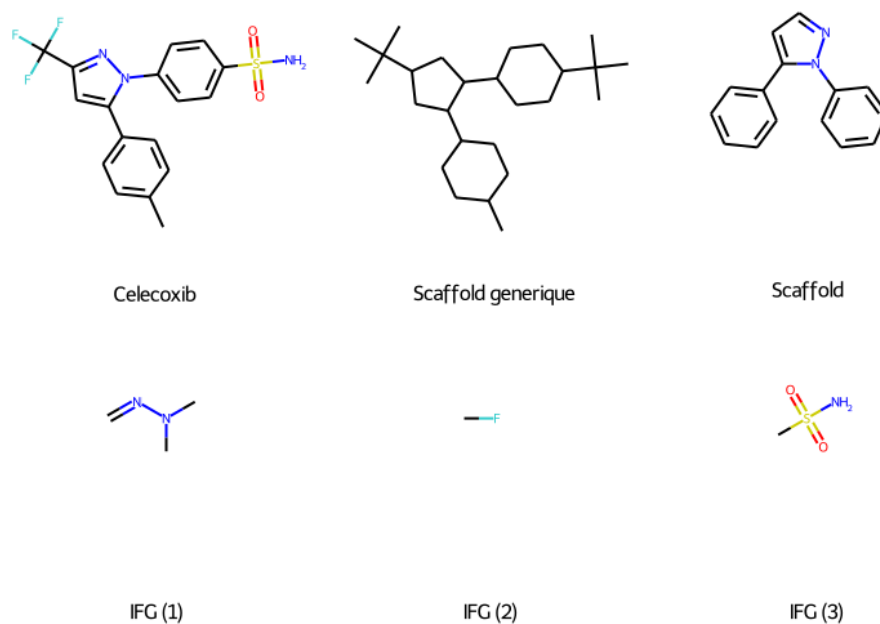


FIGURE 3.1 – Dessin moléculaire de la molécule de celecoxib et de descripteurs moléculaires qui en sont tirés. *Scaffold*, *scaffold* générique et l'ensemble des trois groupes IFG.

*scaffold*, qui comme son nom l'indique est moins générique dans le sens où elle considère le type des atomes et des liaisons. Toutefois ce descripteur ne représente pas l'ensemble de la molécule, sans quoi il serait équivalent au graphe moléculaire, mais plus spécifiquement la partie « centrale » de la molécule qui contient l'ensemble des cycles. Précisément, les *scaffolds* sont définis comme le plus petit sous-graphe moléculaire contenant l'ensemble des cycles de la molécule. Cela revient à représenter le *cœur* de la molécule, sur lequel peuvent être branchés des groupes d'atomes. À titre d'exemple, nous présentons dans la Figure 3.1 le *scaffold* de la molécule de celecoxib. Nous pouvons observer qu'il contient les trois cycles de la molécule ainsi que les liaisons qui les unissent, mais pas les groupes d'atomes qui y sont branchés.

**IFG** Les *scaffolds* et les *scaffolds* génériques sont définis pour représenter une structure globale de la molécule. D'autres descripteurs comme les *shingles* sont au contraire définis pour encoder des structures locales. Nous nous intéressons désormais aux descripteurs nommés IFG, ce qui correspond à l'acronyme de l'algorithme qui les génère et qui est nommé « *identify functional groups* » [ERTL 2017]. Les IFG correspondent également à des sous-graphes moléculaires, mais contrairement aux *shingles* sont de taille variable et



sont définis pour encoder un sous-ensemble des environnements locaux jugé pertinent par des chimistes. Les IFG correspondent en réalité à une façon d’extraire dynamiquement les « groupes fonctionnels », c’est-à-dire des groupes d’atomes dont il est attendu qu’ils aient des propriétés spécifiques lorsqu’ils sont présents dans une molécule. Les IFG sont construits en marquant d’abord tous les hétéroatomes de la molécule, ainsi que certains atomes de carbone selon des règles liées à la chimie et aux groupes fonctionnels que nous ne détaillons pas ici. Tous les sous-graphes connectés composés exclusivement d’atomes marqués sont extraits et forment les IFG de la molécule. En Figure 3.1, nous représentons le dessin moléculaire des trois caractéristiques IFG extraites de la molécule de celecoxib.

**Checkmol** Pour finir, nous nous intéressons à un descripteur obtenu par un programme nommé CheckMol [HAIDER 2010]. Par souci de simplicité, nous nommons également ce descripteur checkmol. Ce descripteur encode également des groupes fonctionnels, mais à partir cette fois d’une liste prédéfinie de groupes admissibles qui sont détectés dans la molécule. Il s’agit donc d’un descripteur plus spécifique que les IFG, puisque les groupes fonctionnels ne sont pas identifiés dynamiquement. La liste des groupes détectés est de plus assez restreinte, puisqu’elle en contient 204. Il s’agit de groupes fonctionnels communément reconnus dans la nomenclature des chimistes. Contrairement aux descripteurs présentés précédemment qui sont basés sur des caractéristiques correspondant à des graphes moléculaires, les caractéristiques de checkmol sont des chaînes de caractères décrivant les groupes détectés au sein des molécules. À titre d’exemple, la molécule de celecoxib présentée en Figure 3.1 possède quatre groupes fonctionnels selon CheckMol. Il s’agit des groupes « dérivé d’halogène » car la molécule contient des atomes de fluor, « sulfonamide » car elle contient le groupe formé autour de l’atome de soufre, « composé aromatique » car elle contient au moins un cycle aromatique et finalement « composé hétérocyclique » car l’un des cycles contient des hétéroatomes<sup>1</sup>.

### 3.3 Méthode

#### Approximation de la mesure d’entropie

Nous proposons d’aborder le problème de génération d’un jeu de données moléculaires divers comme un problème d’optimisation moléculaire. Pour que ce problème puisse être

---

1. Pour rappel, le terme hétéroatome qualifie tout atome lourd qui n’est pas un atome de carbone.

traité par des méthodes d'optimisation moléculaire qui sont basées sur la définition d'une fonction objectif évaluant des molécules, nous allons définir un objectif de contribution à la diversité totale du jeu de données. Baser cette procédure d'optimisation sur la fonction  $H_{\text{exact}}$  définie en équation (3.2) serait très coûteux. Nous allons effectuer plusieurs approximations et optimisations de ce calcul afin d'en limiter le coût. La première approximation consiste à ne considérer que les caractéristiques qui sont présentes dans le descripteur moléculaire, et non les caractéristiques absentes que l'on représente par  $\bar{X}$  dans la définition de  $H_{\text{exact}}(X)$ . Cela permet de simplifier l'expression de la mesure d'entropie, qui peut alors s'exprimer selon les équations (3.3) et (3.4). Cette nouvelle formulation est équivalente à la définition de  $H_{\text{exact}}$  si l'on ignorait le second terme de la somme. Dans cette nouvelle formulation, nous séparons le calcul de l'entropie globale du jeu de données  $X$  en équation (3.3) et le calcul de la contribution de chaque descripteur à l'entropie totale en équation (3.4). Dans cette dernière,  $D_i$  représente la caractéristique dont on cherche à évaluer l'entropie au sein de la population, dont  $i$  est l'indice au sein du descripteur moléculaire binaire.  $C_i(X)$  correspond au nombre d'occurrences de  $D_i$  au sein de  $X$ , c'est-à-dire au nombre de fois que la valeur Vrai est présente en colonne  $i$  de la matrice  $X$ .  $|X|$  correspond à la taille de la population. Selon les notations utilisées préalablement,  $\frac{C_i(X)}{|X|}$  pourrait être remplacé par  $P_i(X)$ . Nous préférons ici utiliser la notation  $\frac{C_i(X)}{|X|}$ , car elle fait apparaître explicitement la matrice  $X$  et sa taille  $|X|$ , qui sont nécessaires pour comprendre les approximations que nous définissons dans la suite de cette section.

$$H(X) = \sum_{i=1}^N H(D_i, X) \quad (3.3)$$

$$H(D_i, X) = -\frac{C_i(X)}{|X|} \log \frac{C_i(X)}{|X|} \quad (3.4)$$

Comme pour l'équation (3.2), l'expression contenue dans la somme c'est-à-dire ici  $H(D_i, X)$  vaut 0 si une caractéristique n'est présente dans aucune molécule ou si elle est présente dans toutes les molécules du jeu de données. Cela correspond aux cas pour lesquels  $P_i(X)$  (ou ici  $\frac{C_i(X)}{|X|}$ ) vaut 0 ou 1. En revanche, le point pour lequel cette contribution est la plus importante est ici  $\frac{C_i(X)}{|X|} = e^{-1}$ , c'est-à-dire environ 0.368. Alors que la contribution d'un descripteur à l'entropie totale est maximale lorsque la proportion d'occurrences de ce descripteur au sein du jeu de données est de 0.5 pour  $H_{\text{exact}}$ , elle est maximale pour une valeur plus faible pour l'approximation de l'entropie que nous définissons ici. Par conséquent, maximiser  $H(X)$  favorisera légèrement l'absence de ca-

ractéristiques du descripteur plutôt que leur présence. En pratique, cela aura un impact très faible car les vecteurs des descripteurs que nous serons amenés à manipuler sont creux, et contiennent donc majoritairement des valeurs Faux. Cela est dû au fait que si l'on considère une contrainte de taille maximale pour les molécules, chaque molécule ne peut contenir qu'une quantité relativement faible de caractéristiques. Cette quantité est nettement inférieure au nombre de caractéristiques possibles.

### **Définition des mesures de contribution à la diversité moléculaire**

Nous cherchons à adapter l'algorithme évolutionnaire que nous avons proposé au Chapitre 2 au problème de la maximisation de la contribution à la diversité moléculaire de la population. Dans notre algorithme évolutionnaire, la population admet une taille maximale, ce qui implique que pour être inséré dans la population un individu doit en remplacer un autre. Pour rappel, nous considérons qu'il peut exister au sein de la population des individus indéfinis qui sont alors remplacés en priorité jusqu'au remplissage de la population par des individus définis. L'impact sur la diversité globale de la population du remplacement d'un individu par un autre dépend de l'individu ajouté mais également de l'individu qui disparaît. D'autre part, rappelons que notre algorithme évolutionnaire nécessite au début de chaque étape d'optimisation un tri des individus selon la fonction objectif pour former la liste notée  $L$  des individus devant être remplacés à l'étape courante. Ce tri est décrit dans la section 2.2.4 de ce mémoire. Cela nous amène à évaluer la différence d'entropie associée à la suppression d'un individu pour la sélection des individus devant être remplacés, et la différence associée à l'ajout d'un individu pour la sélection des mutants insérés au sein de la population.

Ainsi, nous définissons une mesure  $\Delta_s(x, X)$  qui correspond à la différence d'entropie qui résulte de la suppression de la solution  $x$  de la population. La population est représentée par la matrice  $X$ , qui est obtenue par calcul du descripteur pour tous les individus.  $\Delta_s(x, X)$  peut être calculé naïvement selon l'équation (3.5). La liste  $L$  peut être formée selon un tri décroissant des valeurs de  $\Delta_s(x, X)$  des individus de la population. On remplace ainsi les individus dont la suppression fait le moins diminuer la diversité. Lorsque notre algorithme évolutionnaire est utilisé pour maximiser des propriétés moléculaires indépendamment d'un objectif de diversité, les éléments de  $L$  sont sélectionnés selon un tri croissant des valeurs de fonction objectif, afin que les solutions remplacées soient celles qui possèdent les scores les plus faibles. Pour garder un comportement cohérent, nous choisissons ici de trier les valeurs de  $-\Delta_s(x, X)$  selon un tri croissant.

$$\Delta_s(x, X) = H(X \setminus \{x\}) - H(X) \quad (3.5)$$

De manière symétrique, nous pouvons définir une mesure  $\Delta_a(x, X)$  qui correspond à la différence d'entropie qui résulte de l'ajout de la solution  $x$  au sein de la population.

$$\Delta_a(x, X) = H(X \cup \{x\}) - H(X) \quad (3.6)$$

### Approximation des mesures de contribution à la diversité moléculaire

Nous avons formé la mesure  $H(X)$  de la diversité du jeu de données décrit par la matrice  $X$  comme une approximation de la mesure  $H_{\text{exact}}(X)$ . Nous proposons désormais d'effectuer une approximation supplémentaire, qui correspond à une approximation de la contribution à la diversité moléculaire. Cette approximation, que nous notons  $\Delta'$ , est basée sur l'hypothèse que la population est de taille constante  $|X|$ . Elle permet de réutiliser un grand nombre de calculs, puisque le dénominateur des fractions de l'équation (3.4) est alors constant à travers tous les calculs. Cela implique que lorsqu'un individu est supprimé ou inséré, les calculs sont effectués en considérant que la population est de taille inchangée. Cette approximation se prête facilement au fonctionnement de notre algorithme évolutionnaire, dans lequel les individus sont intégrés à la population par remplacement d'un individu qui disparaît. La population est ainsi de taille constante à l'échelle d'une étape d'optimisation. Rappelons toutefois que lorsque la population est initialisée à partir d'un nombre de solutions inférieur au paramètre de taille de la population, nous considérons que la population est composée d'un certain nombre de solutions non définies, qui sont remplacées progressivement jusqu'à remplissage de la population par des solutions définies. Pour les calculs de contribution à la diversité également, nous considérons que la population est de taille  $|X|$  même si une partie des individus n'est pas définie à l'initialisation.

Selon cette hypothèse, nous pouvons définir le calcul  $\delta_s(D_i, x, X)$  de la différence d'entropie pour une caractéristique  $D_i$  lorsqu'une molécule  $x$  est retirée de la population, selon l'équation (3.7). Dans cette équation,  $X \setminus \{x\}$  est une notation représentant la matrice  $X$  privée des valeurs correspondant à la molécule  $x$ .  $\emptyset$  représente la description d'une « molécule vide » qui est ajoutée artificiellement à  $X$  dans l'équation pour souligner le fait que la taille de  $X$  ne change pas. Le descripteur correspondant à  $x$  peut contenir ou ne pas contenir la caractéristique  $D_i$ . Si elle ne la contient pas, la valeur de  $\delta_s$  est nulle.

$$\delta_s(D_i, x, X) = H(D_i, X \setminus \{x\} \cup \{\emptyset\}) - H(D_i, X) \quad (3.7)$$

Ainsi, seules les caractéristiques contenues dans une molécule  $x$  sont impliquées pour calculer la contribution à l'entropie de la population  $X$  lors de la suppression de  $x$ . Cette quantité, que l'on nomme  $\Delta'_s(x, X)$ , peut être calculée de la façon suivante.

$$\Delta'_s(x, X) = \sum_{D_i \in x} \delta_s(D_i, x, X) \quad (3.8)$$

De façon très similaire, nous pouvons définir une estimation de la contribution à la diversité moléculaire lors de l'ajout d'une molécule au sein de la population. En équation (3.9), nous définissons  $\delta_a(D_i, x, X)$  qui permet de calculer la différence d'entropie pour la caractéristique  $D_i$  lorsque  $x$  est ajoutée à la population. Nous faisons apparaître les descripteurs d'une « molécule vide » qui est virtuellement supprimée de la population afin de faire apparaître que la taille de  $X$  est inchangée dans le calcul de  $H$ . Cela permet de définir la quantité  $\Delta'_a(x, X)$  qui permet de calculer la différence d'entropie causée par l'ajout de la molécule  $x$  au sein de la population, en considérant uniquement les caractéristiques présentes dans  $x$ .

$$\delta_a(D_i, x, X) = H(D_i, X \cup \{x\} \setminus \{\emptyset\}) - H(D_i, X) \quad (3.9)$$

$$\Delta'_a(x, X) = \sum_{D_i \in x} \delta_a(D_i, x, X) \quad (3.10)$$

La quantité  $\Delta'_a(x, X)$  peut être utilisée pour évaluer la contribution à l'entropie de l'ajout d'une solution au sein de la population de notre algorithme évolutionnaire. Cependant elle ne prend pas réellement en compte la différence d'entropie qui a lieu lors de l'insertion de la solution au sein de la population, puisque cette solution en remplace nécessairement une autre dont les caractéristiques sont alors retirées de la matrice  $X$ . Ainsi, nous proposons une dernière mesure permettant d'estimer la contribution à l'entropie de la population du remplacement d'une molécule  $x_s$  par une molécule  $x_a$ , selon l'ensemble des approximations que nous avons définies. Cette mesure est notée  $\Delta'_r(x_s, x_a, X)$  et est définie selon l'équation (3.11). La notation  $x_s \setminus x_a$  (resp.  $x_a \setminus x_s$ ) représente l'ensemble des caractéristiques de la molécule  $x_s$  qui ne sont pas présentes dans  $x_a$  (resp. l'ensemble des caractéristiques de  $x_a$  qui ne sont pas présentes dans  $x_s$ ). Cette quantité correspond à la somme de la contribution à l'entropie de la suppression de  $x_s$  et de la contribution à

l'entropie de l'ajout de  $x_a$ . Les caractéristiques qui sont présentes dans les deux solutions sont ignorées lors du calcul puisqu'elles restent inchangées dans la population malgré le remplacement.

$$\Delta'_r(x_s, x_a, X) = \Delta'_s(x_s \setminus x_a, X) + \Delta'_a(x_a \setminus x_s, X) \quad (3.11)$$

### Optimisation évolutionnaire efficace de l'entropie de la population

Nous intégrons les équations que nous avons définies dans cette section au sein de notre algorithme évolutionnaire. Cette intégration est relativement transparente en termes d'implémentation car nos équations sont définies pour l'optimisation de la diversité à l'aide d'un algorithme à base de population. Il existe toutefois deux spécificités de la méthode que nous présentons ici qui doivent être prises en compte pour l'implémentation. La première est que la fonction  $\Delta'_s$  utilisée pour sélectionner les solutions candidates au remplacement est différente de la fonction  $\Delta'_r$  qui est utilisée pour évaluer des solutions pour leur ajout dans la population. Dans un algorithme évolutionnaire classique, la fonction objectif permet de remplir ces deux rôles. La deuxième spécificité est que la fonction  $\Delta'_r$  est dépendante de la solution qui va être remplacée en plus de la solution dont le score de contribution à la diversité est évalué. Or, une fonction objectif est typiquement dépendante uniquement de la solution évaluée.

Notre approximation de la contribution à la diversité met à profit le fait que de nombreux calculs peuvent être réutilisés puisque l'on considère une population de taille constante et que l'on calcule la contribution à la diversité à travers la contribution des caractéristiques des descripteurs, qui sont considérées indépendamment. Cette approximation permet un gain de temps lors de l'optimisation évolutionnaire car elle nous permet de définir un cache des valeurs de contributions à la diversité selon l'état de la population au début de l'étape courante d'optimisation. Selon l'Algorithme 1 qui est présenté en section 2.2.4, chaque étape d'optimisation voit en effet le remplacement de *TailleLot* individus. Pour l'optimisation de la diversité moléculaire, nous calculons au début de chaque étape d'optimisation un cache des valeurs de contribution à la diversité des caractéristiques du descripteur. Ce cache sera utilisé pour le calcul des scores des individus générés lors de l'étape courante. Cela correspond au pré-calcul de  $H(D_i, X)$  pour l'ensemble des caractéristiques  $D_i$  selon l'état de la population  $X$ . Cela correspond également au calcul des quantités  $H_-(D_i, X)$  et  $H_+(D_i, X)$  pour toutes les caractéristiques. Ces deux quantités sont définies ci-dessous.

$$H_-(D_i, X) = -\frac{C_i(X) - 1}{|X|} \log \frac{C_i(X) - 1}{|X|} \quad (3.12)$$

$$H_+(D_i, X) = -\frac{C_i(X) + 1}{|X|} \log \frac{C_i(X) + 1}{|X|} \quad (3.13)$$

L'utilisation des valeurs ainsi stockées permet ainsi un calcul des quantités  $\delta_s(D_i, x, X)$  et  $\delta_a(D_i, x, X)$  durant l'étape courante sans nécessiter aucun nouvel appel à  $H(D_i, x)$ . Elles peuvent en effet être calculées de la façon suivante.

$$\begin{aligned} \delta_s(D_i, x, X) &= H_-(D_i, X) - H(D_i, X) \\ \delta_a(D_i, x, X) &= H_+(D_i, X) - H(D_i, X) \end{aligned} \quad (3.14)$$

**Implémentation** Nous proposons une implémentation de la méthode que nous avons décrite dans cette section, qui est intégrée directement à l'implémentation de notre algorithme évolutionnaire que nous avons déjà évoquée au chapitre précédent. Cette implémentation est disponible et est documentée sur la plateforme GitHub<sup>2</sup>.

## 3.4 Génération d'un jeu de données avec une forte diversité

Dans cette section, nous cherchons à montrer que notre approche permet effectivement de générer une forte diversité moléculaire, à travers la génération d'un jeu de données moléculaires divers de grande taille. Il s'agit d'une partie des travaux dans laquelle ma contribution est minoritaire, et par conséquent nous en proposons ici un résumé. Une présentation détaillée est proposée dans notre article [LEGUY et al. 2021c].

### Génération des données

Nous nous appuyons sur notre approche d'optimisation de la diversité moléculaire pour générer un jeu de données de grande taille contenant une forte diversité moléculaire. Ce jeu de données est généré selon une procédure relativement complexe que nous résumons ici. Elle consiste à générer itérativement et de façon indépendante 6 jeux de données contenant chacun environ 200 000 molécules à l'aide de notre procédure d'optimisation.

---

2. <https://github.com/jules-leguy/EvoMol>

Ces 6 exécutions utilisent des paramètres d'optimisation qui diffèrent légèrement. Parmi ces paramètres, nous pouvons mentionner notamment la population initiale et le choix des descripteurs moléculaires utilisés pour le calcul et l'optimisation de la diversité moléculaire. La population initiale correspond soit à un jeu de données que l'on nomme QMPC9<sup>3</sup>, soit à la molécule de méthane uniquement. QMPC9 est défini comme l'union des jeux de données QM9 et PC9, décrits en section 1.1.4 de ce mémoire. Pour rappel, ces deux jeux de données sont composés de molécules contenant jusqu'à 9 atomes lourds parmi {C, N, O, F}, qui sont associées à leur calcul en DFT. L'union de QM9 et PC9 contient 184 158 molécules différentes.

L'objectif de contribution à la diversité pour les expériences d'optimisation de la contribution à la diversité moléculaire est basé alternativement sur la combinaison des *scaffolds* avec les IFG et sur la combinaison des *scaffolds* génériques avec les IFG. Le but est que la diversité soit exprimée selon un descripteur prenant en compte des aspects structurels (les *scaffolds* ou les *scaffolds* génériques) et un descripteur prenant en compte des aspects locaux (les IFG). L'espace de recherche considéré pour l'optimisation de la diversité moléculaire est également celui des molécules contenant jusqu'à 9 atomes lourds parmi {C, N, O, F}. L'union des solutions générées dans les 6 exécutions forme un jeu de données de grande taille, duquel toutes les molécules comprises dans QMPC9 sont retirées. Cela forme un jeu de données contenant 1 023 624 molécules que l'on nomme Div9. « Div » est une abréviation de « diversité », et 9 fait référence à la taille maximale des molécules en nombre d'atomes lourds.

### Niveaux de validation

La maximisation de la diversité moléculaire favorise la génération de structures moléculaires très instables. Nous tentons dans une certaine mesure de limiter cette instabilité dans les données générées. Ainsi, nous définissons une contrainte sur l'espace de recherche correspondant à la génération de molécules valides selon la mécanique moléculaire. Cela signifie qu'un calcul en mécanique moléculaire de toutes les solutions générées par notre algorithme évolutionnaire est effectué avant leur insertion dans la population. Ils ne peuvent intégrer la population que si le calcul se déroule avec succès, c'est-à-dire qu'il converge et que le graphe moléculaire demeure identique après l'optimisation. Nous avons évoqué cette

---

3. Par mesure de simplicité, nous n'utilisons pas la même terminologie dans ce chapitre que dans les résultats que nous avons publiés au sein de l'article [LEGUY et al. 2021c]. QMPC9 correspond dans l'article au jeu de données OD9\_0, et Div9 correspond dans l'article au jeu de données OD9\_1.



Jeu de données	Validation	Taille	<i>Scaffolds</i> génériques	<i>Scaffolds</i>	IFG
QMPC9	DFT	184 158	3 798	18 850	20 075
Div9	MM	1 023 624	9 163	460 978	461 247
Div9	DFT	250 874	4 858	88 094	124 396

TABLE 3.1 – Taille et nombre de *scaffolds*, de *scaffolds* génériques et d’IFG en fonction du jeu de données et du niveau de validation. QMPC9 est le jeu de données qui sert de base à l’optimisation. Div9 est l’ensemble des données générées excluant les molécules de QMPC9.

procédure en section 1.1.2 de ce mémoire. Cette contrainte est intégrée à notre algorithme évolutionnaire par l’intermédiaire du paramètre  $f_{\text{cont}}$ , que nous avons également utilisé dans le chapitre précédent pour favoriser le réalisme des solutions à travers la définition de plusieurs contraintes.

Nous définissons un second niveau de validation basé sur le calcul en DFT des solutions générées, qui est appliqué à l’ensemble des molécules du jeu de données Div9. Cette procédure permet de filtrer les molécules dont l’optimisation en DFT ne converge pas, ou dont la géométrie converge vers une organisation des atomes qui correspond à une autre molécule. Une procédure comparable est décrite en section 1.1.2 de ce mémoire. Une différence est que les calculs en DFT sont effectués 2 fois de manière indépendante dans le cas présent. Cela représente un effort de calcul considérable, qui n’a été rendu possible que par un projet de calcul collaboratif déposé sur la plateforme BOINC<sup>4</sup>.

## Résultats

La Table 3.1 présente la taille des jeux de données QMPC9 et Div9 en fonction du niveau de validation, et présente également le nombre d’occurrences dans ces jeux de données des trois descripteurs moléculaires qui ont été considérés pour l’optimisation de la diversité. On observe nettement l’effet du filtrage des molécules selon la procédure de validation basée sur la DFT, puisque environ trois quarts des molécules de Div9 sont filtrées. Les données permettent également de faire apparaître et de quantifier la diversité moléculaire générée au sein de Div9. Le jeu de données Div9 en considérant la validation DFT contient environ 1.3 fois plus de molécules que QMPC9, et contient environ 4.7 fois plus de *scaffolds* et 6.2 fois plus d’IFG. En proportion, le nombre de caractéristiques différentes a plus augmenté que la taille du jeu de données, ce qui témoigne d’un succès

4. Voir le projet QuChemPedIA@home <https://quchempedia.univ-angers.fr/athome/>.

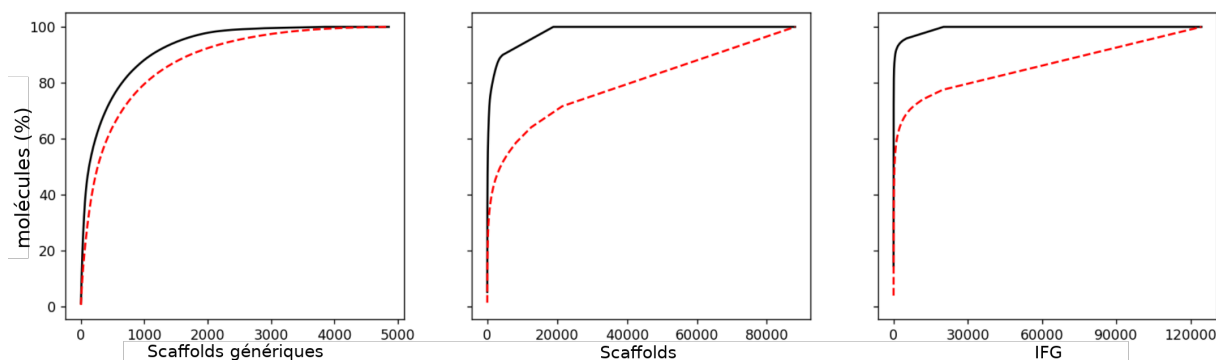


FIGURE 3.2 – Courbes cumulatives du pourcentage de molécules en fonction du nombre de caractéristiques contenues pour les jeux de données QMPC9 (courbe pleine noire) et Div9 (tirets rouges). Pour chaque descripteur, les caractéristiques (axes des abscisses) sont représentées par ordre décroissant de représentativité dans le jeu de données.

de la procédure d'optimisation de la diversité. En revanche le nombre de *scaffolds* génériques évolue nettement moins, puisqu'il est multiplié par 1.3 environ. Cela correspond à une évolution approximativement linéaire entre ces deux jeux de données. Une analyse plus poussée des résultats selon des données que nous ne détaillons pas ici peut montrer qu'un grand nombre de *scaffolds* génériques mène à des divergences de géométrie lors de l'optimisation en DFT. Ainsi, malgré l'objectif de diversité, seule une quantité restreinte de ces caractéristiques peut être présente dans le jeu de données après la validation DFT.

La Figure 3.2 présente une autre visualisation de la concentration de diversité au sein du jeu de données Div9. Nous représentons pour chaque descripteur la courbe cumulative du pourcentage de molécules (axe des ordonnées) qui correspondent aux caractéristiques du descripteur triées par ordre décroissant de représentativité dans le jeu de données (axe des abscisses). Cette figure présente uniquement les résultats après validation DFT. Nous pouvons observer par exemple que pour les *scaffolds* et pour les IFG, une fraction des caractéristiques seulement (extrémité gauche de l'axe des abscisses) est présente dans 80% ou plus des molécules de QMPC9 (courbe noire). Pour le jeu de données que nous avons généré en revanche, on observe que la distribution de ces descripteurs est sensiblement plus homogène, et qu'il faut par exemple plus de 80 000 *scaffolds* différents pour décrire 80% des molécules de Div9. Pour ce jeu de données également, nous observons qu'une poignée de caractéristiques est présente dans une proportion non négligeable des molécules (40% à 60%). Il semble finalement assez attendu que certains descripteurs très communs soient présents très régulièrement. Le gain de diversité peut en fait se mesurer par la présence

de nombreuses caractéristiques qui sont très rares dans le jeu de données. Cette figure permet finalement d’observer une nouvelle fois que les *scaffolds* génériques correspondent au descripteur pour lequel le gain observé de diversité moléculaire est le plus faible. On observe en effet pour ce descripteur que les deux courbes sont relativement proches. En comparaison aux autres descripteurs, nous pouvons également observer qu’il n’existe pas de *scaffolds* génériques qui sont représentés dans une très grande proportion des molécules, pour les deux jeux de données. Pour conclure, l’ensemble de ces résultats permet de valider que notre approche permet effectivement de générer un jeu de données avec une forte diversité et que notre objectif a donc été atteint.

### 3.5 Optimisation conjointe de la diversité et d’une propriété moléculaire

Dans cette section, nous cherchons à étudier l’effet de l’objectif de diversité lors de son optimisation conjointe avec une propriété moléculaire. Dans la littérature, des travaux ont montré qu’associer un algorithme évolutionnaire avec une procédure favorisant un certain niveau de diversité au sein de la population favorise l’exploration de l’espace de recherche et peut permettre d’obtenir des meilleures performances d’optimisation [TSUJIMURA et GEN 1998], y compris pour l’optimisation de propriétés moléculaires [KWON et LEE 2021 ; NIGAM et al. 2020]. Nous concevons une expérience pour chercher à observer et mesurer ce phénomène avec notre approche.

#### Conditions expérimentales

Nous proposons d’étudier l’optimisation conjointe de l’objectif de diversité avec l’objectif correspondant à la maximisation de la valeur de QED [BICKERTON et al. 2012]. Pour rappel, la QED est une estimation de la ressemblance des molécules à des médicaments selon les valeurs de plusieurs propriétés moléculaires. Nous avons déjà étudié l’optimisation des valeurs de QED en l’absence d’objectif de diversité au sein du Chapitre 2. Pour combiner les deux objectifs, nous proposons simplement d’utiliser une fonction objectif définie comme une combinaison linéaire des deux objectifs.

$$f_{\text{obj}}(x_s, x_a, X) = \text{QED}(x_a) + \omega \Delta'_r(x_s, x_a, X) \quad (3.15)$$

Le poids attribué à l’objectif de QED est constant et vaut 1, tandis que nous pouvons

faire varier le poids  $\omega$  attribué à l'objectif de diversité. La fonction permettant de trier les solutions de la population pour sélectionner les individus qui seront remplacés lors de l'étape courante est définie selon l'équation (3.16). En l'absence d'un objectif de contribution à la diversité, la fonction objectif serait utilisée. Comme nous l'avons décrit dans la section précédente, cette sélection doit être exprimée pour l'optimisation de la diversité selon la fonction  $\Delta'_s$ , dont nous considérons l'opposé afin que le tri soit effectué selon un ordre croissant des valeurs.

$$f_{\text{select}}(x, X) = \text{QED}(x) - \omega \Delta'_s(x, X) \quad (3.16)$$

Les expériences sont effectuées selon les mêmes paramètres que les expériences présentées au Chapitre 2 pour l'optimisation des valeurs de QED avec une population de taille 1000. Nous considérons un espace de recherche composé de molécules contenant jusqu'à 38 atomes lourds parmi {C, N, O, F, P, S, Cl, Br}. Les paramètres de l'algorithme sont identiques à l'exception du nombre d'étapes d'optimisation évolutionnaire qui est abaissé à 800, contre 1500 dans les expériences précédentes. Ce choix est simplement lié au fait que nous avons observé qu'il s'agit d'un nombre d'étapes suffisant pour la convergence vers des valeurs élevées de QED. Nous menons des expériences en utilisant indépendamment trois descripteurs de la diversité. Comme les expériences précédentes, nous étudions l'optimisation de la diversité selon les IFG. En revanche, nous n'étudions pas l'optimisation de la diversité selon les *scaffolds* ou les *scaffolds* génériques. La raison est que l'optimisation de la diversité selon l'un de ces descripteurs seulement a peu d'impact, puisqu'il n'existe qu'une unique *scaffold* ou *scaffold* générique par molécule. En tant que descripteur pour le calcul de la diversité, nous utilisons également les *shingles* de rayon 1 ainsi que checkmol dans des expériences indépendantes. Il s'agit de deux descripteurs mettant en évidence des caractéristiques locales, qui sont très génériques mais de petite taille pour les *shingles* et très spécifiques mais de taille variable pour checkmol. En comparaison, les IFG sont de spécificité intermédiaire et également de taille variable.

Pour chacun des descripteurs, nous faisons varier le paramètre  $\omega$  parmi les valeurs {0.1, 1, 10, 100, 1000}. Nous choisissons cet intervalle assez étendu de valeurs car il semble difficile de prédire a priori l'ordre de grandeur des valeurs de  $\Delta'_r$  et  $\Delta'_s$ . Ces dernières sont dépendantes notamment de la taille de la population et de la dimension du descripteur. Or, la dimension effective des descripteurs peut varier beaucoup. Même en considérant que la dimension est connue à l'avance, il est aisé de calculer la plus grande valeur d'entropie  $H(X)$  possible, mais il ne semble pas trivial de déterminer la valeur maximale

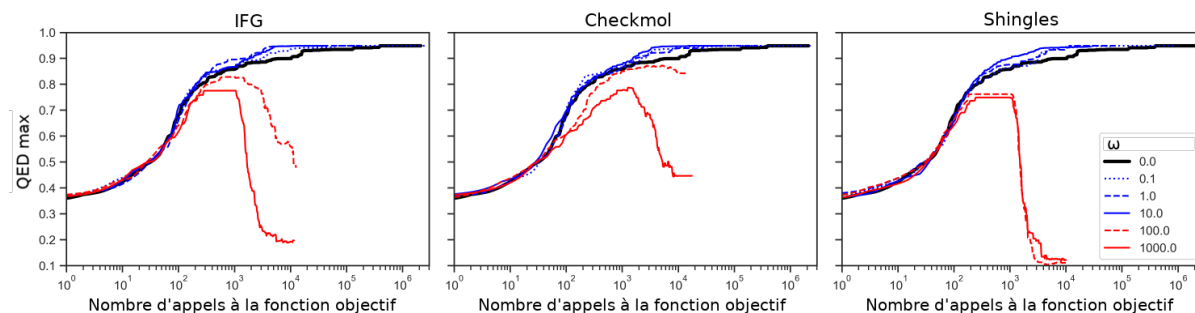


FIGURE 3.3 – Évolution de la valeur maximale de QED au sein de la population exprimée en fonction du nombre d'appels à la fonction objectif, selon le descripteur moléculaire et le poids  $\omega$  attribué à l'objectif de diversité moléculaire. Les valeurs représentées correspondent à la moyenne observée parmi les 10 exécutions.

de contribution à la diversité. Quand bien même elle serait connue, la contribution à la diversité pourra prendre comme valeurs des ordres de grandeur très éloignés en fonction de l'état de la population. Nous effectuons également une expérience de référence en l'absence d'optimisation de la diversité, ce qui correspond à un paramétrage  $\omega = 0$ . Toutes les expériences sont effectuées 10 fois de manière indépendante.

## Résultats

Les résultats sont présentés graphiquement en Figure 3.3. Cette figure représente pour chaque descripteur moléculaire et selon le poids  $\omega$  attribué à l'objectif d'entropie la moyenne parmi les 10 exécutions des valeurs maximales de QED au sein de la population, en fonction du nombre d'appels à la fonction objectif. L'expérience de référence en l'absence d'objectif de diversité est représentée en noir. Indépendamment des descripteurs, on observe deux groupes d'expériences pour lesquelles les performances sont systématiquement supérieures (représentées en bleu) ou inférieures (représentées en rouge) à l'expérience de référence. Les expériences en rouge correspondent à des valeurs élevées du paramètre  $\omega$  (au moins égales à 100), tandis que les expériences en bleu correspondent à des valeurs « modérées » de  $\omega$  (comprises entre 0.1 et 10). Les résultats des expériences en bleu permettent de mettre en évidence que l'objectif de diversité peut effectivement permettre d'optimiser la propriété moléculaire cible de manière plus efficace qu'en l'absence de cet objectif, si la pondération des objectifs respecte un certain équilibre. On observe en effet que des valeurs trop élevées de  $\omega$  (courbes rouges) mènent à une forte

Desc.	$\omega$	0.9		0.94		0.948	
		ERT	Succès	ERT	Succès	ERT	Succès
Aucun	0.0	58994	10	84245	10	131973	9
IFG	0.1	9224	9	15432	10	23420	10
	1.0	<b>1870</b>	10	<b>3524</b>	10	<b>6034</b>	10
	10.0	2485	10	4089	10	8183	9
	100.0	6010	1	-	-	-	-
	1000.0	-	-	-	-	-	-
checkmol	0.1	2781	10	14698	10	63723	10
	1.0	5567	10	9072	10	14136	10
	10.0	<b>1811</b>	10	<b>3978</b>	10	<b>10811</b>	10
	100.0	14748	2	-	-	-	-
	1000.0	-	-	-	-	-	-
shingles	0.1	3271	10	11719	10	16476	10
	1.0	4170	10	<b>5391</b>	10	<b>8636</b>	10
	10.0	<b>1334</b>	10	8129	10	-	-
	100.0	-	-	-	-	-	-
	1000.0	-	-	-	-	-	-

TABLE 3.2 – Mesure d’espérance du coût de l’exécution en nombre d’appels à la fonction objectif (ERT) et nombre de succès parmi 10 exécutions pour l’obtention d’une solution ayant une QED au moins égale aux cibles 0.9, 0.94 et 0.948 selon le descripteur moléculaire et selon le poids  $\omega$  attribué à l’objectif de diversité. Pour chaque stratégie d’initialisation et pour chaque valeur de QED cible, la valeur d’ERT la plus faible est mise en évidence en gras. Les tirets correspondent aux valeurs non définies d’ERT (absence de succès).

dégradation des résultats. Il est intéressant de remarquer que pour ces expériences, les valeurs de QED augmentent pendant un certain temps puis chutent, souvent brutalement et drastiquement. Il semble que l’objectif de contribution à la diversité « prenne le pas » sur l’objectif lié à la QED. Cela signifie que passé un certain seuil, les gains de QED deviennent négligeables par rapport aux valeurs de l’objectif de diversité. Cet effet semble apparaître après qu’il n’existe plus d’individus non définis au sein de la population.

Nous cherchons maintenant à quantifier le gain en termes d’efficacité de la recherche obtenu par l’utilisation de l’objectif de contribution à la diversité. Pour cela, nous représentons en Table 3.2 la mesure d’ERT pour différentes valeurs cibles de QED. Pour rappel, la mesure d’ERT qui est définie en section 1.3.1 de ce mémoire représente l’espérance du coût de l’exécution en nombre d’appels à la fonction objectif pour obtenir une valeur cible de fonction objectif. Ici, nous considérons uniquement l’objectif de QED. Cette table nous permet de comparer les valeurs d’ERT de l’expérience de référence en l’absence de

diversité ( $\omega = 0$ ) avec les résultats obtenus selon des valeurs  $\omega$  non nulles. L'étude des valeurs présentées dans cette table montre que le gain d'efficacité qui peut être apporté par l'objectif de diversité est important puisqu'il est systématiquement compris entre 1 et 2 ordres de grandeur, pour des valeurs élevées de QED (0.9) mais également pour des valeurs très élevées (0.948). Cela peut être observé plus précisément en comparant les résultats de la première ligne de résultats avec les résultats indiqués en gras. Elle montre également que les différents descripteurs peuvent fournir des performances assez proches. Cependant, les valeurs optimales du paramètre  $\omega$  varient en fonction du descripteur et peuvent varier selon la valeur cible, comme nous pouvons l'observer également avec les valeurs représentées en gras. Concernant la proportion de succès parmi les 10 exécutions, nous remarquons que dans la plupart des cas, lorsque la cible peut être atteinte elle est atteinte par toutes les exécutions. Dans les autres cas, la cible n'est jamais ou seulement exceptionnellement atteinte. Cela montre que les résultats sont assez stables pour une valeur de  $\omega$  donnée, ou du moins que les valeurs que nous avons testées ne permettent pas de faire apparaître une transition progressive.

Finalement, nous proposons d'utiliser les arbres d'exploration définis pour notre algorithme évolutionnaire au Chapitre 2 afin de visualiser l'effet de l'objectif de contribution à la diversité sur la recherche au sein de l'espace moléculaire. La Figure 3.4 représente les arbres d'exploration des expériences que nous avons menées, en fonction du descripteur moléculaire (axe horizontal) et de la valeur du paramètre  $\omega$  (axe vertical). La coloration des nœuds dépend des valeurs de QED uniquement et non de l'objectif de diversité. La première ligne représente l'arbre d'exploration obtenu en l'absence d'objectif de diversité. On y observe comme nous l'avons fait lors de la proposition de cette représentation en section 2.3.1 que certains nœuds de l'arbre possèdent de nombreux fils, tandis que de nombreux nœuds n'en possèdent aucun. Nous avons interprété cela comme une marque de la propension de notre algorithme à effectuer une recherche basée prioritairement sur l'intensification de l'espace de recherche. Il est très intéressant d'observer dans les arbres correspondant à l'utilisation d'une valeur non nulle du paramètre  $\omega$  que cette caractéristique disparaît progressivement avec l'augmentation des valeurs de  $\omega$ . Plus cette valeur augmente, plus il semble que tous les nœuds sont susceptibles d'engendrer des descendants. Nous interprétons cela comme une marque de l'exploration de l'espace de recherche qui a lieu lors de l'utilisation de l'objectif de contribution à la diversité.

Cette figure permet de plus de faire apparaître l'effet du compromis entre intensification et exploration permis par le paramètre  $\omega$  sur l'efficacité de la recherche. Cet effet est

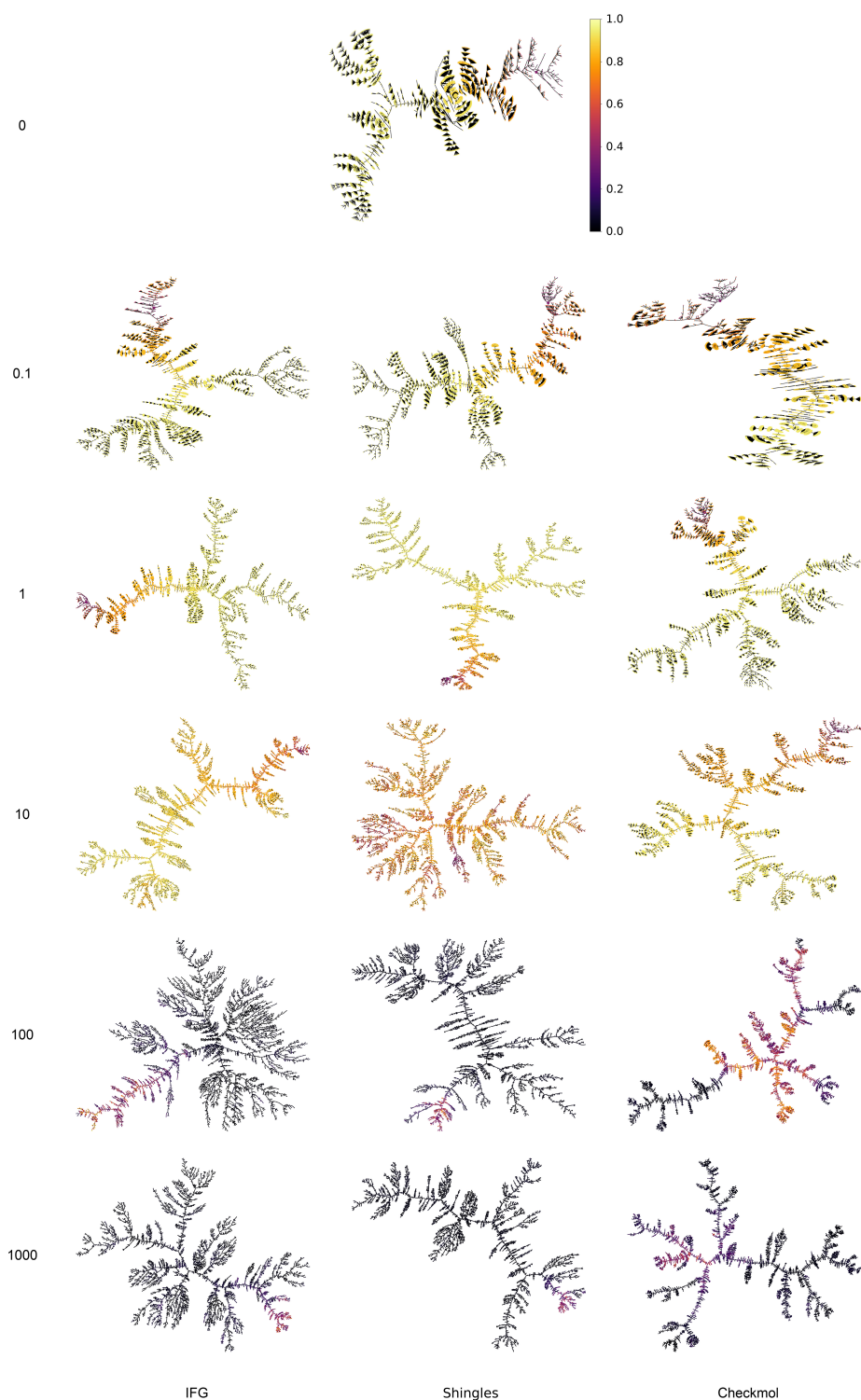


FIGURE 3.4 – Arbres d’exploration pour l’optimisation conjointe de l’objectif de diversité et de l’objectif de QED en fonction du descripteur moléculaire et du poids accordé à l’objectif de diversité. Les solutions sont colorées selon leur valeur de QED.



visible par la coloration des nœuds. Pour les valeurs faibles de  $\omega$  (0 et 0.1) donc avec une forte intensification, une partie non négligeable de l'arbre correspond à de faibles valeurs de QED, et l'on observe aisément le gradient entre les valeurs les plus faibles et les valeurs les plus élevées. Pour les valeurs élevées de  $\omega$  (100 et 1000) donc avec une forte exploration, on observe qu'une partie de l'arbre correspond à des valeurs faibles à modérées de QED, et que l'autre partie correspond à des valeurs très faibles représentées en noir. En lien avec la Figure 3.3 qui montre l'évolution des valeurs de QED lors de la recherche, nous pouvons déduire que la première partie de l'arbre correspond au début de la recherche et que la partie en noir correspond à la fin de la recherche, pour laquelle l'objectif de diversité est prépondérant. Finalement, pour les valeurs modérées de  $\omega$  (1 et 10) qui correspondent à un compromis raisonnable entre intensification et exploration, nous observons que dans l'ensemble la coloration des arbres est plus homogène. Nous observons le gradient entre les valeurs de QED les plus faibles et les plus élevées, mais la partie de l'arbre avec des valeurs faibles semble plus restreinte. Nous pouvons mettre cela en relation avec l'effet déjà observé graphiquement et numériquement, que l'objectif de diversité permet dans ce cas une convergence plus rapide vers des valeurs hautes de QED.

## 3.6 Conclusion et perspectives

Dans ce chapitre, nous nous intéressons à la faible diversité moléculaire qui a été observée dans le jeu de données QM9. Cette faible diversité est susceptible d'altérer les performances des modèles d'apprentissage artificiel de propriétés moléculaires qui l'utiliseraient en tant que jeu de données d'entraînement. Or, de tels modèles pourraient favoriser la découverte de molécules satisfaisant des propriétés moléculaires dont le calcul est coûteux, en particulier pour la chimie des matériaux moléculaires organiques. Ainsi, la diversité des jeux de données moléculaires est un enjeu important pour le domaine de la génération moléculaire. Dans ce chapitre, nous proposons de le traiter en proposant une méthode d'optimisation de la diversité moléculaire, basée sur l'algorithme évolutionnaire que nous avons présenté dans le chapitre précédent.

Nous utilisons la mesure d'entropie de Shannon que nous associons à un descripteur moléculaire afin de définir une mesure de la diversité moléculaire. Nous discutons du fait que pour maximiser cette mesure à l'aide d'une méthode d'optimisation moléculaire, il faut définir une mesure de contribution à la diversité moléculaire de l'ensemble du jeu de données. Or, pour maximiser la diversité d'un jeu de données de grande taille, cela

implique un coût calculatoire trop important. Nous proposons donc une approximation efficace de cette mesure, qui met à profit le fait que les descripteurs moléculaires d'intérêt sont creux et qui suppose l'utilisation d'un jeu de données moléculaires de taille constante. Nous implémentons cette méthode au sein de notre algorithme évolutionnaire.

Par la suite, nous montrons que notre approche permet effectivement la génération d'un jeu de données moléculaires de très grande taille possédant une forte diversité moléculaire. Nous montrons également qu'il est possible d'optimiser conjointement l'objectif de diversité avec une propriété moléculaire cible, et que cela peut permettre une optimisation plus efficace de cette propriété cible. Nous menons une étude des performances d'optimisation en fonction du poids attribué à l'objectif de diversité. Nous observons que des valeurs très élevées de ce paramètre de poids défavorisent l'optimisation de la propriété cible, mais que des valeurs intermédiaires permettent un gain d'efficacité important. Nous mesurons un gain en nombre d'appels à la fonction objectif d'un à deux ordres de grandeur. Nous expliquons l'amélioration des performances d'optimisation par le fait que l'objectif de contribution à la diversité favorise l'exploration de l'espace de recherche. Nous pouvons supposer que ce gain est d'autant plus important que nous avons observé que notre algorithme évolutionnaire possède une prédisposition à l'intensification plutôt qu'à l'exploration de l'espace de recherche. Nous pouvons identifier une limite de notre approche en particulier, qui est que le paramètre de poids attribué à l'objectif de diversité est dépendant de la propriété cible ainsi que du descripteur moléculaire, et qu'il semble difficile d'identifier a priori des valeurs pertinentes pour ce paramètre. Nous pourrions envisager de définir une approche pour sélectionner dynamiquement la valeur de ce paramètre. Nous pouvons de plus imaginer que la valeur optimale de ce paramètre dépend de l'état de la recherche, et qu'une sélection dynamique pourrait donc également améliorer les performances d'optimisation.



# OPTIMISATION BOÎTE-NOIRE GUIDÉE PAR UN MODÈLE D'APPRENTISSAGE DE PROPRIÉTÉS MOLÉCULAIRES

---

## Sommaire

---

<b>4.1</b>	<b>Introduction</b>	<b>161</b>
<b>4.2</b>	<b>Optimisation basée sur un modèle de substitution</b>	<b>163</b>
4.2.1	Contexte	163
4.2.2	Algorithme	164
4.2.3	Sélection du jeu de données initial	165
4.2.4	Modèle de substitution et fonction de mérite	168
4.2.5	Optimisation de la fonction de mérite	171
<b>4.3</b>	<b>Optimisation de graphes moléculaires basée sur un modèle de substitution</b>	<b>172</b>
4.3.1	Méthode	172
4.3.2	Apprentissage du modèle de substitution	175
4.3.3	Implémentation	176
4.3.4	Travaux liés	178
<b>4.4</b>	<b>Étude de l'optimisation d'une propriété peu coûteuse</b>	<b>179</b>
4.4.1	Apprentissage et évaluation du modèle de substitution	179
4.4.2	Évaluation de notre méthode d'optimisation	188
4.4.3	Conclusion : apprentissage et optimisation de la QED	206
<b>4.5</b>	<b>Optimisation d'une propriété électronique</b>	<b>207</b>
4.5.1	Apprentissage et évaluation du modèle de substitution	208
4.5.2	Évaluation de notre méthode d'optimisation	217
4.5.3	Conclusion : apprentissage et optimisation de l'énergie HOMO	221
<b>4.6</b>	<b>Conclusion et perspectives</b>	<b>221</b>

---

Ce chapitre fait l'objet des publications et communications suivantes.

[LEGUY et al. 2021a]

[LEGUY et al. 2021b]



Dans ce chapitre, nous proposons une méthode d’optimisation boîte-noire basée sur un modèle de substitution, que nous concevons pour l’optimisation efficace de propriétés moléculaires coûteuses. Le modèle de substitution est un algorithme d’apprentissage artificiel qui permet d’estimer à faible coût les valeurs de la propriété cible. Il est utilisé au sein d’une procédure d’optimisation pour sélectionner des solutions prometteuses, qui sont intégrées au sein des données d’entraînement afin que le modèle de substitution s’adapte progressivement à la propriété cible. Notre approche est basée sur un modèle de substitution GPR, et notre algorithme évolutionnaire EvoMol est utilisé pour obtenir des candidats à partir du modèle de substitution. Nous étudions notre approche de façon approfondie pour l’optimisation des valeurs de QED, puis nous montrons qu’elle permet l’optimisation d’une propriété électronique coûteuse. Dans les deux cas, nous montrons que notre algorithme permet une optimisation efficace de la propriété cible.

## 4.1 Introduction

Nous avons proposé dans le Chapitre 2 de ce mémoire une méthode d’optimisation moléculaire générique basée sur un algorithme évolutionnaire. Nous avons montré que notre algorithme permet d’optimiser avec succès différentes propriétés moléculaires. Nous avons observé cependant que cela nécessite parfois de nombreux appels à la fonction objectif. Nous souhaitons dans le cadre de cette thèse développer des approches pour l’optimisation de propriétés moléculaires pertinentes dans le domaine de la chimie des matériaux moléculaires organiques. Or, ces propriétés dépendent souvent de calculs coûteux en mécanique quantique. Nous cherchons dans ce chapitre à développer une approche permettant de limiter le coût de l’optimisation de ces propriétés.

Lors de la dernière décennie, de nombreux travaux ont proposé d’utiliser des méthodes d’apprentissage artificiel pour l’estimation de ces propriétés à moindre coût. Nous pouvons citer entre autres l’utilisation de modèles de régression par processus gaussien (abrégiés GPR en anglais) [BARTÓK et al. 2017; DERINGER et al. 2021] ou des architectures d’apprentissage profond comme le modèle SchNet [SCHÜTT et al. 2018] (voir le Chapitre 1).

La capacité de généralisation des modèles de prédiction dépend entre autres de la qualité des jeux de données d’entraînement. Nous avons observé dans des travaux antérieurs un manque de diversité moléculaire dans le jeu de données QM9 [GLAVATSKIKH et al. 2019], communément utilisé en tant que jeu de données d’entraînement pour l’apprentissage de propriétés moléculaires électroniques. Cela limite les capacités de généralisation

des modèles pour lesquels il sert de jeu d'entraînement. Par extension, on peut craindre que cela limite également les performances des méthodes de génération moléculaire basées sur ces modèles. On s'attend en effet à ce qu'un tel modèle soit susceptible de prédire les valeurs de propriétés de points quelconques de l'espace de recherche, et qu'il puisse prédire les valeurs extrêmes de propriétés, dont les exemples d'entraînement potentiels sont très rares.

Une possibilité est de concevoir un jeu de données adapté à l'entraînement d'un modèle de prédiction d'une propriété donnée. On ne cherche alors pas un jeu de données représentatif de l'espace de recherche complet, mais un jeu de données pertinent pour l'apprentissage d'une propriété donnée tout en limitant les exemples d'apprentissage nécessaires. Cela correspond au champ de l'apprentissage actif de propriétés moléculaires, étudié entre autres par [GUBAEV et al. 2018].

Une autre approche consiste à associer un modèle d'apprentissage à une procédure d'optimisation, dans le cadre de l'optimisation basée sur un modèle de substitution [VU et al. 2017]. Il s'agit de l'approche que nous proposons d'utiliser au sein de ce chapitre. Elle est définie notamment pour l'optimisation de fonctions boîte-noires coûteuses. Dans ce cadre, les procédures d'optimisation et d'apprentissage sont effectuées conjointement. Le modèle de substitution est utilisé pour obtenir des solutions candidates à moindre coût, puis ces solutions sont évaluées par la fonction objectif et intégrées dans les données d'apprentissage. L'objectif est d'explorer l'espace de recherche de manière efficace, tout en intégrant la connaissance obtenue dans les données d'entraînement, afin que ces dernières soient pertinentes par rapport à la propriété cible. Cette procédure est effectuée itérativement.

Dans ce chapitre, nous proposons une approche d'optimisation boîte-noire basée sur un modèle de substitution pour l'optimisation de propriétés moléculaires. Nous nommons notre approche BBOMol. « BBO » correspond à l'acronyme de « *black-box optimization* », qui signifie optimisation boîte-noire. Notre approche est basée sur un modèle de substitution GPR dont l'estimation de l'incertitude des prédictions est mise à profit pour améliorer l'efficacité de la recherche. L'optimisation basée sur un modèle de substitution est typiquement étudiée pour des problèmes d'optimisation continue. Un certain nombre de problèmes se posent pour adapter ce cadre à l'espace de recherche des graphes moléculaires. Dans ce chapitre, nous présentons d'abord le cadre théorique. Puis, nous présentons les choix que nous avons effectués pour définir notre approche. Par la suite, nous menons une étude des modèles de substitution et des performances d'optimisation de BBOMol pour

l'optimisation des valeurs de QED, dont le faible coût de calcul nous permet de mener une étude détaillée. Finalement, nous étudions notre approche dans le cadre de l'optimisation d'une propriété électronique coûteuse. Nous démontrons l'efficacité de notre approche, que nous comparons notamment aux résultats de notre algorithme évolutionnaire. Nous discutons de l'influence des différents paramètres, et en particulier de la sélection du jeu de données initial.

## 4.2 Optimisation basée sur un modèle de substitution

### 4.2.1 Contexte

L'optimisation de fonctions boîte-noire correspond à l'optimisation de fonctions dont on ne connaît pas de définition analytique ou dont on n'utilise pas d'informations issues de la définition (voir la section 1.3.1 de ce mémoire). Une fonction boîte-noire peut seulement être évaluée en différents points de l'espace de recherche. Lorsque l'on considère la génération de molécules satisfaisant des propriétés moléculaires comme un problème d'optimisation moléculaire, cela correspond à un problème d'optimisation boîte-noire. Plusieurs approches permettent d'aborder ce genre de problème. Les métaheuristiques en font partie, et nous avons d'ailleurs proposé au Chapitre 2 un algorithme évolutionnaire pour l'optimisation de propriétés moléculaires. Nous avons cependant observé que notre approche est susceptible de nécessiter un nombre important d'appels à la fonction objectif. Cela est problématique pour l'optimisation de fonctions objectif coûteuses, comme le sont les propriétés électroniques dont l'estimation dépend de calculs DFT.

Des approches sont spécialement définies pour l'optimisation de fonctions boîte-noire coûteuses. On peut les séparer entre méthodes de recherche locale et méthodes de recherche globale. Les méthodes de recherche locale offrent des garanties fortes de convergence vers un optimum local de la fonction objectif, à partir d'un point quelconque de l'espace de recherche. Ces méthodes font partie d'un domaine de recherche nommé « *derivative-free optimization* » (optimisation sans dérivée) [RIOS et SAHINIDIS 2013]. Les méthodes de recherche globale sont susceptibles de trouver des optimaux sur l'ensemble du domaine de définition de la fonction objectif, mais offrent peu de preuves de convergence. Les métaheuristiques sont la plupart du temps des méthodes de recherche globale, mais sont peu adaptées à l'optimisation de fonctions coûteuses. Dans ce chapitre, nous nous intéresserons



exclusivement aux méthodes de recherche globale, qui correspondent mieux à nos intentions pour les problèmes d'optimisation moléculaire. Nous nous intéressons en particulier au domaine de l'optimisation basée sur un modèle de substitution de la fonction objectif [QUEIPO et al. 2005 ; VU et al. 2017]. Dans ce cadre, le modèle de substitution est global, c'est-à-dire qu'il est défini dans le but de prédire les valeurs de la fonction objectif sur l'ensemble de son domaine de définition. Il s'agit d'une approche définie principalement pour des problèmes d'optimisation continue. Nous allons cependant montrer dans la suite de ce chapitre qu'elle peut être adaptée à des problèmes d'optimisation combinatoire de graphes moléculaires. Dans la suite de cette section, nous présentons plus en détail le fonctionnement de cette approche. Nous nous appuyons sur la classification proposée par [VU et al. 2017].

### 4.2.2 Algorithme

---

**Algorithme 3** Algorithme générique pour l'optim. basée sur un modèle de substitution.

---

**entrée:**  $f$  la fonction objectif boîte-noire,

$s$  le modèle de substitution de  $f$

$k \leftarrow 0$

$X \leftarrow \emptyset$

▷  $X$  : jeu de données complet

$X_k \leftarrow$  sélection des données initiales

▷  $X_k$  : données générées à l'étape courante

**tant que** le critère d'arrêt n'est pas atteint **faire**

$k \leftarrow k + 1$

$X \leftarrow X \cup X_{k-1}$

évaluation des solutions de  $X_{k-1}$  par  $f$

entraînement du modèle de substitution  $s$  sur les données  $\{(x, f(x)), \forall x \in X\}$

$X_k \leftarrow$  recherche de solutions candidates à l'aide de  $s$

**fin tant que**

---

Le fonctionnement général des méthodes d'optimisation boîte-noire basées sur un modèle de substitution est décrit dans l'Algorithme 3. Avant le début de la procédure d'optimisation, le jeu de données initial doit être sélectionné. Cette sélection, appelée *design of experiments* en anglais, est importante puisqu'elle détermine la connaissance initiale du modèle de substitution sur la fonction pour laquelle il sert d'estimateur. Notons que nous faisons la distinction entre le jeu de données  $X_k$  qui contient un ensemble de points de l'espace de recherche sélectionnés à l'étape  $k$ , et  $X$  l'ensemble des solutions générées à toutes les étapes précédentes. Après la phase d'initialisation, la procédure d'optimisation est amorcée jusqu'à ce qu'un critère d'arrêt soit atteint (typiquement, un budget d'appels

à la fonction objectif). Au début de chaque étape, les solutions générées à l'étape précédente sont évaluées par la fonction objectif  $f$ . Ces valeurs sont gardées en cache afin que leur calcul ne soit plus nécessaire dans la suite de l'exécution. Ensuite, le modèle de substitution est entraîné à prédire les valeurs de la fonction objectif, à partir de l'ensemble des données connues. Finalement, il est utilisé au sein d'une procédure qui permettra d'obtenir un ensemble de solutions candidates  $X_k$  pour l'étape courante. Cette procédure est dépendante du contexte et sera décrite dans notre cas en section 4.3.1. Dans cet algorithme, le modèle de substitution  $s$  est utilisé à chaque étape pour rechercher un ensemble restreint de solutions candidates prometteuses, sans nécessiter d'appels à  $f$ . L'évaluation de  $s$  étant censée être nettement moins coûteuse que celle de  $f$ , ces candidats sont sélectionnés avec une économie de coût. Précisons toutefois qu'à chaque étape, ils sont évalués de manière exacte par  $f$ . Cela permet d'affiner la connaissance du modèle de substitution au cours de la recherche.

### 4.2.3 Sélection du jeu de données initial

Le jeu de données initial est utilisé pour entraîner le modèle de substitution au début de la procédure d'optimisation. Le modèle de substitution étant censé prédire les valeurs de  $f$  sur la totalité de son ensemble de définition, il est attendu que le jeu de données d'entraînement soit le plus représentatif possible de l'espace de recherche. On cherche pour cela à générer une répartition « homogène » des points dans l'espace de recherche. Le sens que prend cette notion d'homogénéité dépend de l'approche de résolution de ce problème. Nous présentons ici plusieurs de ces approches. Nous insistons sur leurs avantages relatifs en fonction des caractéristiques de l'espace de recherche.

**Sélection par construction** La première classe est adaptée aux espaces de recherche définis au sein de  $\mathbb{R}^n$  ou  $\mathbb{Z}^n$  et consiste à construire la sélection en divisant l'espace de recherche en un ensemble de cellules de mêmes tailles, puis à placer un point au centre de chacune ou d'une partie de ces cellules. La sélection exhaustive des cellules implique que la taille du jeu de données initial est liée de façon exponentielle au nombre de dimensions. Cela peut rapidement mener à une explosion combinatoire. De plus, l'ensemble des points devra être évalué par la fonction coûteuse  $f$ . Certaines stratégies, comme le Latin Hypercube Sampling, permettent de sélectionner un sous-ensemble de ces points dont la cardinalité est indépendante du nombre de dimensions [MCKAY et al. 1979]. Notons que cette sélection implique de résoudre un nouveau problème d'optimisation combinatoire

pour conserver l'uniformité de la sélection. Par construction, cette approche est adaptée pour les espaces de recherche inclus dans  $\mathbb{R}^n$  ou  $\mathbb{Z}^n$ , associés à un ensemble de contraintes fixant pour chaque dimension une borne minimale et maximale. Il semble très difficile de transposer une approche de ce type à des objets que l'on considère sous la forme de graphes, comme des molécules. Notons qu'il en est de même pour d'autres représentations potentielles des molécules, comme les SMILES ou la représentation sous la forme d'un nuage d'atomes.

**Sélection basée sur une mesure de distance** Une seconde approche, proposée la première fois par [JOHNSON et al. 1990], consiste à définir un problème d'optimisation combinatoire basé sur une mesure de distance  $d$  entre deux points de l'espace de recherche. Deux façons d'aborder le problème sont proposées. La première, nommée *minimax*, consiste à rechercher un ensemble de points  $S$  tel que tout point de l'espace de recherche  $D$  ( $S \subseteq D$ ) doit être proche d'au moins un point de  $S$ . Si l'on note  $d(x, S) = \min_{y \in S} d(x, y)$  la distance minimale entre un point  $x$  et un point de l'ensemble  $S$ , on cherche donc un ensemble  $S^*$  défini de la façon suivante.

$$S^* = \operatorname{argmin}_S \max_{x \in D} d(x, S) \quad (4.1)$$

La stratégie *minimax* consiste donc à minimiser la distance maximale entre tout point de l'espace  $D$  et un point de la sélection  $S$ . L'inconvénient de cette approche est que le terme  $d(x, S)$  doit être calculé pour tout point  $x$  de l'espace de recherche, ce qui risque de ne pas être envisageable dans beaucoup de cas. Une autre façon complémentaire d'aborder le problème est également proposée. Il s'agit de la stratégie *maximin*, qui consiste à rechercher un ensemble  $S$  tel que les points contenus doivent être éloignés les uns des autres. Cette stratégie consiste à rechercher un ensemble  $S^*$  défini de la façon suivante.

$$S^* = \operatorname{argmax}_S \min_{x, y \in S} d(x, y) \quad (4.2)$$

La stratégie *maximin* consiste donc à maximiser la distance minimale entre toute paire de points de  $S$ , ce qui revient effectivement à rechercher un ensemble de points éloignés les uns des autres. Elle peut être évaluée de façon plus raisonnable car son calcul ne dépend que des points qui sont sélectionnés. Un avantage de la sélection par mesure de distance par rapport à la sélection par construction présentée précédemment est qu'elle ne dépend pas d'un type d'espace de recherche en particulier. Elle peut être adaptée à tout type

de problème, si une mesure de distance pertinente peut être définie, et qu'une méthode d'optimisation adaptée peut être utilisée pour résoudre le problème d'optimisation défini. Il est envisageable de transposer cette stratégie à l'espace moléculaire, puisqu'il existe des mesures de distances entre des graphes moléculaires, telle que la distance de Tanimoto basée sur des caractéristiques ECFP4 que nous avons présentée au Chapitre 1. Toutefois, l'objectif défini ici évalue l'ensemble  $S$  des solutions sélectionnées et non chaque solution indépendamment. Une méthode d'optimisation moléculaire classique dont les solutions sont des molécules (et non des ensembles de molécules) telle que l'algorithme évolutionnaire que nous avons proposé au Chapitre 2 ne peut donc pas résoudre directement ce problème d'optimisation. Il faudrait pour cela envisager une relaxation du problème d'optimisation, ou une adaptation de la méthode d'optimisation.

**Sélection statistique** Une dernière approche consiste à considérer le problème de sélection du jeu de données initial de façon statistique. On considère alors un processus aléatoire permettant de prédire les valeurs de  $f$ , tel qu'un processus Gaussien (voir la section 1.2 de ce mémoire). Il est possible de définir différents critères d'optimalité pour sélectionner un jeu de données représentant au mieux l'espace de recherche [JOHNSON et al. 1990]. L'un de ces critères d'optimalité, connu sous le nom de « *D-optimality* », consiste à maximiser le déterminant de la matrice de covariance du processus stochastique. Cela revient à minimiser la corrélation entre les variables aléatoires représentant les points sélectionnés. Comme la sélection basée sur une mesure de distance, la sélection statistique n'est pas dépendante d'un type d'espace de recherche en particulier, car elle s'appuie sur une fonction définissant la covariance entre les variables aléatoires représentant les points de l'espace de recherche. Elle peut donc être utilisée pour tout type de problème, sous réserve de pouvoir définir une fonction de covariance et une méthode d'optimisation pertinentes. Il est ainsi envisageable d'appliquer cette stratégie à l'espace moléculaire. Nous montrerons par ailleurs dans la suite de ce chapitre qu'un processus Gaussien, basé sur une fonction de covariance, peut être défini sur l'espace moléculaire. Toutefois, comme la stratégie *maximin*, l'objectif défini ici évalue l'ensemble des solutions sélectionnées et non les solutions indépendamment. Il ne peut donc pas être traité directement par une méthode classique d'optimisation moléculaire.

**Résolution du problème d'optimisation** Nous avons évoqué plusieurs stratégies pour la sélection du jeu de données initial. Deux d'entre elles sont susceptibles d'être

appliquées à l'espace moléculaire, mais elles définissent un objectif évaluant la sélection de solutions, et la qualité de chaque solution au sein de cette sélection. Cela rend plus complexe la résolution du problème d'optimisation sous-jacent, puisque les méthodes d'optimisation moléculaire dépendent classiquement d'une fonction objectif évaluant chaque solution indépendamment des autres. Nous pouvons mentionner deux pistes pour résoudre ce problème. La première est d'effectuer une relaxation du problème en considérant que toutes les solutions de la sélection à l'exception d'une sont fixées. Ainsi, la fonction objectif précédemment définie est adaptée à la recherche d'une valeur optimale pour cette solution variable. Il est envisageable d'effectuer cette procédure d'optimisation plusieurs fois en faisant varier la solution non fixe. L'autre approche est de définir une mesure de *contribution* à l'objectif total, comme nous l'avons fait au sein du Chapitre 3 pour la diversité moléculaire. Cette mesure pourrait être utilisée en tant que fonction objectif, mais il est probable qu'en l'absence d'une approximation efficace de cette mesure les calculs soient très coûteux. Finalement, la sélection d'un jeu de données initial pour l'optimisation de propriétés moléculaires est un problème non trivial.

#### 4.2.4 Modèle de substitution et fonction de mérite

À chaque étape de l'Algorithme 3, les données sur l'espace de recherche et les valeurs calculées de  $f$  sont utilisées pour entraîner le modèle de substitution. Par la suite, ce modèle est utilisé au sein d'une procédure d'optimisation pour produire des candidats prometteurs. Généralement, ces candidats ne sont pas obtenus par optimisation du modèle de substitution directement mais plutôt par optimisation d'une *fonction de mérite*  $m_s$ , qui dépend du modèle de substitution  $s$ . La recherche de candidats correspond donc à un sous-problème d'optimisation qui est exprimé de la façon suivante. Pour ce problème, nous ne considérons pas la recherche d'un optimum global de  $m_s$  mais plutôt la recherche d'un ensemble de solutions approchées  $x^*$ .

$$x^* = \operatorname{argmax}_{x \in D} m_s(x) \quad (4.3)$$

A priori, tout type de modèle d'apprentissage peut être utilisé en tant que modèle de substitution. Cependant, les fonctions de mérite peuvent dépendre de certaines caractéristiques du modèle de substitution, qui vont donc conditionner son choix. Nous présentons ici plusieurs de ces fonctions de mérite, en nous attardant notamment sur les fonctions de mérite probabilistes que nous utiliserons dans la suite de ce chapitre.

**Optimisation du modèle de substitution** L'approche la plus directe pour résoudre un problème d'optimisation boîte-noire à l'aide d'un modèle de substitution est d'optimiser directement les valeurs de ce modèle. Cela revient à considérer que la fonction de mérite est égale au modèle de substitution  $s$ . L'avantage majeur de cette approche est qu'aucune hypothèse n'est effectuée sur  $s$ , et donc que tout type de modèle d'apprentissage peut être utilisé. Cela revient cependant à accorder une confiance absolue au modèle de substitution, qui n'est toutefois qu'une approximation de  $f$  dépendant de l'état de la connaissance courante. [JONES 2001] montre que dans beaucoup de cas cette approche peut ne pas converger vers un optimum, même local.

**Modèles et fonctions probabilistes** En supposant que le modèle de substitution peut prédire pour chaque point  $x$  de l'espace une variable aléatoire représentant une distribution des valeurs probables de  $f(x)$  selon  $s$ , on peut chercher à tirer partie de cette information supplémentaire pour le guidage de la recherche. Cette distribution peut être interprétée comme une mesure d'incertitude. Intuitivement, l'incertitude est faible si la distribution est resserrée autour de la valeur prédite, et forte si elle est étendue. Plusieurs fonctions de mérite peuvent alors être définies. Nous en présentons deux, qui sont très populaires [JONES 2001]. La première, que l'on nomme POI (acronyme de « *probability of improvement* ») représente la probabilité d'amélioration par rapport au meilleur point connu  $x^+$ . Cette fonction est exprimée en équation (4.4), pour un problème de maximisation. Le paramètre  $\xi \in \mathbb{R}_+$  permet de favoriser l'exploration de l'espace de recherche. Plus  $\xi$  est élevé, plus la recherche va être menée vers des zones de fortes incertitudes. Intuitivement, cela est expliqué par le fait que ce paramètre va artificiellement exagérer la qualité des meilleurs points connus. Cela limite la probabilité que des solutions améliorantes soient dans le voisinage proche des solutions connues selon le modèle et favorise la sélection de candidats dans des zones plus éloignées de forte incertitude. POI permet d'estimer la probabilité d'une amélioration, mais ne quantifie pas cette amélioration. La fonction d'« *expected improvement* » (EI) a été proposée comme une alternative, qui prend en compte une notion d'« espérance d'amélioration » [JONES et al. 1998]. Ainsi, si deux points ont la même probabilité d'améliorer le meilleur point connu, cette mesure permettra de sélectionner celui produisant l'amélioration la plus grande selon  $s$ . EI est définie en équation (4.5) pour un problème de maximisation, avec  $Y = s(x)$  la variable aléatoire correspondant à la prédiction du modèle de substitution au point  $x$ . Comme pour POI,  $\xi$  est un paramètre positif permettant de favoriser l'exploration de l'espace de recherche.

$$\text{POI}(x) = \mathbb{P}(f(x) \geq f(x^+) + \xi) \quad (4.4)$$

$$\text{EI}(x) = \mathbb{E}[\max(Y - f(x^+) - \xi, 0)] \quad (4.5)$$

Ces fonctions de mérite sont très souvent utilisées avec un modèle de régression par processus gaussien (GPR), dont le fonctionnement est décrit en détail dans la section 1.2 de ce mémoire. Les GPR ont la particularité de prédire une distribution gaussienne paramétrée par une moyenne  $\mu(x)$  et un écart-type  $\sigma(x)$  pour tout point  $x$  de l'espace de recherche. Nous notons cette relation  $Y \sim \mathcal{N}(\mu(x), \sigma(x)^2)$ . POI et EI peuvent être calculées de façon analytique lorsque  $s(x)$  suit une loi normale selon les équations (4.6) et (4.7) respectivement. Ces deux équations dépendent d'un terme  $Z$  défini en équation (4.8).  $\Phi$  et  $\phi$  correspondent respectivement à la fonction de distribution cumulative et à la fonction de densité de la loi normale centrée et réduite.

$$\text{POI}(x) = \Phi(Z) \quad (4.6)$$

$$\text{EI}(x) = (\mu(x) - f(x^+) - \xi)\Phi(Z) + \sigma(x)\phi(Z) \quad (4.7)$$

$$Z = \frac{\mu(x) - f(x^+) - \xi}{\sigma(x)} \quad (4.8)$$

**Interpolation par RBF** Il existe d'autres approches qui ne sont pas basées sur un modèle ou une fonction de mérite statistique. Dans la littérature, des modèles d'interpolation basés sur des fonctions de base radiale sont notamment utilisés. Pour rappel, les fonctions à base radiale sont généralement nommées RBF, ce qui correspond à leur acronyme en anglais (voir la section 1.2.2 de ce mémoire). Nous ne rentrerons pas dans le détail du fonctionnement de ces modèles, décrit précisément par [GUTMANN 2001]. Brièvement, l'interpolation par RBF s'exprime comme la somme d'une combinaison linéaire de polynômes et de fonctions RBF centrées sur les différents points connus de l'espace de recherche. Ce genre de modèle peut être associé à des fonctions de mérite spécifiques, telle que la mesure de « *bumpiness* » proposée par [GUTMANN 2001]. Il s'agit d'un concept qui permet de représenter à quel point une fonction est « bosselée ». Cette mesure est utilisée pour définir une fonction de mérite, qui permet de sélectionner un point  $x$  prédit

à une valeur  $s(x)$  prometteuse et dont l'ajout dans le jeu de données d'entraînement minimise cette mesure pour le modèle de substitution de l'étape suivante. L'intuition est que moins le modèle de substitution possède de « bosses », plus il est susceptible d'être fidèle à  $f$ , et donc que l'estimation de  $s(x)$  est correcte. À notre connaissance, ce genre d'approche n'a pas été étudié dans le cadre de la prédiction ou de l'optimisation de propriétés moléculaires.

Pour l'optimisation de propriétés moléculaires, nous choisirons d'utiliser l'approche probabiliste basée sur un modèle GPR et les fonctions de mérite POI et EI. Des travaux ont montré l'efficacité des modèles GPR pour la prédiction de propriétés moléculaires électroniques [DERINGER et al. 2021]. L'association de ces modèles avec les fonctions de mérite probabilistes permet un certain contrôle sur la façon dont la recherche est menée, notamment via le compromis entre intensification et exploration réglé par le paramètre  $\xi$ . Finalement, notons que lorsque le modèle de substitution est probabiliste, comme les GPR, il existe une intersection entre le champ de l'optimisation boîte-noire et le champ de l'optimisation bayésienne [SHAHRIARI et al. 2016]. Dans la suite de ce chapitre, nous continuerons à parler systématiquement d'optimisation boîte-noire basée sur un modèle de substitution. Il s'agit d'une dénomination plus générique que nous considérons subjectivement comme s'intégrant mieux avec le domaine de l'optimisation combinatoire, développé dans les chapitres précédents. Il aurait toutefois également été raisonnable de parler d'optimisation bayésienne lorsque des fonctions de mérite probabilistes sont utilisées.

### 4.2.5 Optimisation de la fonction de mérite

À chaque étape d'optimisation de l'Algorithme 3, les solutions candidates sont obtenues par maximisation de la fonction de mérite  $m_s$ . En fonction des caractéristiques de l'espace de recherche, différentes méthodes d'optimisation peuvent être utilisées. Si un ensemble de points prédéfini est disponible, il est possible simplement de sélectionner le sous-ensemble de ces points maximisant  $m_s$  [REGIS et SHOEMAKER 2007]. Cela nécessite toutefois que l'ensemble soit de taille raisonnable par rapport au coût de calcul de  $m_s$ , et ne constitue pas à proprement parler une recherche dans l'espace des solutions. Dans certains cas, il est possible d'utiliser des méthodes de recherche exacte, qui garantiront l'optimalité des candidats selon  $m_s$ . Il s'agit de l'approche utilisée par [JONES et al. 1998] pour maximiser la fonction EI dans un espace de recherche compris dans  $\mathbb{R}^n$ . Lorsque cela n'est pas possible, des méthodes de résolution approchée telles que des algorithmes évolutionnaires peuvent être utilisées [EMMERICH et al. 2006]. Dans le cadre de l'optimisation



de propriétés moléculaires, il semble indispensable d'utiliser une méthode de résolution approchée. Nous montrerons dans la suite de ce chapitre que nous pouvons utiliser pour cela l'algorithme évolutionnaire que nous avons proposé au Chapitre 2.

## 4.3 Optimisation de graphes moléculaires basée sur un modèle de substitution

### 4.3.1 Méthode

Nous proposons d'utiliser une approche d'optimisation boîte-noire à l'aide d'un modèle de substitution pour le problème d'optimisation de propriétés moléculaires. Notre objectif est de permettre l'optimisation de propriétés moléculaires coûteuses, de manière plus efficace qu'un algorithme évolutionnaire. L'Algorithme 3 présenté en section 4.2 de ce chapitre est un algorithme très générique, dans lequel la recherche de solutions candidates par optimisation de la fonction de mérite correspond à un sous-problème cloisonné qui peut être traité de différentes façons en fonction du problème d'application. Pour l'optimisation de propriétés moléculaires, nous choisissons comme nous l'avons fait dans les chapitres précédents de considérer l'espace de recherche des graphes moléculaires. Cela nous permet d'utiliser pour ce problème une méthode d'optimisation métaheuristique, et plus précisément l'algorithme évolutionnaire que nous avons présenté au Chapitre 2. Notons que comme pour l'optimisation de propriétés moléculaires en général, il serait possible d'utiliser une méthode d'optimisation continue dans l'espace latent d'un auto-encodeur variationnel (voir la section 1.3.3 de ce mémoire et à titre d'exemple les travaux de [GÓMEZ-BOMBARELLI et al. 2018]). L'inconvénient de cette approche est qu'il n'existe pas de garantie que la totalité de l'espace moléculaire soit représentée au sein de cet espace de recherche. Il ne semble en revanche pas envisageable d'utiliser une méthode de génération basée sur un modèle d'apprentissage profond devant être entraîné à chaque étape de l'Algorithme 3, puisque cet entraînement représenterait un coût de calcul très important.

**Recherche de solutions candidates** Le fonctionnement général de notre méthode est décrit en Algorithme 3 (section 4.2 de ce chapitre). À chaque étape, le modèle de substitution est entraîné sur l'ensemble des données connues. Il est utilisé pour obtenir un ensemble de solutions candidates, qui seront évaluées par  $f$  et insérées dans le jeu de

**Algorithme 4** Recherche de solutions candidates à l'aide d'un algorithme évolutionnaire

**entrée:**  $S$  l'espace de recherche de l'optim. évol. (atomes, taille max. des molécules),  
 $X$  l'ensemble des solutions connues  
 $m_s$  la fonction de mérite dépendant du modèle de substitution  $s$ ,  
 $l$  la taille de la population initiale des optimisations évolutionnaires,  
 $n$  le nombre de solutions candidates générées

$C \leftarrow \emptyset$  ▷ Ensemble des solutions candidates générées

**pour**  $i$  de 1 à  $n$  **faire** ▷  $n$  redémarrages, pour générer  $n$  solutions

sélection aléatoire d'un ens. de solutions  $D \subseteq X$  de taille  $l$ , selon les valeurs de  $f$

$pop \leftarrow$  maximisation de  $m_s$  dans  $S$  par EvoMol avec  $D$  la pop. initiale et  $L_{\text{tabou}} = X$

$c \leftarrow \max_{x \in pop \setminus C} m_s(x)$  ▷  $c$  la meilleure solution de  $pop$  absente de  $C$

$C \leftarrow C \cup \{c\}$

**fin pour**

**retourner**  $C$

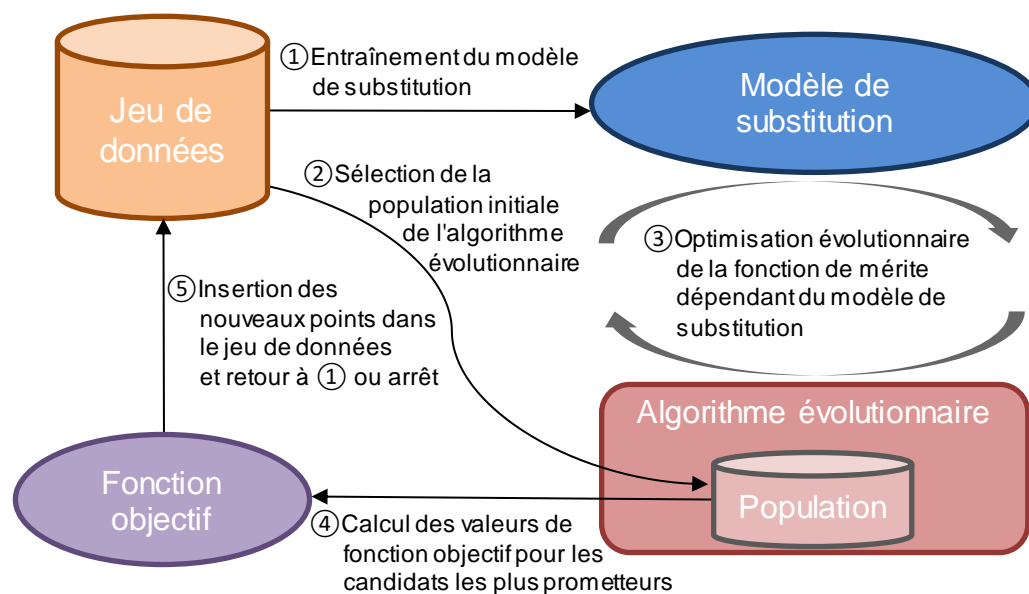


FIGURE 4.1 – Représentation schématique du fonctionnement de notre méthode.

données d'entraînement lors de l'étape suivante. La façon dont les candidats sont générés est décrite dans l'Algorithme 4. Nous choisissons pour cela d'utiliser EvoMol, l'algorithme évolutionnaire que nous avons conçu et décrit dans le Chapitre 2. EvoMol nous permet d'optimiser la fonction de mérite dans l'espace des graphes moléculaires. Ainsi, notre méthode n'est pas restreinte à la sélection des candidats les plus prometteurs depuis un jeu de données prédéfini, mais peut générer des solutions dans l'espace des graphes moléculaires, tout en profitant du gain d'efficacité potentiel induit par l'utilisation du modèle de substitution. La Figure 4.1 représente le fonctionnement de notre méthode de façon schématique. Les Algorithmes 3 et 4 y sont représentés dans un schéma unique.

Pour tirer partie au maximum du modèle de substitution, nous choisissons de générer un lot de  $n$  candidats à chaque étape d'optimisation, qui seront évalués par la fonction objectif et insérés dans le jeu de données d'entraînement simultanément. Pour cela, nous choisissons d'effectuer  $n$  redémarrages (« *restarts* ») de la procédure d'optimisation évolutionnaire. À chaque redémarrage, la population initiale est sélectionnée parmi l'ensemble des solutions connues. Cette sélection est effectuée de façon indépendante pour chaque redémarrage, dans l'objectif de favoriser l'obtention de solutions diverses. La stratégie de sélection que nous utilisons consiste à sélectionner aléatoirement  $l$  solutions du jeu de données, avec une probabilité de sélection proportionnelle à leur évaluation par  $f$ . Nous souhaitons ainsi favoriser la sélection des meilleures solutions connues, tout en laissant la possibilité que toute solution puisse faire partie du jeu de données de départ.

Puisque les appels à  $f$  à l'issue de la procédure de génération de candidats sont coûteux, nous prenons garde à concevoir cette procédure de manière à éviter les appels inutiles. Ainsi, une liste taboue  $L_{\text{tabou}}$  correspondant à l'ensemble des points connus ( $X$ ) est donnée en paramètre à l'optimisation évolutionnaire, afin de garantir que les candidats proposés ne sont pas déjà connus. De même, les candidats déjà sélectionnés à l'étape courante sont supprimés de la population finale des optimisations évolutionnaires suivantes, afin de garantir l'unicité des candidats proposés. Finalement, notons que notre algorithme est facilement parallélisable puisque les  $n$  optimisations de la fonction de mérite sont indépendantes et peuvent donc être effectuées en même temps. De même, l'évaluation par la fonction objectif des  $n$  candidats peut être effectuée en parallèle.

**Sélection du jeu de données initial** Lors de l'initialisation de la procédure d'optimisation boîte-noire à l'aide d'un modèle de substitution, le jeu de données initial doit être sélectionné. Ce dernier servira à l'entraînement du modèle de substitution lors de la

première étape d’optimisation. Nous avons évoqué dans la section précédente différentes approches pour cette sélection, qui consistent en fait à définir un sous-problème d’optimisation combinatoire. Il n’existe pas de façon triviale d’adapter ces approches à l’espace des graphes moléculaires. Les méthodes par construction sont très inadaptées puisqu’elles dépendent explicitement d’un espace de recherche dans  $\mathbb{R}^n$  ou  $\mathbb{Z}^n$ . Il serait envisageable d’adapter les approches statistiques ou par mesure de distance. Cependant, cela impliquerait de devoir résoudre un problème d’optimisation dont la fonction objectif évaluerait non pas une molécule, mais l’ensemble du jeu de données. Nous pourrions imaginer une mesure de contribution à l’objectif total permettant d’évaluer les graphes moléculaires, sur le modèle de la contribution à la diversité totale que nous avons définie au Chapitre 3. Cependant, il s’agit d’un sujet de recherche à part entière, que nous choisissons d’éluder dans le cadre de ce mémoire. En effet, il n’est pas indispensable d’y répondre de manière aboutie pour l’étude que nous proposons dans la suite de ce chapitre.

Pour la sélection des jeux de données initiaux au sein de ce chapitre, nous utiliserons deux approches relativement simples. La première consiste à utiliser comme jeu de données initial la molécule de méthane uniquement, comme nous l’avons fait dans les chapitres précédents. Cela nous permettra d’étudier le comportement de notre méthode d’optimisation lorsque la connaissance initiale est minimale. La seconde consiste à tirer un sous-ensemble aléatoire d’un jeu de données de référence. Cela nous permettra de comparer l’efficacité de notre méthode en fonction de la pertinence du jeu de données pour le problème d’optimisation considéré.

### 4.3.2 Apprentissage du modèle de substitution

Nous faisons le choix d’utiliser des modèles de régression par processus gaussien (GPR) en tant que modèles de substitution, en association avec les fonctions de mérite probabilistes que nous avons présentées dans la section précédente. Nous justifions ce choix par le fait qu’il s’agit d’une méthode de référence dans l’état de l’art, qui à notre connaissance n’a pas été utilisée antérieurement pour l’optimisation de propriétés moléculaires. Leur étude semble donc pertinente et importante méthodologiquement. Les GPR seront étudiés en utilisant deux types de noyaux, présentés dans la section 1.2 de ce mémoire. Il s’agit des noyaux  $k_{\text{RBF}}$  et  $k_{\text{DOTPRODUCT}}$ . Dans certains cas, nous utiliserons également le noyau  $k_{\text{WHITE}}$  permettant de représenter dynamiquement le bruit des données.

Puisque nous utilisons des modèles d’apprentissage artificiel pour la prédiction de propriétés moléculaires, il est nécessaire d’utiliser un descripteur moléculaire pour représenter

les molécules en entrée de ces modèles. Les descripteurs permettent notamment de mettre en évidence des caractéristiques des molécules pertinentes pour le problème d'apprentissage, et permettent également d'obtenir différentes invariances (à l'ordre des atomes, etc.). Nous avons dédié la section 1.1.3 de ce mémoire à la présentation de plusieurs descripteurs moléculaires communément utilisés. Dans le cadre de ce chapitre, nous en étudierons deux.

Le premier descripteur, le vecteur entier de *shingles*, est un descripteur basé exclusivement sur le graphe moléculaire. Il représente dans un vecteur de nombres entiers le nombre d'occurrences des sous-graphes moléculaires d'un rayon donné. Ainsi, la  $i^{\text{ème}}$  valeur correspond au nombre de fois qu'apparaît le graphe moléculaire identifié par  $i$  dans la molécule. Nous limiterons ici le rayon des graphes moléculaires à 1, afin de limiter la combinatoire et d'utiliser un descripteur de taille raisonnable. Un vecteur de taille 2000 sera suffisant pour l'ensemble des expériences que nous effectuerons dans ce chapitre.

Le second descripteur, MBTR, est un descripteur basé sur la géométrie moléculaire qui représente dans un vecteur de nombres réels le type des atomes, les distances entre les couples d'atomes et les angles entre les triplets d'atomes. En tant que descripteur basé sur la géométrie, il dépend d'une procédure pour obtenir cette géométrie. Nous utiliserons pour cela la mécanique moléculaire, qui possède un coût relativement faible. Rappelons cependant que ce coût dépend de la taille des molécules, et peut devenir trop élevé pour une utilisation à grande échelle sur des molécules de grandes tailles. Nous définirons un paramétrage de ce descripteur de sorte qu'il possède une taille comparable au vecteur de *shingles*.

### 4.3.3 Implémentation

Comme pour notre algorithme évolutionnaire EvoMol, nous mettons à disposition sur GitHub une implémentation en langage Python de notre algorithme<sup>1</sup>. Notre programme est conçu pour pouvoir être utilisé par des personnes ayant une connaissance minimale de la programmation. Une interface sous forme de dictionnaire permet de spécifier l'ensemble des paramètres. Cela inclut la sélection de la fonction objectif, qui peut être une fonction définie par l'utilisateur, mais également les paramètres du modèle de substitution et de la fonction de mérite. Cette interface est documentée sur la page GitHub. À titre d'exemple, nous reportons ci-dessous le code permettant de reproduire l'expérience de maximisa-

---

1. <https://github.com/jules-leguy/BBOMol>

tion de l'énergie HOMO qui sera présentée en section 4.5.2, en utilisant le descripteur MBTR et le noyau  $k_{\text{RBF}}$ . Cet extrait de code dépend de la bibliothèque Scikit-Learn pour l'implémentation du modèle de substitution GPR [PEDREGOSA et al. 2011].

```
1 from sklearn.gaussian_process import GaussianProcessRegressor
2 from sklearn.gaussian_process.kernels import RBF
3 from bbomol import run_optimization
4
5 run_optimization({
6     "obj_function": "homo",          # Fonction objectif
7     "io_parameters":{
8         "smiles_list_init": ["C"], # Jeu de donnees initial (methane)
9     },
10    "merit_optim_parameters": {
11        "evomol_parameters": {
12            "optimization_parameters": {
13                "max_steps": 10,    # Nb. etapes optim. EvoMol
14            },
15            "action_space_parameters": {
16                "max_heavy_atoms": 9, # Taille max. des molecules
17                "atoms": "C,N,O,F"   # Types des atomes lourds
18            }
19        },
20        "merit_EI_xi": 0.01,        # Valeur du param. d'exploration xi
21        "n_merit_optim_restarts": 10, # Nb. sol. generees (param. n)
22    },
23    "bbo_optim_parameters": {
24        "max_obj_calls": 1000      # Critere arret (appels a la f. obj.)
25    },
26    "surrogate_parameters": {
27        "GPR_instance": GaussianProcessRegressor(1.0*RBF(1.0),
28        normalize_y=False, alpha=1e-1), # Param. du modele subst.
29        "descriptor": {
30            "type": "MBTR"         # Descripteur moleculaire
31        }
32    })
```

### 4.3.4 Travaux liés

Si l’on omet les métaheuristiques qui sont peu efficaces en termes de nombre d’appels à la fonction objectif, relativement peu de travaux proposent d’utiliser des méthodes d’optimisation boîte-noire pour la découverte de solutions en chimie. La revue [TERAYAMA et al. 2021] référence plusieurs méthodes d’optimisation à l’aide d’un modèle de substitution appliquées à la chimie. La nette majorité de ces travaux est dédiée à la chimie du solide, dans laquelle les objets d’étude ne sont pas des molécules. Les caractéristiques des problèmes d’optimisation pour cette chimie sont souvent très différentes, et l’espace de recherche peut parfois être décrit dans  $\mathbb{R}^d$ . On peut par exemple citer le travail de [HOMMA et al. 2020], qui utilisent un modèle d’apprentissage au sein d’une procédure d’optimisation pour trouver un mélange de composés qui maximise une propriété électronique. Plus précisément, le problème consiste à déterminer la proportion relative optimale d’un ensemble prédéfini de composés.

Nous pouvons également mentionner COMBO, une bibliothèque conçue pour l’optimisation bayésienne efficace en chimie du solide [UENO et al. 2016]. COMBO est notamment basée sur une fonction de mérite probabiliste dont le calcul est très efficace, et utilise une approximation rapide du noyau  $k_{\text{RBF}}$ . La bibliothèque est utilisée pour sélectionner de manière efficace les solutions les plus prometteuses d’un jeu de données prédéfini, mais ne propose pas de manière de générer des solutions dans l’ensemble de l’espace de recherche.

Nous pouvons également citer le travail de [TERAYAMA et al. 2020], qui est également basé sur un modèle de substitution et dont l’objectif est de rechercher des solutions dont les propriétés cibles sont éloignées de la distribution connue dans le jeu de données. La fonction de mérite évalue à quel point la valeur de propriété prédite est éloignée de cette distribution. Elle présente l’avantage de pouvoir être appliquée simultanément pour plusieurs propriétés, et d’être indépendante d’un type de modèle de substitution en particulier. Cependant, cette approche ne permet pas de spécifier des valeurs cibles de propriétés, et ne propose pas non plus de manière de générer des solutions.

Finalement, le travail qui se rapproche le plus du nôtre est un algorithme nommé MOLGENGO [KANG et al. 2021]. Ce dernier combine un algorithme d’optimisation métaheuristique CSA (voir la section 1.3 de ce mémoire) avec une fonction de substitution basée sur un modèle de type arbre de décision. Un arbre de décision est un arbre binaire encodant une route de décisions selon les valeurs des caractéristiques. La représentation utilisée est la représentation sous forme de vecteur d’entiers tirée de la grammaire sans contexte proposée par [YOSHIKAWA et al. 2018] dans le cadre de l’algorithme ChemGE

(voir la section 1.3.2). MOLGENGO permet donc d'optimiser une propriété coûteuse dans l'espace des graphes moléculaires, qui est équivalent à l'espace des SMILES. Les auteurs effectuent une preuve de concept pour l'optimisation conjointe de deux propriétés électroniques. Une différence importante avec notre méthode est que les solutions candidates ne sont pas évaluées par la fonction objectif avant la fin de la recherche, et ne sont donc pas intégrées au jeu de données d'entraînement du modèle de substitution. Ce dernier est en fait entraîné uniquement à l'initialisation de la recherche. Précisons finalement que MOLGENGO et BBOMol ont été proposés de façon quasi-simultanée<sup>2</sup>.

## 4.4 Étude de l'optimisation d'une propriété peu coûteuse

Dans cette section, nous proposons de mener une étude générale de notre approche, dans l'objectif d'étudier ses performances d'optimisation, et l'effet des principaux paramètres. Notre méthode est conçue pour optimiser de façon efficace des propriétés moléculaires coûteuses, telles que les propriétés électroniques en chimie des matériaux moléculaires organiques. Cela implique cependant qu'une étude étendue de notre approche sur ce genre de propriété serait très coûteuse. Nous choisissons donc dans un premier temps de mener cette étude sur une propriété peu coûteuse que nous avons déjà étudiée dans les chapitres précédents, à savoir la QED. Dans la première partie de cette section, nous étudions la capacité du modèle de substitution à prédire les valeurs de QED, en dehors de toute procédure d'optimisation. Dans la seconde partie, nous intégrons les modèles de substitution au sein de notre procédure d'optimisation boîte-noire, et nous étudions l'efficacité de la recherche en fonction des paramètres d'optimisation et par rapport à la performance de l'algorithme évolutionnaire EvoMol que nous avons proposé dans le Chapitre 2.

### 4.4.1 Apprentissage et évaluation du modèle de substitution

Nous cherchons d'abord à évaluer la capacité des modèles que nous utiliserons ultérieurement en tant que modèles de substitution à prédire les valeurs de QED. Pour cela, nous consacrons cette section à l'étude de la prédiction des valeurs de QED à l'aide de

---

2. L'article présentant BBOMol a été soumis pour publication le 20 juillet 2021, tandis que l'article présentant MOLGENGO a été soumis le 13 août 2021.



modèles GPR selon différents paramétrages et différentes conditions expérimentales.

**Jeux de données et prétraitements** Les deux jeux de données que nous choisissons d’utiliser pour cette étude sont QM9 [RAMAKRISHNAN et al. 2014] et ChEMBL [GAULTON et al. 2017]. Ils sont présentés en détail dans la section 1.1.4 de ce mémoire. Pour rappel, QM9 correspond à une énumération partielle de l’espace moléculaire contenant jusqu’à 9 atomes lourds parmi {C, N, O, F}. Il s’agit d’un jeu de données synthétique potentiellement utile pour l’ensemble de la chimie organique. ChEMBL est un jeu de données orienté pour le domaine de la chimie pharmaceutique, qui contient notamment des composés naturels et des médicaments, de plus grandes tailles que QM9 (28 atomes lourds en moyenne). ChEMBL contient également une plus grande variété d’atomes lourds (comme par exemple les éléments P, S, Cl ou Br).

Afin de travailler sur des jeux de données homogènes et correspondant à l’espace chimique que l’on peut explorer à l’aide de notre algorithme évolutionnaire, et donc également de l’approche d’optimisation que l’on propose dans ce chapitre, nous effectuons un prétraitement de ChEMBL et QM9. Nous supprimons ainsi les radicaux et les molécules contenant une ou plusieurs charges (ions et zwitterions). Pour rappel, les opérateurs de mutation d’EvoMol ne permettent pas la création d’atomes chargés. Il est possible de traiter des populations de solutions contenant des atomes chargés, mais nous choisissons par simplicité de ne pas les considérer ici.

Pour que les données issues de QM9 et de ChEMBL soient plus comparables, nous choisissons de retirer de ChEMBL les molécules contenant des atomes lourds n’appartenant pas à l’ensemble {C, N, O, F}. À l’issue de ce prétraitement, notre sous-ensemble de QM9 contient 132 040 molécules (contre 133 885 initialement) et notre sous-ensemble de ChEMBL contient 867 606 molécules (contre 1 817 766 initialement). Dans la suite de ce chapitre, nous faisons référence à ces données lorsque nous mentionnons QM9 ou ChEMBL.

Nous étudions maintenant la distribution des valeurs de QED pour chacun de ces deux jeux de données. Ces distributions sont représentées en Figure 4.2. On y observe que les valeurs de QED sont très étendues dans ChEMBL, tandis qu’elles sont majoritairement comprises entre 0.2 et 0.6 dans QM9. Notons que la valeur maximale observée dans ChEMBL (0.948) correspond à la valeur optimale obtenue par de nombreuses méthodes d’optimisation de la QED (ces résultats sont présentés dans la section 2.3 de ce mémoire).

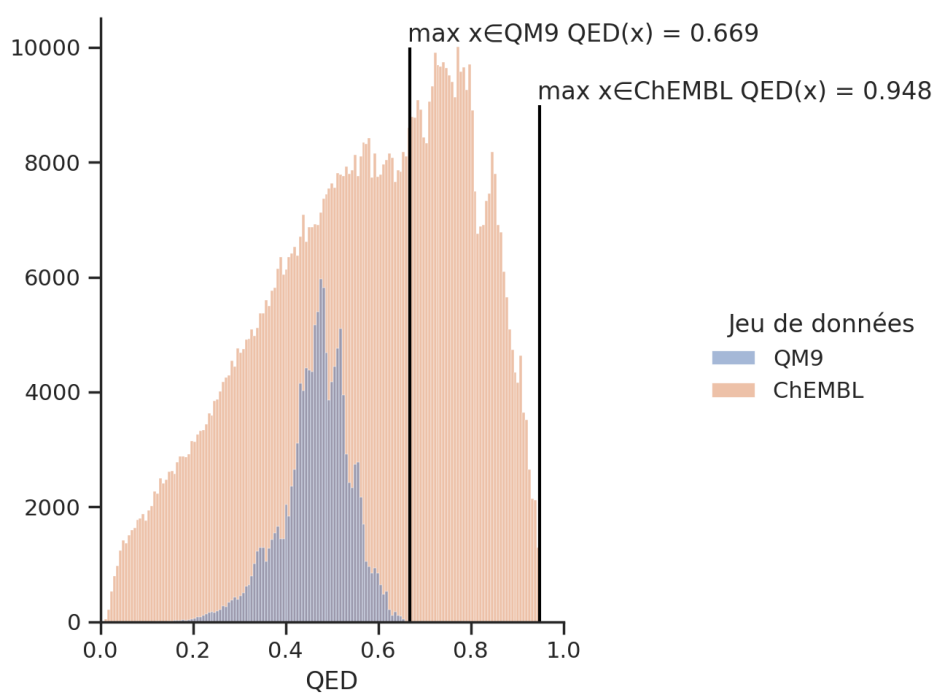


FIGURE 4.2 – Distribution des valeurs de QED en fonction du jeu de données. Les deux lignes verticales indiquent la valeur maximale observée dans chacun des jeux de données.

**Descripteur moléculaire** Nous choisissons d’utiliser pour cette étude le vecteur entier des occurrences de *shingles* de rayon 1 en tant que descripteur moléculaire. En tant que descripteur dépendant uniquement du graphe et non de la géométrie moléculaire, le vecteur de *shingles* peut être calculé relativement rapidement. Nous estimons qu’il faut en moyenne 0.27 secondes pour transformer 100 molécules de QM9, et 0.91 secondes pour transformer 100 molécules de ChEMBL<sup>3</sup>. Bien que non négligeable lorsque la fonction prédite est peu coûteuse, il s’agit d’un coût raisonnable dans le cadre de notre algorithme d’optimisation. Nous considérons en revanche qu’il serait trop coûteux d’utiliser un descripteur moléculaire basé sur la géométrie tel que MBTR dans ce contexte. Nous avons estimé dans la section 1.1.2 de ce mémoire qu’il faut au minimum  $10^{-2}$  à  $10^{-1}$  secondes pour calculer la géométrie d’une molécule de ChEMBL, ce qui implique qu’il faudrait jusqu’à  $10^1$  secondes pour calculer les descripteurs nécessaires à l’évaluation par le modèle de substitution de 100 solutions de l’espace de recherche.

**Fonction noyau et gestion du bruit** Nous étudions le modèle GPR dans le cadre où il est associé avec une fonction noyau  $k_{\text{RBF}}$  et  $k_{\text{DOTPRODUCT}}$ . Nous utilisons également un noyau  $k_{\text{WHITE}}$  afin que le bruit des données soit estimé dynamiquement. Dans la suite de cette étude pour la prédiction et l’optimisation des valeurs de QED, nous étudions donc les deux noyaux  $k_{\text{RBF}}(x, x') + k_{\text{WHITE}}(x, x')$  ainsi que  $k_{\text{DOTPRODUCT}}(x, x') + k_{\text{WHITE}}(x, x')$ . Pour simplifier la notation, nous y ferons référence simplement par  $k_{\text{RBF}}$  et par  $k_{\text{DOTPRODUCT}}$ .

**Implémentation** Nous utilisons l’implémentation des modèles GPR de la bibliothèque Scikit-Learn [PEDREGOSA et al. 2011]. Nous activons le paramètre de normalisation dynamique des valeurs cibles (les valeurs de QED) en entrée du modèle, afin qu’elles aient une moyenne nulle et un écart type unitaire. Cette transformation est effectuée selon la moyenne et l’écart-type observés sur le jeu de données d’entraînement, et la réciproque de cette transformation est appliquée dynamiquement sur les valeurs prédites.

## Étude des performances de prédiction

Nous menons d’abord une étude des performances de prédiction des valeurs de QED en fonction de la taille des données d’entraînement. Nous cherchons également à mesurer le temps d’entraînement et de prédiction des modèles. Ces informations nous permettront

---

3. Pour produire cette estimation, nous mesurons le temps de calcul du descripteur pour 50 000 molécules tirées aléatoirement de QM9 et ChEMBL.

de régler plus facilement les paramètres de notre algorithme d'optimisation dépendant de ces modèles par la suite. Rappelons que les GPR ne nécessitent à proprement parler pas de phase d'entraînement dans laquelle ils construiraient un modèle de la fonction prédite. Chaque prédiction dépend en effet d'un calcul impliquant l'ensemble des points connus. Cependant, une partie de ce calcul est indépendante des points dont la valeur doit être prédite, et peut donc être effectuée au préalable. Ce que nous appelons entraînement correspond donc plus exactement au calcul du cache de la matrice  $K(X, X)$  (voir la section 1.2 de ce mémoire). Cela comprend également l'optimisation des hyper-paramètres de la fonction noyau, qui comprend ici l'estimation du bruit gaussien dans les données d'apprentissage.

L'expérience que nous menons consiste à tester notre modèle en fonction de la taille du jeu d'entraînement. Nous effectuons d'abord une séparation aléatoire de chacun des jeux de sorte à obtenir un jeu de test contenant 50 000 molécules. Le reste des molécules forme le jeu complet d'entraînement  $E$ . Nous entraînons alors les deux modèles d'apprentissage utilisant chacune des fonctions noyau en utilisant comme jeu d'entraînement un sous-ensemble de  $E$  de taille variant parmi  $\{10, 100, 1000, 10000\}$ . Ce modèle est alors évalué sur le jeu de test. Cette procédure est effectuée à la fois sur ChEMBL et QM9.

La Figure 4.3 représente les erreurs observées pour les différents modèles pour la prédiction des valeurs de QED, et plus exactement la mesure d'erreur moyenne absolue, communément abrégée par « MAE » en anglais (pour *Mean Absolute Error*). On observe que dans toutes les conditions, l'erreur est plus faible sur QM9 que sur ChEMBL. Cela peut s'expliquer par le fait que les données sont plus homogènes (notamment en taille des molécules), et par le fait que les valeurs de QED sont également moins étendues (voir la Figure 4.2). On observe de plus que le noyau  $k_{\text{RBF}}$  est le plus efficace pour les deux jeux de données. Finalement, il semble que l'erreur diminue fortement lorsque le jeu de données augmente jusqu'à contenir environ 3000 molécules, puis diminue plus lentement voire semble stagner lorsque  $k_{\text{DOTPRODUCT}}$  est utilisé.

Nous étudions maintenant la Figure 4.4, qui représente les temps mesurés en fonction de la taille des données d'entraînement. Nous observons d'abord que le noyau  $k_{\text{DOTPRODUCT}}$  est systématiquement plus rapide que  $k_{\text{RBF}}$ . Il s'agit d'un résultat attendu puisque ce dernier dépend du calcul d'une exponentielle, plus coûteux qu'un produit scalaire. Nous observons également (partie gauche de la figure) que le temps nécessaire à l'entraînement du GPR devient rapidement très important. L'entraînement nécessite en effet plusieurs centaines à plus d'un millier de secondes lorsque la taille du jeu de données d'entraînement

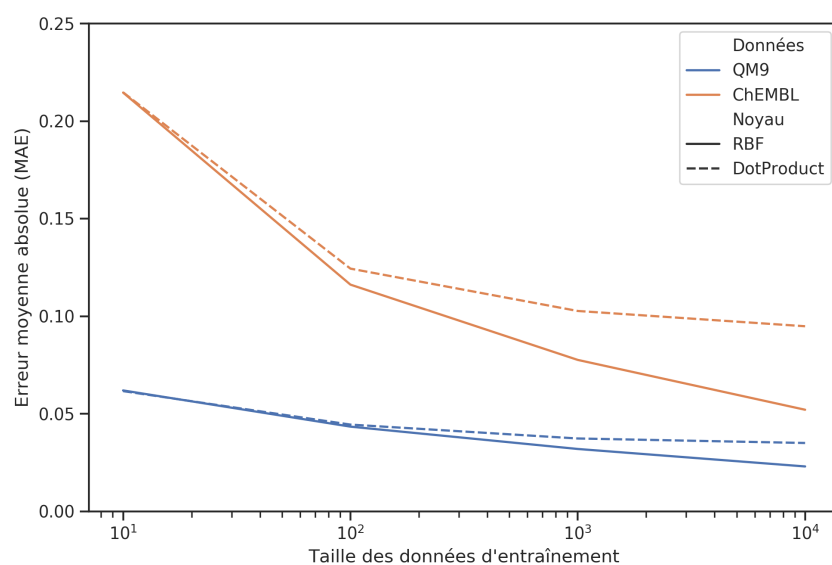


FIGURE 4.3 – Erreur moyenne absolue pour la prédiction de la valeur de QED à l'aide du modèle GPR en fonction du jeu de données (QM9 ou ChEMBL), de la fonction noyau ( $k_{\text{RBF}}$  ou  $k_{\text{DOTPRODUCT}}$ , notée resp. RBF et DotProduct), et de la taille du jeu de données d'entraînement. Tous les points sont obtenus par test sur un sous-ensemble fixe du jeu de données contenant 50 000 molécules. Les données d'entraînement sont extraites du sous-ensemble complémentaire du jeu de données.

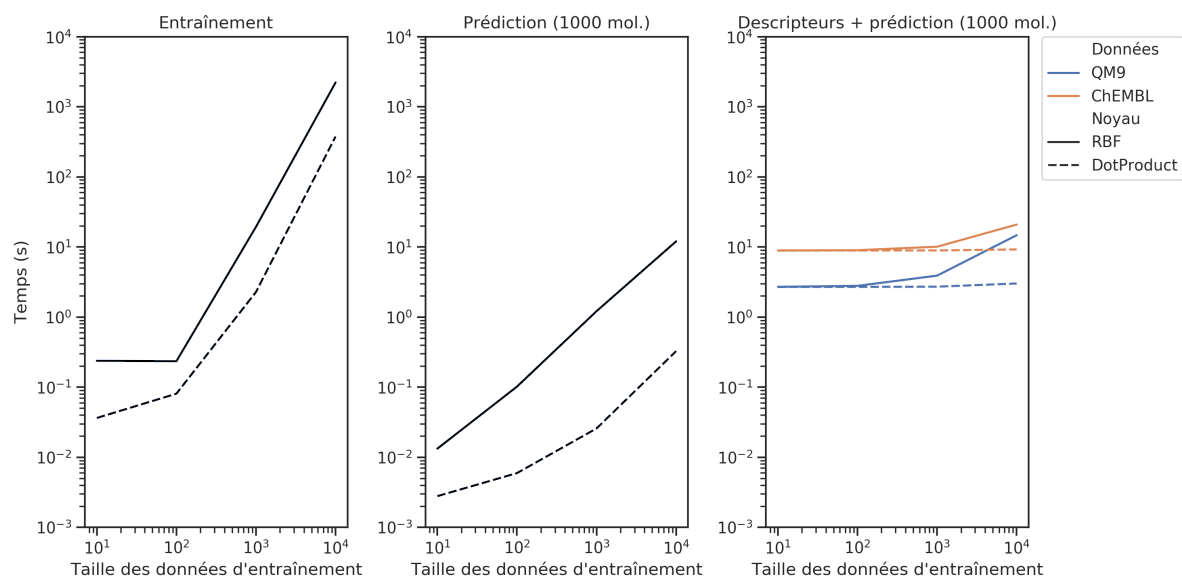


FIGURE 4.4 – À gauche : temps pour l'entraînement du modèle de prédiction de la QED en fonction de la fonction noyau et de la taille du jeu de données d'entraînement. Au centre : temps pour la prédiction de la valeur de QED de 1000 molécules en fonction de la fonction noyau et de la taille du jeu de données d'entraînement. À droite : temps pour le calcul des descripteurs et la prédiction de la valeur de QED pour 1000 molécules en fonction du jeu de données, de la fonction noyau et de la taille du jeu de données d'entraînement. QM9 et ChEMBL correspondent aux deux jeux de données d'entraînement et de test, RBF et DotProduct correspondent aux deux fonctions noyau  $k_{\text{RBF}}$  et  $k_{\text{DOTPRODUCT}}$  respectivement. Les valeurs sont représentées en noir à gauche et au centre car le temps pour l'entraînement et la prédiction du modèle est indépendant du jeu de données, contrairement au calcul des descripteurs.

Fonction noyau	MAE inter-modèles			
	Moyenne		Écart-type	
	QM9	ChEMBL	QM9	ChEMBL
$k_{\text{RBF}}$	0.031	0.078	0.000	0.001
$k_{\text{DOTPRODUCT}}$	0.037	0.103	0.000	0.001

TABLE 4.1 – Moyenne et écart-type des valeurs de MAE pour la prédiction des valeurs de QED, mesurées sur les différents plis en fonction du jeu de données et de la fonction noyau.

atteint 10 000. Il s’agit d’une limitation non négligeable mais connue des GPR. Elle est accentuée ici par le fait que l’on utilise un descripteur de grande dimension (2000) pour un modèle de ce type.

Le temps nécessaire pour la prédiction des valeurs de QED (partie centrale de la Figure 4.4) est plus faible, mais devient non négligeable lorsque la taille des données d’entraînement augmente. Il faut en effet environ 10 secondes pour prédire les valeurs de 1000 molécules lorsque le jeu de données d’entraînement en contient 10 000. Lorsque l’on prend également en compte le temps nécessaire au calcul des descripteurs (partie droite de la figure), la transformation du graphe moléculaire en vecteur de *shingles* implique un coût de calcul constant non négligeable, en particulier pour les faibles tailles de jeu de données d’entraînement. Il faut ainsi 2 à 20 secondes pour évaluer les valeurs de QED de 1000 solutions de l’espace de recherche, en considérant le calcul des descripteurs et un jeu de données d’entraînement de taille comprise entre 10 et 10 000. Nous concluons de ces expériences que  $10^3$  semble un ordre de grandeur raisonnable pour la taille du jeu de données d’entraînement. Cela correspond à un compromis raisonnable entre coût de calcul et qualité des prédictions. Il s’agit par conséquent de l’ordre de grandeur des tailles de jeux de données que nous utiliserons dans les expériences d’optimisation basée sur le modèle de substitution.

Nous cherchons désormais à obtenir une estimation plus précise des erreurs de nos modèles, dans un contexte comparable à l’utilisation que nous en ferons en tant que modèles de substitution dans notre approche d’optimisation. Nous menons pour cela une nouvelle expérience. Pour chacun des jeux de données, nous extrayons de façon aléatoire (sans remise) 100 plis contenant chacun 1000 molécules. Pour chacun des jeux de données et pour chacune des deux fonctions noyau, nous effectuons une validation croisée en utilisant itérativement chaque pli comme jeu de données d’entraînement. Chaque modèle est évalué sur l’ensemble des plis n’ayant pas servi à son entraînement. Cette configuration

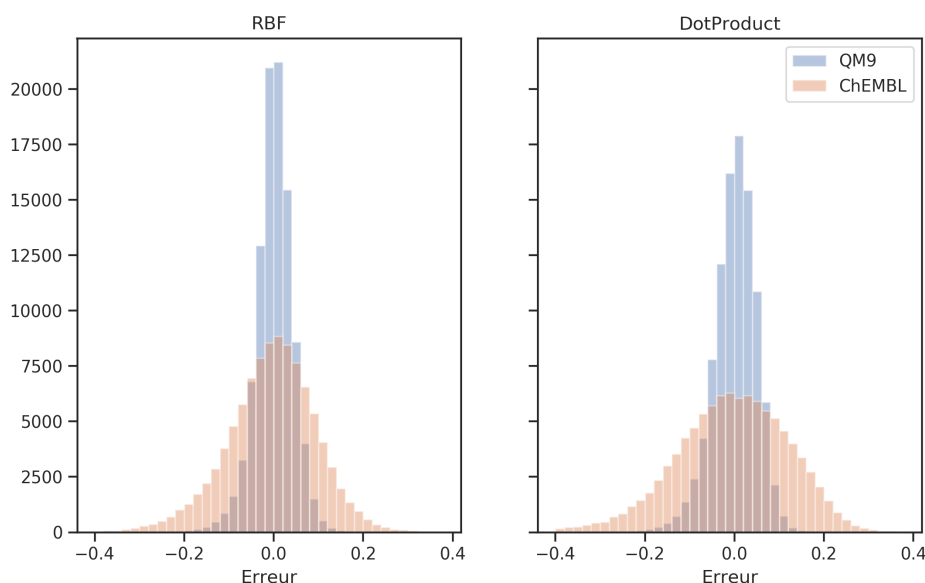


FIGURE 4.5 – Distribution des erreurs du modèle entraîné sur le premier pli pour la prédiction des valeurs de QED en fonction du jeu de données et de la fonction noyau.

nous permet d'étudier le comportement de nos modèles lorsque le jeu de données d'entraînement comprend 1000 molécules, ce qui correspond à l'ordre de grandeur de la taille des données que nous utiliserons dans les expériences que nous mènerons dans la suite de cette section pour l'optimisation des valeurs de QED.

En Table 4.1, nous présentons plusieurs statistiques de la MAE obtenue selon cette méthodologie sur les différents plis pour chaque modèle étudié. Comme pour les expériences précédentes, on observe que les erreurs sont plus faibles lorsque la fonction  $k_{\text{RBF}}$  est utilisée, et lorsque le modèle est testé sur QM9. Rappelons toutefois que ChEMBL contient une plus grande diversité de molécules et de valeurs de QED, ce qui rend le problème d'apprentissage plus complexe. Nous ne pouvons donc pas conclure que le modèle entraîné sur le jeu de données QM9 possède de meilleures capacités de généralisation sur l'espace moléculaire. L'étude de l'écart-type nous permet d'observer que les modèles sont extrêmement stables, quelle que soit la configuration.

L'écart-type des MAE étant très faible, nous savons que la valeur de MAE varie très peu en fonction du pli d'entraînement. Nous souhaitons dorénavant étudier les erreurs commises par un modèle sur son jeu de test. Pour cela, nous représentons en Figure 4.5 la distribution des erreurs du modèle entraîné sur le premier pli et testé sur l'ensemble



Fonction noyau	Erreur intra-modèle			
	Moyenne		Écart-type	
	QM9	ChEMBL	QM9	ChEMBL
$k_{\text{RBF}}$	0.001	-0.005	0.041	0.100
$k_{\text{DOTPRODUCT}}$	0.001	-0.006	0.047	0.129

TABLE 4.2 – Moyenne et écart-type des erreurs enregistrées pour le modèle entraîné sur le premier pli pour la prédiction des valeurs de QED.

des autres plis, en fonction du jeu de données et de la fonction noyau. Nous observons comme attendu que les erreurs sont plus resserrées au centre de la distribution pour les modèles évalués sur QM9 que pour les modèles évalués sur ChEMBL, et pour les modèles basés sur un noyau  $k_{\text{RBF}}$  plutôt qu’un noyau  $k_{\text{DOTPRODUCT}}$ . Ces distributions semblent approximativement gaussiennes. Nous observons dans les valeurs de la Table 4.2 qu’en supposant que ces distributions sont effectivement gaussiennes, elles sont paramétrées par une moyenne proche de 0, et un écart-type proche de 0.04 pour les modèles évalués sur QM9 et compris entre 0.10 et 0.13 pour les modèles évalués sur ChEMBL.

Les performances des modèles GPR que nous avons étudiés pour la prédiction de la QED nous semblent satisfaisantes. Ces modèles produisent des erreurs moyennes qui sont comprises entre 0.03 et 0.1, pour la prédiction d’une propriété qui est définie entre 0 et 1. Nous pouvons nous attendre à ce que ces modèles permettent d’identifier des solutions prometteuses pour la recherche de valeurs de QED élevées, et par conséquent il semble raisonnable de les utiliser en tant que modèles de substitution au sein de notre approche d’optimisation. Il faut cependant retenir que le temps d’entraînement et de la prédiction de points arbitraires de l’espace de recherche peut devenir important. Nous veillerons par conséquent à utiliser des jeux de données de taille raisonnable par rapport à ce facteur.

#### 4.4.2 Évaluation de notre méthode d’optimisation

Nous proposons maintenant d’étudier les performances d’optimisation de notre approche, en intégrant les modèles de substitution étudiés dans la section précédente au sein de notre procédure d’optimisation boîte-noire. Nous divisons cette étude en trois sous-parties. Dans un premier temps, nous effectuons une étude par énumération exhaustive d’une grille de plusieurs paramètres, à savoir le jeu de données initial, la fonction noyau du modèle de substitution et la fonction de mérite. Toutes ces expériences sont effectuées avec et sans filtrage par contrainte de l’espace de recherche. Dans un deuxième

Paramètre	Valeur
Jeu de données initial	Méthane, QM9, ChEMBL
Fonction de mérite	POI, EI, id
Fonction noyau	$k_{\text{RBF}}$ , $k_{\text{DOTPRODUCT}}$
Contrainte sur l'espace de recherche	Aucune, sillywalks

TABLE 4.3 – Grille de paramètres étudiés pour l'optimisation de la valeur de QED à l'aide de notre approche.

temps, nous étudions l'influence du paramètre d'exploration  $\xi$  sur un sous-ensemble de cette grille. Finalement, nous proposons une étude de l'influence de la connaissance apprise par le modèle de la substitution. Pour cela, nous étudions les performances d'optimisation lorsque le descripteur moléculaire est tiré aléatoirement pour chaque molécule, et donc que le modèle de substitution perd la capacité de prédire des valeurs pertinentes.

### Étude générale de notre approche

**Paramètres de l'étude** La grille de paramètres dont nous étudions toutes les combinaisons est présentée en Table 4.3. Le premier de ces paramètres est le jeu de données initial. Nous utilisons deux stratégies différentes. La première consiste à utiliser un jeu de données contenant la molécule de méthane uniquement. Il s'agit d'une stratégie que nous avons déjà utilisée pour EvoMol dans le Chapitre 2. La deuxième consiste à utiliser comme jeu de données un sous-ensemble de taille 1000 tiré aléatoirement d'un jeu de données de référence. Cette taille, relativement petite, est choisie pour que le modèle de substitution puisse être entraîné en temps raisonnable. Les jeux de données de référence sont les sous-ensembles de QM9 et ChEMBL étudiés dans la section précédente pour l'apprentissage des valeurs de QED. Nous appliquons cependant une transformation supplémentaire au jeu de données ChEMBL. Ce dernier contient des molécules possédant des valeurs de QED très élevées (voir la Figure 4.2). Certaines des molécules de ChEMBL possèdent même une valeur de QED de 0.948, correspondant à la meilleure valeur obtenue par de nombreux travaux d'optimisation (voir le Chapitre 2). Puisque notre objectif est ici d'utiliser le jeu de données comme un point de départ pour l'optimisation de la QED, nous choisissons de retirer les solutions en possédant déjà une valeur élevée. Nous fixons le seuil à 0.669, qui correspond à la plus haute valeur de QED dans notre sous-ensemble de QM9. Cela permettra de plus de rendre les résultats d'optimisation plus facilement comparables. Les solutions retirées correspondent donc à l'ensemble de molécules situées entre les deux lignes verticales en Figure 4.2.

Le deuxième paramètre est la fonction de mérite. Nous étudions les fonctions POI et EI, présentées plus tôt dans ce chapitre. Le paramètre d’exploration  $\xi$  est fixé pour ces deux fonctions à la valeur de 0.01. Nous étudions également la fonction identité (id), qui correspond en fait à l’optimisation des valeurs prédites par le modèle de substitution. Le troisième paramètre étudié est la fonction noyau du modèle de substitution. Nous étudions les mêmes fonctions que pour l’étude de l’apprentissage de la QED dans la section précédente, à savoir  $k_{\text{RBF}}$  et  $k_{\text{DOTPRODUCT}}$ . Comme pour les modèles étudiés précédemment, la fonction noyau contient également la fonction  $k_{\text{WHITE}}$  afin d’estimer dynamiquement le bruit des données d’apprentissage.

Nous souhaitons également étudier l’influence de la contrainte dite « sillywalks » permettant de filtrer l’espace de recherche accessible, telle que définie et étudiée dans le Chapitre 2 pour favoriser le réalisme des solutions. Pour rappel, il s’agit d’ignorer les solutions de l’espace de recherche possédant des caractéristiques ECFP4 inconnues dans ChEMBL, un jeu de données de référence. Toutes les expériences de l’étude que nous menons sont effectuées avec et sans l’application de cette contrainte sur l’espace de recherche. Notons que lorsqu’elle est appliquée pour l’espace de recherche, nous l’appliquons également sur notre sous-ensemble de QM9. Ainsi, le jeu de données initial ne peut contenir des solutions qui ne font pas partie de l’espace de recherche. Par définition de la contrainte, il n’est pas nécessaire de l’appliquer pour ChEMBL.

**Paramètres de la recherche évolutionnaire** Les autres paramètres de BBOMol sont définis à une valeur fixe pour l’ensemble des expériences. Pour faciliter les explications puisque notre algorithme EvoMol est utilisé au sein de notre algorithme BBOMol et qu’ils sont tous deux constitués d’une boucle d’étapes d’optimisation, nous parlerons de boucle externe d’optimisation pour BBOMol et de boucle interne pour EvoMol. Le nombre de solutions  $n$  générées à chaque étape d’optimisation externe est fixé à 10. Le paramètre  $l$  qui définit la taille de la population initiale des optimisations évolutionnaires est fixé à 10 également. L’espace de recherche de l’optimisation évolutionnaire est fixé de la façon suivante. Les solutions peuvent contenir jusqu’à 50 atomes lourds parmi {C, N, O, F}, afin de correspondre à l’espace moléculaire représenté dans les deux jeux de données utilisés pour la sélection du jeu de données initial. Nous choisissons d’effectuer 10 étapes d’optimisation interne à chaque appel d’EvoMol. Cela signifie que les solutions candidates sont distantes d’au plus 10 mutations de solutions du jeu de données  $X$ . Il s’agit d’une valeur relativement peu élevée, qui permet d’espérer obtenir des solutions candidates

relativement proches des solutions connues, dans des zones de l'espace de recherche dans lesquelles l'interpolation effectuée par le modèle de substitution est susceptible d'être pertinente. La paramètre *TailleLot* d'EvoMol est fixé à 10. Cela signifie qu'à chaque étape d'optimisation interne, 10 solutions sont générées. Puisque 10 solutions sont générées à chacune de ces 10 étapes, un total de 100 solutions est généré en plus des 10 solutions de la population initiale. La taille maximale *TailleMax* de la population est fixée à une valeur supérieure (1000). Notons que cela signifie qu'aucun remplacement d'un individu défini n'a lieu lors de l'optimisation interne.

**Comparaison aux performances d'un algorithme évolutionnaire** Afin d'obtenir un point de comparaison, une expérience correspondant à la maximisation de la QED à l'aide d'un algorithme évolutionnaire uniquement est également effectuée dans des conditions comparables. Nous utilisons pour cela notre algorithme EvoMol. La taille maximale de la population est fixée à 1000, le nombre d'individus remplacés par étape est fixé à 10 et l'espace de recherche est composé de molécules contenant jusqu'à 50 atomes lourds parmi {C, N, O, F}. Comme pour les expériences de BBOMol, plusieurs expériences sont effectuées, en utilisant ou non la contrainte sillywalks, et en faisant varier la stratégie de sélection du jeu de données initial utilisé ici en tant que population initiale.

**Conditions expérimentales** Toutes les expériences sont effectuées 10 fois, avec un budget de 2000 appels à la fonction objectif pour chaque exécution. Ce budget est choisi de sorte à obtenir un jeu de données final contenant au plus 3000 solutions, puisque nous utiliserons des jeux de données initiaux contenant 1 ou 1000 solutions. Cela correspond à une taille permettant d'utiliser un modèle de substitution GPR en temps raisonnable selon l'étude que nous avons effectuée dans la section précédente.

**Résultats : impact du jeu de données initial** Dans un premier temps, nous étudions les résultats de l'optimisation sans filtrage de l'espace de recherche, selon les différentes stratégies pour sélectionner la population initiale. Nous commençons par la stratégie consistant à utiliser la molécule de méthane, dont les résultats sont présentés en Figure 4.6 (à gauche). On peut y observer que notre approche associée à une fonction de mérite pertinente (nous commenterons ultérieurement la fonction de mérite identité) est nettement plus efficace que l'optimisation évolutionnaire. Les courbes représentant ces exécutions de BBOMol augmentent nettement plus vite vers des valeurs élevées de QED.

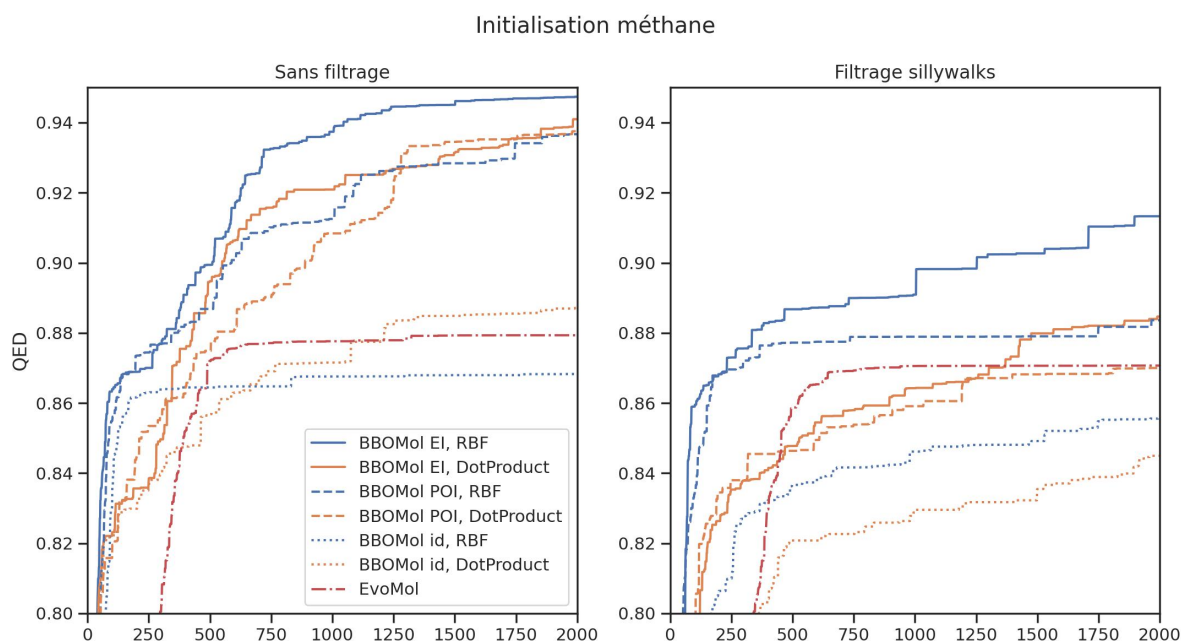


FIGURE 4.6 – Moyenne des meilleurs scores obtenus en fonction du nombre d’appels à la fonction objectif, pour différents paramétrages de BBOMol et pour EvoMol. La population initiale des expériences représentées contient la molécule de méthane uniquement. À gauche : résultats sans filtrage de l’espace de recherche. À droite : résultats avec filtrage selon la contrainte sillywalks. Les paramètres EI, POI et id représentent la fonction de mérite utilisée et sont différenciés par les types de lignes. Les paramètres RBF et DotProduct correspondent à la fonction noyau utilisée ( $k_{\text{RBF}}$  et  $k_{\text{DOTPRODUCT}}$ ) et sont différenciés par la couleur des lignes. L’expérience basée sur un algorithme évolutionnaire uniquement (EvoMol) est représentée par un type de ligne et une couleur uniques.

La Table 4.4 permet d'en obtenir une mesure quantitative. Elle contient les valeurs d'ERT pour l'ensemble des expériences et pour différentes valeurs cibles de QED. Pour rappel, la mesure d'ERT qui est définie en section 1.3.1 de ce mémoire représente l'espérance du coût de l'exécution en nombre d'appels à la fonction objectif pour obtenir une valeur cible de fonction objectif. Nous pouvons observer au sein de cette table que 18 583 appels à EvoMol sont nécessaires pour obtenir une seule fois une valeur de QED supérieure à 0.9, contre seulement 482 appels en moyenne pour le paramétrage le plus efficace de BBOMol.

Ces résultats peuvent être mis en relation avec la Table 3.2 du Chapitre 3, présentée en section 3.5. Les expériences d'optimisation des valeurs de QED à l'aide de notre algorithme évolutionnaire sont assez similaires, mais les résultats diffèrent de façon importante. Pour une cible de 0.948 avec une population initiale composée du méthane uniquement et sans contrainte sur l'espace de recherche, nous mesurons une ERT de 131 973 tandis que dans le cas présent nous mesurons une ERT de 19 625 seulement. L'expérience précédente était effectuée dans l'espace de recherche des graphes moléculaires composés d'au plus 38 atomes lourds parmi {C, N, O, F, P, S, Cl, Br} tandis que l'expérience courante est effectuée dans l'espace de recherche des graphes moléculaires composés d'au plus 50 atomes lourds parmi {C, N, O, F}. Nous ne pensons toutefois pas qu'il s'agit du facteur déterminant pour l'explication de la différence entre les résultats observés. Nous pensons qu'elle est expliquée principalement par la variation du critère d'arrêt. Il correspondait à l'application de 800 étapes d'optimisation dans les expériences précédentes, contre 2000 appels à la fonction objectif dans l'expérience courante. Or, dans notre algorithme ces deux quantités ne sont pas liées de façon linéaire et 800 étapes d'optimisation peuvent correspondre à un nombre d'appels à la fonction objectif très supérieur à 2000. Une étude manuelle des résultats avec le critère d'arrêt fixé à 800 étapes montre qu'il faut jusqu'à 400 000 appels à la fonction objectif pour obtenir une solution atteignant la cible de 0.948, mais que les 10 exécutions permettent d'obtenir une solution atteignant la cible. Pour l'expérience courante, seule 1 exécution permet d'obtenir cette cible, mais par définition de l'expérience le coût de chacune des 9 exécutions qui ne permettent pas d'obtenir la cible est seulement de 2000 appels. Cela met en évidence de façon expérimentale que le budget fixé pour l'optimisation doit être comparable pour que les valeurs d'ERT puissent être comparées.

Lorsque la population initiale est un sous-ensemble de QM9, on peut observer (à gauche de la Figure 4.7) que cela favorise nettement l'ensemble des exécutions. La connaissance contenue dans le jeu de données initial  $a$ , comme on pourrait l'attendre, une influence

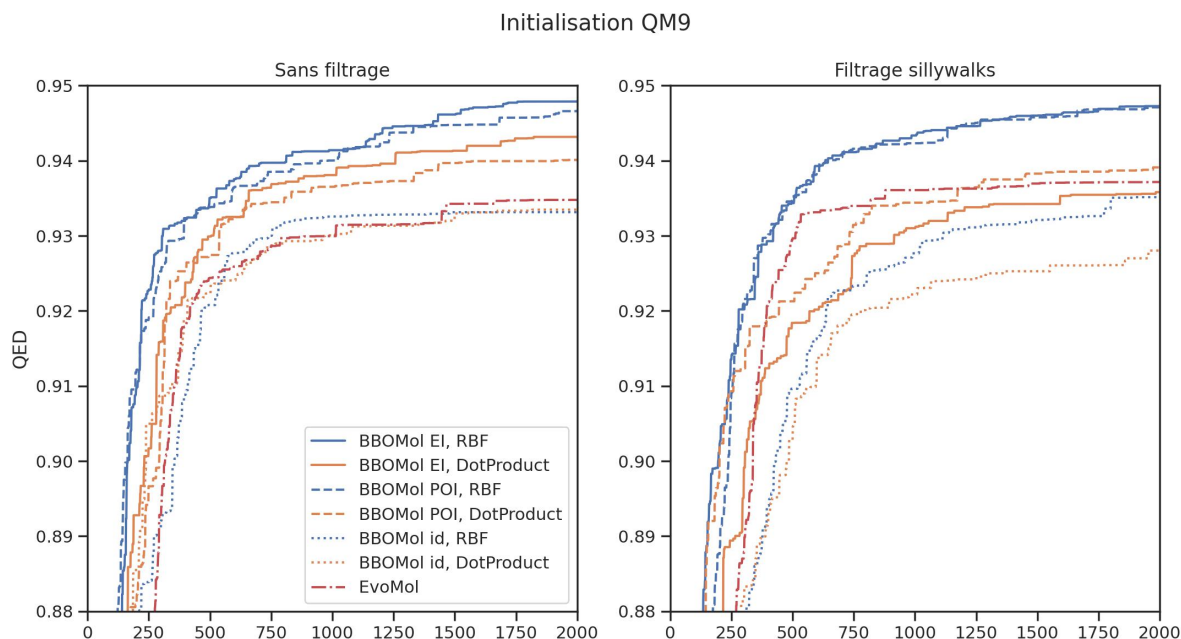


FIGURE 4.7 – Moyenne des meilleurs scores obtenus en fonction du nombre d’appels à la fonction objectif, pour différents paramétrages de BBOMol et pour EvoMol. La population initiale des expériences représentées est un sous-ensemble du jeu de données QM9. À gauche : résultats sans filtrage de l’espace de recherche. À droite : résultats avec filtrage selon la contrainte sillywalks. Les paramètres EI, POI et id représentent la fonction de mérite utilisée et sont différenciés par les types de lignes. Les paramètres RBF et DotProduct correspondent à la fonction noyau utilisée ( $k_{\text{RBF}}$  et  $k_{\text{DOTPRODUCT}}$ ) et sont différenciés par la couleur des lignes. L’expérience basée sur un algorithme évolutionnaire uniquement (EvoMol) est représentée par un type de ligne et une couleur uniques.

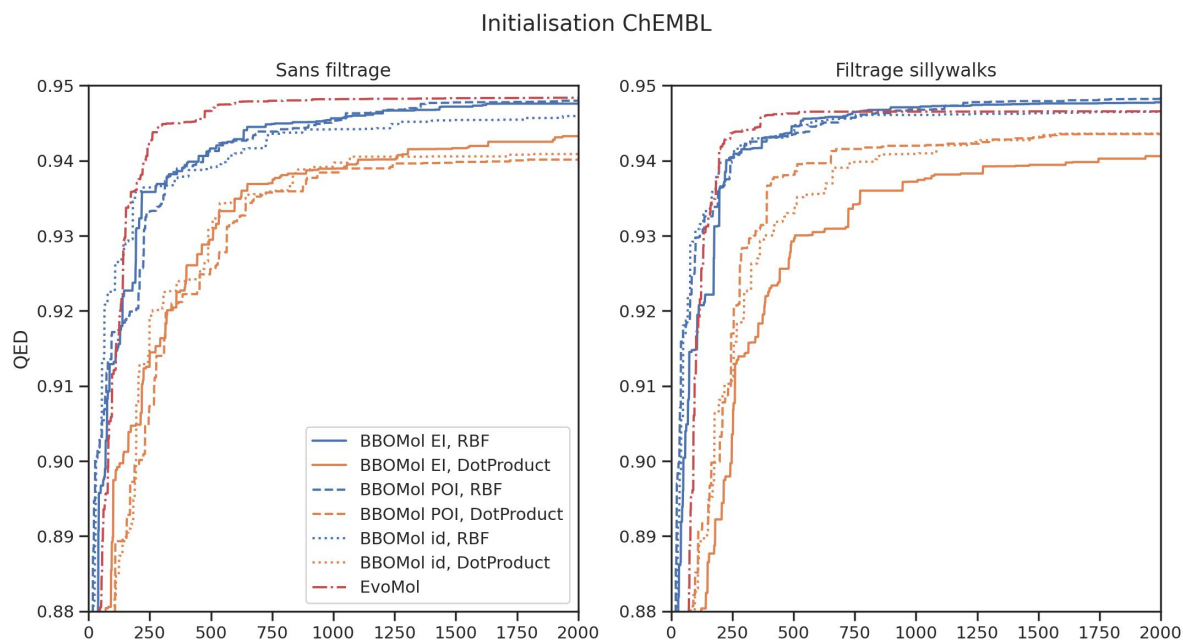


FIGURE 4.8 – Moyenne des meilleurs scores obtenus en fonction du nombre d'appels à la fonction objectif, pour différents paramétrages de BBOMol et pour EvoMol. La population initiale des expériences représentées est un sous-ensemble du jeu de données ChEMBL. À gauche : résultats sans filtrage de l'espace de recherche. À droite : résultats avec filtrage selon la contrainte sillywalks. Les paramètres EI, POI et id représentent la fonction de mérite utilisée et sont différenciés par les types de lignes. Les paramètres RBF et DotProduct correspondent à la fonction noyau utilisée ( $k_{\text{RBF}}$  et  $k_{\text{DOTPRODUCT}}$ ) et sont différenciés par la couleur des lignes. L'expérience basée sur un algorithme évolutionnaire uniquement (EvoMol) est représentée par un type de ligne et une couleur uniques.



importante sur l’efficacité de l’optimisation. Il est intéressant de noter que cela favorise également EvoMol. Cela signifie que le jeu de données ne permet pas seulement de prédire de façon plus précise les valeurs de fonction objectif, mais qu’il offre également des points de départ pertinents pour une recherche par voisinage. En Table 4.4, on observe qu’EvoMol requiert 527 appels à  $f$  pour obtenir une QED supérieure à 0.9, contre 161 pour BBOMol dans le cas le plus favorable. Pour résumer, l’utilisation d’un jeu de données initial extrait de QM9 plutôt que d’un jeu de données contenant seulement la molécule de méthane permet d’optimiser la valeur de QED de manière plus efficace, et notre approche basée sur un modèle de substitution demeure plus efficace qu’une approche purement évolutionnaire.

Il est intéressant d’étudier les résultats d’optimisation lorsque la population initiale est basée sur ChEMBL (voir la partie gauche de la Figure 4.8). L’utilisation de ChEMBL plutôt que QM9 favorise l’ensemble des exécutions. Cela peut s’expliquer par le fait que la QED est une mesure de similarité à des médicaments. Or, ChEMBL est une base de données très orientée pour la chimie pharmaceutique, contrairement à QM9 qui est une base synthétique pour la chimie organique en général. Les molécules de ChEMBL semblent donc plus pertinentes par rapport au problème d’optimisation que l’on souhaite résoudre. Notons également que la taille des molécules entre les deux jeux de données diffère grandement, puisque les molécules de QM9 possèdent jusqu’à 9 atomes lourds, tandis que les molécules de ChEMBL en possèdent en moyenne près de 30 (voir la section 1.1.4 de ce mémoire). Il est probable que ce facteur soit également déterminant, et que la taille des molécules favorise l’obtention de valeurs élevées de QED. Rappelons que dans le cas présent, ChEMBL a subi un filtrage pour ne conserver que les types d’atomes présents dans QM9, et pour éliminer les solutions possédant une valeur de QED supérieure à la valeur maximale de QM9 (0.669). Cela limite l’impact de ces facteurs. Toutefois, nous savons que ChEMBL contient une plus grande proportion de solutions proches de la limite de 0.669 (voir la Figure 4.2). Cela peut certainement avoir une influence positive sur les résultats.

Une autre observation très intéressante est que dans le cas présent, l’optimisation évolutionnaire est plus efficace que l’optimisation à l’aide d’un modèle de substitution. Cela signifie que bien que nous utilisions une version de ChEMBL dans laquelle les solutions correspondant aux plus hautes valeurs de QED sont filtrées, relativement peu de mutations sont nécessaires pour transformer une solution du jeu de données initial en une solution possédant une valeur de QED élevée. Cela signifie également que la mécanique déployée par BBOMol est moins efficace dans ce cas précis. Nous pouvons émettre plusieurs hypothèses pour expliquer cela. On peut supposer que l’exploration de l’espace de

recherche, promue à la fois par la fonction de mérite et son paramètre  $\xi$  non nul, est défavorable dans le cas où les solutions du jeu de données initial sont déjà très proches en nombre de mutations de très bonnes solutions. Nous verrons en étudiant la fonction de mérite identité ainsi que la variation de  $\xi$  que ces effets sont observables, mais qu'ils n'expliquent pas la totalité de la différence entre BBOMol et EvoMol dans ce cas précis.

L'autre hypothèse que nous formulons est que le modèle de substitution n'est pas assez précis pour prédire efficacement les valeurs de  $f$  lorsqu'elles sont proches de l'optimalité, et que l'approche en devient même moins efficace que l'exploration aléatoire effectuée par EvoMol lorsque peu de mutations sont nécessaires. Il est peu surprenant qu'un modèle de substitution fonctionnant par interpolation ne soit pas capable de prédire efficacement des valeurs extrêmes par rapport à la distribution représentée dans le jeu de données d'entraînement, en particulier lorsque les amplitudes d'améliorations attendues diminuent. Pour la QED, il devient relativement vite nécessaire de prédire efficacement les valeurs au centième puis au millième. Cet effet est observable en Table 4.4. Lorsque la valeur cible est de 0.9, BBOMol est autant voire plus efficace qu'EvoMol. Lorsque la cible et sa précision augmentent (0.94 puis 0.948), notre approche devient de moins en moins efficace relativement à EvoMol. Notons par ailleurs que cet effet s'observe pour les trois stratégies d'initialisation.

La stratégie d'initialisation possède une grande influence sur les performances de notre approche. Cela justifie des travaux ultérieurs pour déterminer des approches pertinentes pour la sélection du jeu de données initial pour l'espace des graphes moléculaires, ou plus généralement pour l'espace des molécules. Rappelons toutefois que notre étude est basée sur un problème synthétique pour évaluer à coût raisonnable l'effet de plusieurs paramètres de notre approche d'optimisation. L'efficacité de l'optimisation évolutionnaire de la QED en utilisant la stratégie d'initialisation basée sur ChEMBL est telle qu'il serait discutable d'utiliser une approche basée sur un modèle de substitution pour un problème réel lorsqu'un jeu de données aussi pertinent est disponible.

**Résultats : impact de la fonction de mérite et de la fonction noyau** Nous étudions désormais l'effet des trois fonctions de mérite et des deux fonctions noyau. Pour l'optimisation au départ du méthane (Figure 4.6), les exécutions basées sur les fonctions de mérite probabilistes (EI et POI) permettent d'obtenir des résultats comparables, avec un avantage possible pour EI (notamment associé à la fonction noyau  $k_{\text{RBF}}$ ). L'utilisation de la fonction identité offre des performances nettement inférieures, proches des résultats

Init.	Expérience	Sans contrainte			Contrainte sillywalks		
		0.9	0.94	0.948	0.9	0.94	0.948
Méthane	EI, $k_{\text{RBF}}$	<b>482</b> (10)	<b>1078</b> (10)	- (-)	<b>1842</b> (7)	<b>5587</b> (3)	<b>9748</b> (2)
	EI, $k_{\text{DOTPRODUCT}}$	674 (10)	1946 (7)	<b>8895</b> (2)	4355 (4)	- (-)	- (-)
	POI, $k_{\text{RBF}}$	958 (9)	1894 (7)	- (-)	9166 (2)	18735 (1)	19768 (1)
	POI, $k_{\text{DOTPRODUCT}}$	816 (10)	2223 (7)	- (-)	19192 (1)	- (-)	- (-)
	id, $k_{\text{RBF}}$	- (-)	- (-)	- (-)	- (-)	- (-)	- (-)
	id, $k_{\text{DOTPRODUCT}}$	5641 (3)	18767 (1)	18975 (1)	- (-)	- (-)	- (-)
	EvoMol	18583 (1)	19417 (1)	19625 (1)	- (-)	- (-)	- (-)
QM9	EI, $k_{\text{RBF}}$	162 (10)	<b>825</b> (10)	<b>3495</b> (5)	<b>188</b> (10)	<b>696</b> (10)	9063 (2)
	EI, $k_{\text{DOTPRODUCT}}$	214 (10)	1402 (8)	19746 (1)	304 (10)	2925 (5)	- (-)
	POI, $k_{\text{RBF}}$	<b>161</b> (10)	904 (10)	9512 (2)	230 (10)	743 (10)	<b>8871</b> (2)
	POI, $k_{\text{DOTPRODUCT}}$	243 (10)	2258 (6)	9274 (2)	229 (10)	2276 (6)	19247 (1)
	id, $k_{\text{RBF}}$	361 (10)	- (-)	- (-)	449 (10)	6143 (3)	- (-)
	id, $k_{\text{DOTPRODUCT}}$	261 (10)	19500 (1)	- (-)	572 (10)	18976 (1)	- (-)
	EvoMol	527 (9)	3099 (5)	4360 (4)	315 (10)	5307 (3)	- (-)
ChEMBL	EI, $k_{\text{RBF}}$	79 (10)	369 (10)	3414 (5)	84 (10)	232 (10)	3423 (5)
	EI, $k_{\text{DOTPRODUCT}}$	138 (10)	1362 (8)	18939 (1)	241 (10)	1860 (6)	- (-)
	POI, $k_{\text{RBF}}$	49 (10)	481 (10)	2608 (6)	<b>42</b> (10)	330 (10)	<b>1347</b> (9)
	POI, $k_{\text{DOTPRODUCT}}$	345 (9)	1093 (8)	8993 (2)	177 (10)	1116 (8)	- (-)
	id, $k_{\text{RBF}}$	<b>44</b> (10)	658 (9)	3876 (4)	46 (10)	<b>183</b> (10)	9109 (2)
	id, $k_{\text{DOTPRODUCT}}$	185 (10)	1880 (6)	8772 (2)	184 (10)	859 (9)	18225 (1)
	EvoMol	78 (10)	<b>241</b> (10)	<b>592</b> (10)	93 (10)	371 (9)	1472 (7)

TABLE 4.4 – Mesure d’espérance du coût de l’exécution en nombre d’appels à la fonction objectif (ERT) pour obtenir une solution ayant une QED au moins égale aux cibles 0.9, 0.94 et 0.948 pour différents paramétrages de BBOMol et pour EvoMol. Le nombre de fois que la cible a été atteinte parmi les 10 exécutions est indiqué entre parenthèses. La première colonne indique le jeu de données initial. La deuxième colonne indique la méthode utilisée et les paramètres de la recherche. Les paramètres EI, POI et id représentent la fonction de mérite. Les paramètres  $k_{\text{RBF}}$  et  $k_{\text{DOTPRODUCT}}$  correspondent à la fonction noyau. La partie de gauche des résultats correspond aux résultats sans filtrage de l’espace de recherche. La partie de droite correspond aux résultats lorsque l’espace de recherche est restreint aux solutions valides selon la contrainte sillywalks. Pour chaque stratégie d’initialisation et pour chaque colonne de résultats, la valeur d’ERT la plus faible est mise en évidence en gras. Les tirets correspondent aux valeurs non définies d’ERT (absence de succès).

d'EvoMol. Cette différence est attendue, puisque l'optimisation directe des valeurs du modèle de substitution est plus susceptible de mener vers des optimums locaux virtuels ne correspondant à des optimums de la fonction objectif  $f$  [JONES 2001]. De plus, cette stratégie n'intègre pas une façon de favoriser l'exploration de l'espace de recherche, ce qui limite les chances d'obtenir un meilleur modèle de  $f$  à l'étape suivante grâce aux points sélectionnés et insérés dans le jeu de données. Ces effets sont accentués ici par le fait que la population initiale contient un minimum de connaissance.

Lorsque la population initiale est basée sur QM9 ou ChEMBL (Figures 4.7 et 4.8), la différence entre les exécutions basées sur EI et POI est atténuée. Numériquement (voir la Table 4.4), il semble difficile de déterminer un paramètre plus efficace que l'autre, pour une fonction noyau donnée. Pour les exécutions basées sur QM9, la fonction identité est toujours moins efficace que EI et POI, et donne également des résultats comparables à ceux d'EvoMol. Pour ChEMBL en revanche, la fonction identité semble aussi efficace que les autres fonctions de mérite. Cela est un indice supplémentaire que le jeu de données initial basé sur ChEMBL contient déjà des connaissances très pertinentes pour l'optimisation de la QED.

Contrairement à la fonction de mérite, la fonction noyau semble avoir systématiquement un effet important pour l'ensemble des exécutions. Pour les trois figures étudiées dans les paragraphes précédents, les séries de données bleues, qui correspondent à l'utilisation d'un noyau  $k_{\text{RBF}}$ , sont quasiment systématiquement situées au dessus des lignes oranges correspondant à l'utilisation d'un noyau  $k_{\text{DOTPRODUCT}}$  pour une fonction de mérite donnée. Les valeurs d'ERT confirment cette tendance pour les trois valeurs de QED cibles, à quelques rares exceptions près.

**Résultats : impact du filtrage de l'espace de recherche** Pour conclure cette étude générale, nous étudions l'effet du filtrage de l'espace de recherche selon la contrainte silly-walks. Les Figures 4.6, 4.7 et 4.8 présentent (à droite) graphiquement ces résultats pour les différentes stratégies d'initialisation. La Table 4.4 (à droite également) représente quantitativement l'efficacité des expériences lorsque ces filtres sont activés. Lorsque l'initialisation est effectuée avec la molécule de méthane, il est intéressant de noter que l'utilisation de la contrainte détériore les performances de l'ensemble des exécutions. Lorsque cette stratégie d'initialisation est appliquée, les optimums locaux obtenus sont donc des molécules possédant des caractéristiques inconnues dans ChEMBL, et donc a priori peu réalistes. En Figure 4.9, nous présentons un sous-ensemble des solutions obtenues lors de

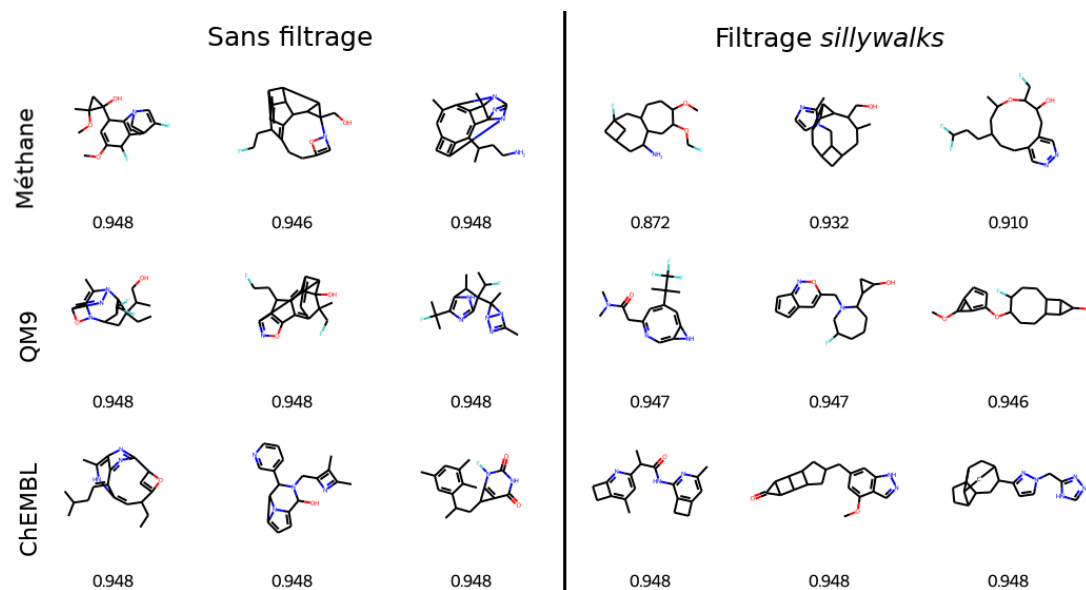


FIGURE 4.9 – Exemples de solutions obtenues par BBOMol avec et sans application de la contrainte *sillywalks* en fonction de la stratégie d'initialisation. Les résultats affichés correspondent à l'utilisation de la fonction de mérite EI et de la fonction noyau  $k_{\text{RBF}}$ . Pour chaque expérience, les solutions possédant la valeur de QED la plus élevée parmi 3 exécutions indépendantes sélectionnées aléatoirement sont reportées avec leur score.

ces expériences en fonction de l'initialisation et du filtrage. On y observe effectivement que dans le cas présent, les solutions possèdent des caractéristiques peu réalistes, et en particulier la présence de nombreux cycles imbriqués et de liaisons entre des atomes éloignés. Lorsque le filtrage de l'espace de recherche selon la contrainte *sillywalks* est activé, on remarque comme attendu que ces caractéristiques sont moins présentes et que les valeurs de QED sont moins élevées. Notons toutefois que les solutions obtenues dans le cadre de la restriction de l'espace de recherche par la contrainte *sillywalks* présentent également des caractéristiques peu réalistes. Il s'agit en particulier de la présence de cycles de taille relativement grande dépourvus de liaisons doubles et dans lesquels sont imbriqués des cycles de plus petite taille. Il s'agit de limites de ce type de filtrage que nous avons déjà identifiées en section 2.4 de ce mémoire.

Lorsque l'initialisation du jeu de données est effectuée à partir d'un sous-ensemble de QM9 ou de ChEMBL, l'effet du filtrage de l'espace de recherche selon la contrainte *sillywalks* est moins important. Certaines tendances semblent se dessiner lorsque l'on étudie comparativement les données à gauche et à droite des Figures 4.7 et 4.8 et de la Table 4.4, mais il semble difficile de déterminer à quel point elles sont significatives. Il

Paramètre	Valeur
Jeu de données initial	Méthane, QM9, ChEMBL
Fonction de mérite	POI, EI, id
Fonction noyau	$k_{\text{RBF}}$
Contrainte sur l'espace de recherche	Aucune
$\xi$	0.1, 0.01, 0

TABLE 4.5 – Grille de paramètres pour l'étude du paramètre d'exploration  $\xi$ .

semblerait que pour l'optimisation basée sur QM9 l'utilisation des filtres pénalise BBOMol lorsqu'il est associé au noyau  $k_{\text{DOTPRODUCT}}$  par rapport à EvoMol, en particulier au début de la recherche. Il semblerait pour l'optimisation basée sur ChEMBL que l'application de la contrainte pénalise plus les exécutions d'EvoMol que celles de BBOMol associé au noyau  $k_{\text{RBF}}$ . Il est difficile de conclure sur ces performances relatives, et il faut rappeler que toutes les exécutions étudiées ici sont très performantes. On observe en Figure 4.9 qu'avec et sans l'application des contraintes, les solutions ont des valeurs de QED très élevées. On observe également que les exécutions profitent des groupes chimiques typiques de la chimie du médicament présents dans le jeu de données initial (en particulier les cycles aromatiques, présents dans QM9 comme ChEMBL). L'utilisation de la contrainte sillywalks permet de conserver ces groupes pertinents pour la maximisation de la QED, tout en limitant les liaisons rendant les solutions irréalistes.

### Effet du paramètre d'exploration

Dans les expériences précédentes, nous avons utilisé une valeur fixe du paramètre  $\xi$ . Pour rappel, il s'agit d'un paramètre des fonctions de mérite EI et POI dont l'objectif est de favoriser l'exploration de l'espace de recherche. Nous souhaitons désormais étudier son influence pour l'optimisation de la QED en fonction de la stratégie d'initialisation du jeu de données, sur un sous-ensemble des paramètres étudiés dans l'expérience précédente. La Table 4.5 présente la grille de paramètres que nous utilisons pour cette nouvelle expérience. Nous avons précédemment utilisé une valeur par défaut de 0.01 pour  $\xi$ . Pour étudier son impact, nous choisissons d'étudier les résultats lorsque sa valeur est plus élevée (0.1) et lorsqu'elle est nulle. Pour limiter la combinatoire des paramètres, nous effectuons cette étude dans le cadre où la fonction noyau est fixe ( $k_{\text{RBF}}$ ) et où la contrainte sillywalks n'est pas appliquée sur l'espace de recherche. Nous effectuons à nouveau l'étude en faisant varier la stratégie d'initialisation et la fonction de mérite. Tous les autres paramètres sont inchangés.

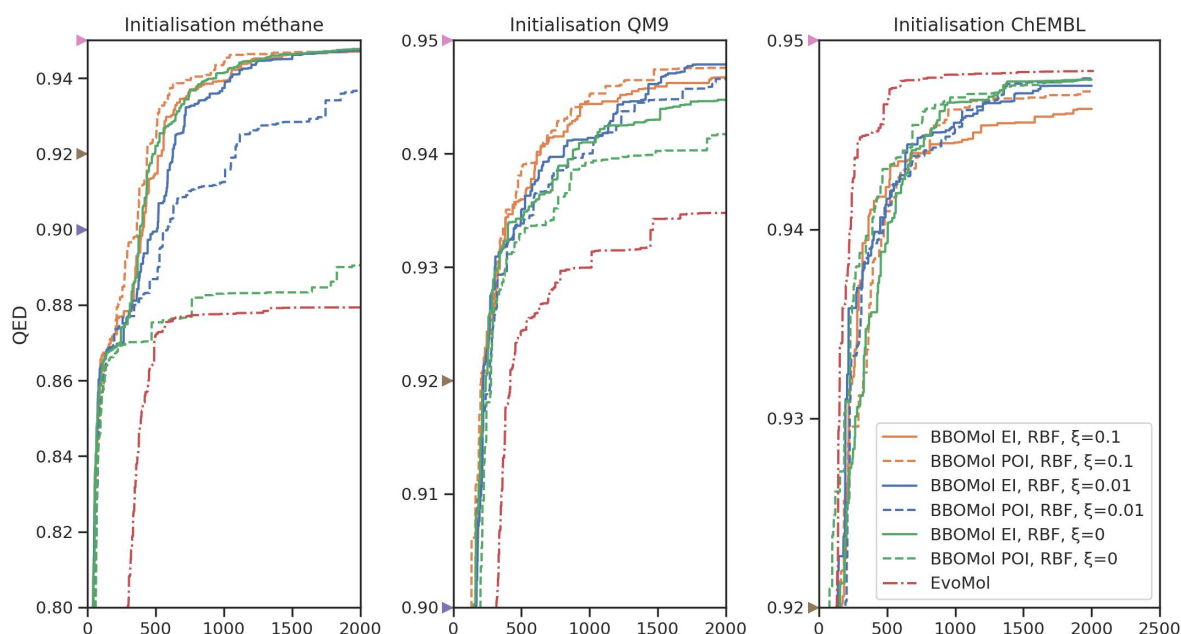


FIGURE 4.10 – Étude du paramètre d’exploration  $\xi$ . Moyenne des meilleurs scores obtenus en fonction du nombre d’appels à la fonction objectif, pour différents paramétrages de BBOMol et pour EvoMol. Chaque graphique correspond à l’utilisation d’un jeu de données initial différent (respectivement la molécule de méthane, un sous-ensemble de QM9 et un sous-ensemble de ChEMBL). Les paramètres EI et POI représentent la fonction de mérite utilisée et sont différenciés par les types de lignes. Toutes les expériences de BBOMol utilisent la fonction noyau  $k_{\text{RBF}}$ . Les différentes valeurs du paramètre d’exploration  $\xi$  sont différenciées par les couleurs des lignes. L’expérience basée sur un algorithme évolutionnaire uniquement (EvoMol) est représentée par un type de ligne et une couleur uniques. Les marqueurs sur les axes des ordonnées permettent de visualiser les changements d’échelle.

Les résultats de cette étude sont présentés graphiquement en Figure 4.10. La première observation que l'on peut faire est que lorsque l'initialisation du jeu de données est effectuée à partir du méthane ou de QM9, le paramètre  $\xi$  semble avoir une influence importante voire très importante lorsqu'il est associé à la fonction de mérite POI. Dans ce contexte, de meilleures solutions sont trouvées plus rapidement lorsque sa valeur augmente, en particulier avec l'initialisation depuis la molécule de méthane. Ces résultats semblent très cohérents car il est attendu que l'exploration de l'espace de recherche favorise l'obtention de bonnes solutions lorsque la connaissance sur la fonction objectif est faible.

L'effet de  $\xi$  sur la fonction de mérite EI semble en revanche nettement moins important, voire inexistant pour l'initialisation à partir du méthane. Il semble difficile d'expliquer ce phénomène de façon définitive, mais l'on peut supposer que l'utilisation de la fonction EI permet en soi de favoriser l'exploration par rapport à POI. Cette hypothèse est appuyée par le fait que pour les résultats au départ du méthane, les courbes représentant l'utilisation de EI sont très proches de la courbe représentant l'utilisation de POI avec une valeur de  $\xi$  élevée.

Lorsque l'initialisation est basée sur ChEMBL, l'effet de  $\xi$  semble faible pour l'ensemble des paramètres. On peut toutefois rappeler que dans ce cas tous les paramètres permettent d'obtenir d'excellentes performances. Bien qu'il soit difficile de conclure sur la significativité de ce résultat, il semble néanmoins que l'optimisation est favorisée par l'utilisation d'une valeur faible de  $\xi$ , ce qui revient à ne pas favoriser l'exploration de l'espace de recherche par rapport à l'intensification. Ce résultat formerait une tendance cohérente avec l'observation effectuée lorsque le jeu de données initial est basé sur le méthane, c'est-à-dire que plus le modèle de substitution possède de connaissance sur la fonction objectif, moins l'exploration de l'espace de recherche est nécessaire, voire plus elle est néfaste. Pour une étude plus poussée des résultats, il est possible de consulter en Annexe A.3 les valeurs d'ERT pour ces expériences.

### **Apport du modèle de substitution**

Pour terminer cette étude, nous souhaitons déterminer de façon plus précise l'effet de la connaissance apprise par le modèle de substitution sur l'efficacité de la recherche. Le modèle de substitution est en effet la pièce centrale de notre méthode d'optimisation, dont la connaissance permet d'espérer pouvoir effectuer une recherche plus efficace qu'une autre méthode « naïve » n'intégrant pas cette connaissance. Notre approche comprend cependant d'autres composants (la fonction de mérite, notre algorithme évolutionnaire EvoMol)



rendant plus difficile de mesurer l'apport produit par le seul modèle de substitution. On pourrait par exemple émettre l'hypothèse que les bonnes performances de notre approche en comparaison à un algorithme évolutionnaire ne seraient pas dues à la connaissance du modèle de substitution, mais à la mécanique induite par les fonctions de mérite qui intègrent une stratégie de modulation entre exploration et intensification de l'espace de recherche. Nous proposons d'étudier cet apport en effectuant un nouvel ensemble d'expériences, dans lequel nous privons le modèle de substitution de sa capacité à prédire les valeurs de QED de manière pertinente. Pour cela, nous choisissons d'utiliser comme descripteur moléculaire un vecteur aléatoire. Ainsi, il n'est pas possible d'associer de façon pertinente une zone de l'espace de description moléculaire avec un intervalle de valeurs de QED. Le vecteur est composé de valeurs tirées aléatoirement selon une loi normale centrée réduite.

Nous effectuons ces expériences avec les trois stratégies d'initialisation du jeu de données ainsi qu'avec les trois fonctions de mérite. Pour limiter la combinatoire des expériences, nous étudions seulement la fonction noyau  $k_{\text{RBF}}$ , le paramètre  $\xi$  est fixé à la valeur de 0.01 et nous n'appliquons pas la contrainte *sillywalks* à l'espace de recherche. Les autres paramètres sont les mêmes que pour les expériences précédentes. La dimension du descripteur est fixée à 2000, comme pour le vecteur de *shingles* étudié précédemment.

Les résultats de ces nouvelles expériences sont représentés en Figure 4.11, qui reprend également une partie des résultats présentés plus tôt dans ce chapitre. On observe pour les trois stratégies d'initialisation que les performances sont nettement inférieures, à la fois aux exécutions comparables de BBOMol, et à la fois à EvoMol. Il s'agit d'un résultat plutôt attendu, mais qui confirme l'importance de la connaissance apportée par le modèle de substitution, sans laquelle notre approche serait moins efficace qu'un algorithme évolutionnaire. EvoMol nécessite un nombre d'appels conséquent à la fonction objectif, mais garantit à chaque étape que les nouveaux individus sont des améliorants. L'expérience que nous menons ici correspond plutôt à une exploration aléatoire de l'espace de recherche. À chaque étape, des solutions sont obtenues par optimisation stochastique d'un modèle qui ne peut extraire de connaissance des données d'apprentissage.

Nous pouvons également observer que dans ce contexte la fonction de mérite ne semble pas avoir d'influence, les courbes des exécutions basées sur le descripteur moléculaire aléatoire étant très proches. Ce résultat n'est pas surprenant mais est intéressant. Cela semble confirmer que les expériences que nous menons sont comparables à une exploration aléatoire de l'espace de recherche. Il est donc très intéressant d'étudier les valeurs de

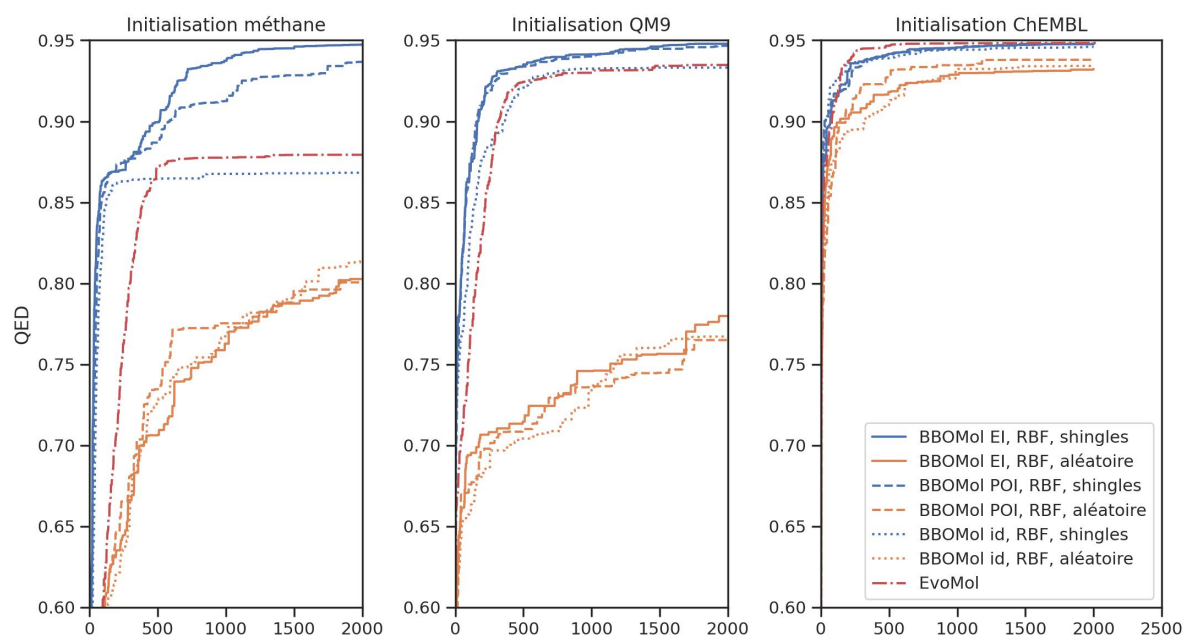


FIGURE 4.11 – Étude de l'influence du modèle de substitution, par ablation de sa capacité d'apprentissage en utilisant un descripteur moléculaire aléatoire. Moyenne des meilleurs scores obtenus en fonction du nombre d'appels à la fonction objectif, pour différents paramètres de BBOMol et pour EvoMol. Chaque graphique correspond à l'utilisation d'un jeu de données initial différent (respectivement la molécule de méthane, un sous-ensemble de QM9 et un sous-ensemble de ChEMBL). Les paramètres EI, POI et id représentent la fonction de mérite utilisée et sont différenciés par les types de lignes. Toutes les expériences de BBOMol utilisent la fonction noyau  $k_{\text{RBF}}$ . Les paramètres shingles et aléatoire représentent le descripteur moléculaire utilisé et sont différenciés par la couleur des lignes. L'expérience basée sur un algorithme évolutionnaire uniquement (EvoMol) est représentée par un type de ligne et une couleur uniques.

QED obtenues dans ce contexte, puisque cela peut permettre d'obtenir un référentiel plus « neutre » que les valeurs obtenues à l'aide d'EvoMol, qui correspond à un algorithme évolutionnaire spécifique.

On observe que pour l'initialisation basée sur la molécule de méthane ou sur QM9, des valeurs relativement proches de QED sont obtenues en moyenne (entre 0.75 et 0.8), bien qu'il soit clair que des valeurs plus élevées seraient obtenues en augmentant le budget d'appels à la fonction objectif. De manière surprenante, l'exécution au départ de la molécule de méthane est légèrement plus efficace. Il semble difficile d'expliquer cette observation, si ce n'est par la présence de molécules avec de faibles valeurs de QED dans QM9, et dont le voisinage serait également composé de molécules avec des faibles valeurs de QED. Lorsque l'initialisation est effectuée à partir de ChEMBL, on observe en revanche que les performances sont excellentes, même lorsque le descripteur aléatoire est utilisé. En moyenne, des solutions possédant des valeurs de QED supérieures à 0.9 sont obtenues en moins de 500 appels à la fonction objectif. Cela confirme les observations que nous avons effectuées au préalable, c'est-à-dire que ChEMBL, même lorsque l'on retire les molécules possédant des valeurs de QED élevées, contient des molécules dont le voisinage proche possède de très hautes valeurs de QED. Pour une étude plus poussée de ces résultats, il est possible de consulter en Annexe A.3 les valeurs d'ERT pour ces expériences.

### 4.4.3 Conclusion : apprentissage et optimisation de la QED

Nous avons mené une étude complète de notre approche d'optimisation boîte-noire basée sur un modèle de substitution dans le cadre d'un problème jouet, à savoir l'optimisation des valeurs de QED. Nous avons étudié la capacité des modèles de substitution à prédire les valeurs de QED, puis nous les avons intégrés au sein de notre algorithme pour étudier ses performances d'optimisation. Le faible coût de la QED nous a permis d'étudier un ensemble de paramètres de la recherche, qui inclut entre autres la stratégie d'initialisation du jeu de données. Nous avons observé que de façon générale, notre approche est plus efficace qu'un algorithme évolutionnaire et qu'une stratégie d'optimisation que l'on peut considérer comme une stratégie d'exploration aléatoire de l'espace de recherche. Nous avons mis en évidence l'effet de différents paramètres, et nous avons observé que le paramètre le plus déterminant est la stratégie de sélection du jeu de données initial. Plus la connaissance incluse dans le jeu de données initial est pertinente, meilleurs sont les résultats. La connaissance extraite de QM9, jeu de données moyennement pertinent pour l'optimisation de la QED permet d'obtenir de meilleurs résultats que la simple molécule

de méthane. La connaissance extraite de ChEMBL, jeu de données orienté pour la chimie pharmaceutique, permet d'obtenir d'excellents résultats d'optimisation. ChEMBL est un jeu de données très pertinent pour l'optimisation de la QED. Il est d'ailleurs si pertinent que nous pouvons discuter de l'intérêt d'une approche comme la nôtre dans le cas où un jeu de données aussi adapté existe. Pour la résolution d'un problème réel, une approche métaheuristique telle que notre algorithme EvoMol serait certainement suffisante dans un cas comparable. Ce problème jouet nous a toutefois permis d'évaluer notre approche et d'étudier en détail les effets des différents paramètres, sur un problème suffisamment réaliste.

## 4.5 Optimisation d'une propriété électronique

Nous souhaitons désormais évaluer notre approche dans un contexte plus réaliste, c'est-à-dire pour l'optimisation d'une propriété électronique dépendant d'un calcul coûteux en DFT. Nous souhaitons mesurer le gain d'efficacité que notre approche peut permettre d'obtenir en termes d'appels à la fonction objectif par rapport à un algorithme évolutionnaire. Nous souhaitons également vérifier que les coûts supplémentaires induits par le modèle de substitution sont compensés par ce gain, en nombre d'appels à la fonction objectif mais également en temps de calcul. Le coût de la fonction objectif implique cependant que nous mènerons une étude des paramètres moins exhaustive que lorsque nous avons optimisé les valeurs de QED dans la section précédente.

Comme dans le Chapitre 2, nous choisissons ici de mener une maximisation de l'énergie HOMO, qui correspond à l'énergie de l'électron de plus haute énergie dans la molécule. Rappelons que la maximisation de cette énergie ne résout pas un problème chimique réel en soi, puisque cela revient à chercher des molécules très réactives et donc peu stables. Cependant, l'énergie HOMO est un critère pouvant intervenir dans la formulation de problèmes d'optimisation pour la chimie des matériaux moléculaires organiques. Sa maximisation peut donc être considérée comme une première étape pertinente pour la résolution de problèmes d'optimisation réalistes dans ce domaine de la chimie.

Comme pour l'étude que nous avons menée pour l'optimisation de la QED, nous étudions dans un premier temps les modèles GPR pour la prédiction de l'énergie HOMO en dehors d'un contexte d'optimisation. Puis, nous intégrons ces modèles au sein de notre procédure d'optimisation afin de l'évaluer dans le cadre de la maximisation de l'énergie HOMO.

**Évolution de notre approche** Les expériences que nous effectuons pour l’étude de la maximisation de l’énergie HOMO sont semblables sur de nombreux points aux expériences que nous avons effectuées dans la section précédente pour la maximisation des valeurs de QED. Cependant, elles ont en pratique été effectuées à des périodes différentes de la thèse. Les résultats présentés dans la section précédente ont été obtenus plus récemment, après que des évolutions dans l’implémentation de notre algorithme et la conception des modèles de substitution ont eu lieu.

Nous choisissons de présenter dans cette section la version originale des résultats que nous avons obtenus pour la maximisation de l’énergie HOMO, et cela pour deux raisons. Premièrement, cela correspond aux résultats que nous avons publiés dans [LEGUY et al. 2021a]. Deuxièmement, cela nous évite de payer à nouveau le coût de calcul des expériences que nous présentons ici, qui dépendent d’évaluations DFT coûteuses. Précisons cependant que malgré les changements d’implémentation, les résultats que nous présentons sont toujours reproductibles en sélectionnant les options pertinentes, indiquées de manière explicite dans le dossier dédié à la reproduction de ces résultats sur notre dépôt GitHub<sup>4</sup>. De plus, précisons que si ces changements limitent les comparaisons directes que l’on peut faire entre les expériences d’optimisation de la QED et les expériences d’optimisation de l’énergie HOMO, ils n’altèrent pas les conclusions que l’on peut tirer de l’une ou de l’autre section.

Dans la suite de ce chapitre, nous précisons de manière explicite les points de changement par rapport aux expériences présentées précédemment. En l’absence de précision, les conditions expérimentales sont identiques.

### 4.5.1 Apprentissage et évaluation du modèle de substitution

Dans cette section, nous étudions nos modèles d’apprentissage pour la prédiction des valeurs d’énergie HOMO. Nous cherchons en particulier à mesurer les erreurs commises par le modèle ainsi que les temps pour l’apprentissage et la prédiction, en fonction de la taille du jeu de données d’entraînement.

**Jeux de données** Nous menons cette étude sur le jeu de données QM9. Ce choix est motivé par le fait qu’il s’agit d’un jeu de données communément utilisé pour l’étude de modèles d’apprentissage artificiel de propriétés électroniques. De plus, QM9 contient également les valeurs calculées en DFT de l’énergie HOMO (parmi d’autres propriétés)

---

4. <https://github.com/jules-leguy/BBOMol>

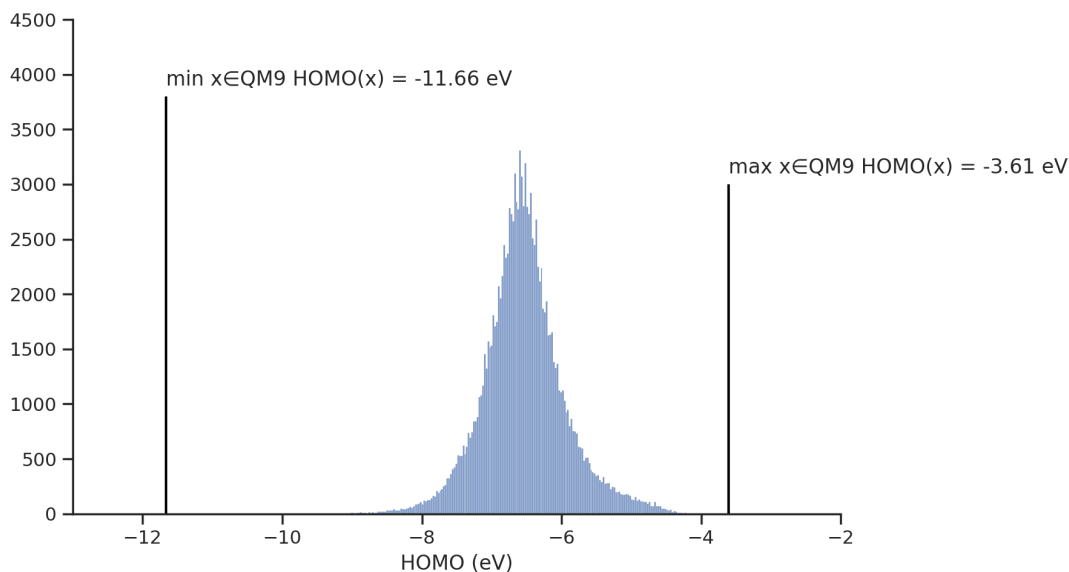


FIGURE 4.12 – Distribution des valeurs d'énergie HOMO sur le jeu de données QM9. Les deux lignes verticales indiquent les valeurs minimales et maximales.

des molécules contenues. Ainsi, il n'est pas nécessaire d'effectuer des calculs en DFT supplémentaires pour cette étude. Nous utilisons le même sous-ensemble de QM9 que pour les expériences précédentes, c'est-à-dire le sous-ensemble ne contenant ni radicaux, ni molécules contenant des atomes chargés. La Figure 4.12 présente la distribution des valeurs d'énergie HOMO sur le jeu de données QM9, ainsi que les valeurs extrêmes. La distribution semble assez proche d'une distribution gaussienne, avec néanmoins un léger déséquilibre des valeurs vers la droite. La moyenne calculée est de -6.54 eV.

**Descripteur moléculaire et fonction noyau** Le coût de calcul de l'énergie HOMO implique que nous devons limiter les expériences d'optimisation à l'aide de notre approche dans la section suivante. Nous choisissons d'ores et déjà de limiter notre étude à deux modèles GPR, qui nous serviront comme modèles de substitution par la suite. Un modèle GPR est défini principalement par sa fonction noyau, et la représentation moléculaire sur laquelle elle s'applique. Nous concevons deux combinaisons de ces éléments, formant deux modèles distincts. La première combine un noyau  $k_{\text{RBF}}$  avec le descripteur moléculaire MBTR. Ce modèle est défini avec des éléments relativement coûteux car l'exponentielle intervient dans le calcul du noyau et le descripteur moléculaire dépend d'une optimisation géométrique en mécanique moléculaire. Nous pouvons supposer que

l’information géométrique contenue par MBTR peut être pertinente pour la prédiction de l’énergie HOMO, très liée à la géométrie des molécules calculée par l’optimisation DFT. Le deuxième combinaison associe un noyau  $k_{\text{DOTPRODUCT}}$  avec le vecteur d’occurrences des *shingles*. Il s’agit d’un modèle que nous avons déjà étudié pour l’optimisation de la QED, et qui est conçu ici pour permettre une prédiction plus rapide de l’énergie HOMO, en comparaison au premier modèle.

Nous définissons un paramétrage de MBTR dans l’objectif que le descripteur ait une taille comparable au vecteur de *shingles*, c’est-à-dire proche de 2000. Chaque type d’atome est représenté par un vecteur de 10 valeurs réelles. Les distances entre chaque couple de types d’atomes sont représentées par 25 valeurs réelles. Il en est de même pour les angles entre chaque triplet de types d’atomes. Cette configuration correspond à une représentation contenant 2300 valeurs réelles. Les distributions gaussiennes intervenant dans le calcul de MBTR sont définies telles qu’elles ont une variance de 0.1. La géométrie moléculaire nécessaire au calcul de MBTR est calculée par la mécanique moléculaire telle qu’implémentée par la bibliothèque RDKit [LANDRUM 2010 ; TOSCO et al. 2014].

Nous estimons qu’il faut en moyenne 0.27 secondes pour le calcul du vecteur d’occurrences des *shingles* pour 100 molécules de QM9, contre 8.21 secondes pour le calcul de MBTR<sup>5</sup>. Le calcul de MBTR est nettement plus coûteux. Il faut cependant mettre cette valeur en perspective par rapport au coût de calcul de la DFT nécessaire à l’obtention de l’énergie HOMO. Pour des molécules de QM9, nous estimons en effet que ce coût est de l’ordre de la centaine de secondes pour une seule molécule (voir la section 1.1 de ce mémoire).

**Gestion du bruit et normalisation** Contrairement aux modèles que nous avons définis pour la prédiction des valeurs de QED, nous n’intégrons pas de fonction  $k_{\text{WHITE}}$  estimant dynamiquement la variance du bruit gaussien des données d’apprentissage au sein de la fonction noyau. Nous utilisons à la place une estimation fixe de la variance du bruit  $\sigma_n^2$ , insérée dans la diagonale du terme  $K(X, X)$  (voir la section 1.2 de ce mémoire). Nous menons dans la suite de cette section une étude pour déterminer une valeur appropriée de ce paramètre. Une autre différence avec les expériences précédemment effectuées est que nous n’appliquons pas de normalisation des valeurs cibles en entrée du modèle. Les modèles GPR prédisent donc les valeurs d’énergie HOMO sur leur étendue de valeurs

---

5. Pour produire cette estimation, nous mesurons le temps de calcul du descripteur pour 50 000 molécules tirées aléatoirement de QM9.

Modèle	$\sigma_n^2$	MAE inter-modèles	
		Moyenne	Écart-type
GPR $k_{\text{RBF}}$ , MBTR	0.001	5.41	0.51
GPR $k_{\text{DOTPRODUCT}}$ , shingles		0.24	0.01
GPR $k_{\text{RBF}}$ , MBTR	0.01	0.29	0.01
GPR $k_{\text{DOTPRODUCT}}$ , shingles		0.23	0.00
GPR $k_{\text{RBF}}$ , MBTR	<b>0.1</b>	0.27	0.00
GPR $k_{\text{DOTPRODUCT}}$ , shingles		0.23	0.00
GPR $k_{\text{RBF}}$ , MBTR	1	0.44	0.00
GPR $k_{\text{DOTPRODUCT}}$ , shingles		0.27	0.03
SchNet [SCHÜTT et al. 2018]	-	0.04	-

TABLE 4.6 – Moyenne et écart-type des valeurs de MAE mesurées sur les différents plis en fonction du modèle et de la valeur du paramètre  $\sigma_n^2$  pour la prédiction des valeurs d'énergie HOMO en eV. La valeur de  $\sigma_n^2$  sélectionnée suite à cette expérience est indiquée en gras ( $\sigma_n^2 = 0.1$ ). L'erreur moyenne du modèle SchNet est également reportée.

réelle.

### Étude des performances de prédiction

Nous définissons plusieurs expériences pour évaluer nos modèles GPR dans le cadre de la prédiction de l'énergie HOMO. Dans un premier temps, nous souhaitons déterminer une valeur appropriée du paramètre de bruit  $\sigma_n^2$ , tout en obtenant une estimation précise des erreurs commises par nos modèles. Nous concevons pour cela l'expérience suivante. Nous extrayons aléatoirement et sans remise 100 plis contenant chacun 1000 molécules de QM9. Pour chacun des deux couples (fonction noyau, descripteur moléculaire) et pour plusieurs valeurs possibles de  $\sigma_n^2$ , nous effectuons une validation croisée en utilisant itérativement chaque pli en tant que jeu de données d'entraînement. Chaque modèle est évalué sur l'ensemble des plis qui n'ont pas servi à son entraînement. Les valeurs de  $\sigma_n^2$  étudiées sont  $\{1, 0.1, 0.01, 0.001\}$ . Comme pour l'expérience précédemment effectuée pour l'étude de la prédiction des valeurs de QED, cela nous permet d'étudier les modèles lorsque les jeux de données d'entraînement contiennent 1000 molécules. Cela correspond à l'ordre de grandeur de la taille des jeux d'entraînement du modèle de substitution que nous utiliserons au sein de notre approche d'optimisation. Les valeurs d'énergie HOMO étant de l'ordre de grandeur de l'unité en électronvolts (voir la Figure 4.12), il est raisonnable d'attendre que l'ordre de grandeur de la variance du bruit des données soit dans la plage de valeurs que nous avons choisie.



Pour chaque modèle et pour chaque valeur de  $\sigma_n^2$ , nous reportons en Table 4.6 la moyenne et l’écart type de la MAE des modèles entraînés sur les différents plis. Sauf exception, les erreurs moyennes sont proches et sont assez faibles (de l’ordre de grandeur du dixième d’unité) en comparaison aux valeurs d’énergie HOMO observées dans le jeu de données. Sauf exception, nous observons également que les modèles sont très stables, l’écart-type de la MAE étant généralement dans l’ordre de grandeur du centième ou inférieur. Nous observons que les erreurs moyennes les plus faibles sont obtenues lorsque la variance du bruit est estimée à une valeur de 0.1 eV. Nous choisissons donc de sélectionner cette valeur du paramètre, que nous utiliserons dans les expériences suivantes au sein de cette section. En revanche, nous remarquons qu’une mauvaise estimation du bruit ( $\sigma_n^2 = 0.001$ ) peut mener à des erreurs très grandes et à un modèle nettement moins stable. Il s’agit donc d’un paramètre de grande importance. Finalement, on observe que le modèle basé sur le noyau  $k_{\text{DOTPRODUCT}}$  et le vecteur d’occurrences de *shingles* produit systématiquement des erreurs plus faibles que le modèle basé sur la fonction noyau  $k_{\text{RBF}}$  et MBTR, en plus de dépendre de composants moins coûteux.

Nous reportons également dans la Table 4.6 l’erreur moyenne absolue obtenue sur le jeu de données QM9 par un modèle d’apprentissage profond de référence de l’état de l’art, SchNet [SCHÜTT et al. 2018]. Précisons que nous n’effectuons pas de comparaison avec les modèles GPR de l’état de l’art, car ces derniers n’ont à notre connaissance pas été évalués pour la prédiction de l’énergie HOMO [BARTÓK et al. 2017; DERINGER et al. 2021]. L’erreur produite par SchNet est très faible, inférieure d’un ordre de grandeur à nos modèles. Cela montre une grande marge de progression potentielle pour nos modèles. Cette différence doit cependant être nuancée pour plusieurs raisons. Premièrement, SchNet a été entraîné par validation croisée sur la totalité de QM9, sur un jeu de données d’entraînement contenant 110 000 molécules, contre seulement 1000 molécules pour nos modèles. Il peut s’agir d’une première explication de la différence observée, mais nous devons également rappeler que nos modèles GPR ne pourraient pas être entraînés raisonnablement sur un jeu de données de cette taille. Deuxièmement, il faut rappeler que les valeurs d’énergie HOMO calculées ici par la DFT sont elles-mêmes des approximations théoriques des valeurs réelles qui peuvent être mesurées expérimentalement. Des travaux en chimie théorique ont montré que l’estimation par la DFT des valeurs d’énergie HOMO produit elle-même une erreur moyenne de 0.37 eV par rapport à des valeurs réelles [ZHANG et MUSGRAVE 2007]. Il semble pertinent de mettre les résultats en perspective avec cette valeur, car la signification d’une erreur de l’ordre du centième d’électronvolt n’est pas

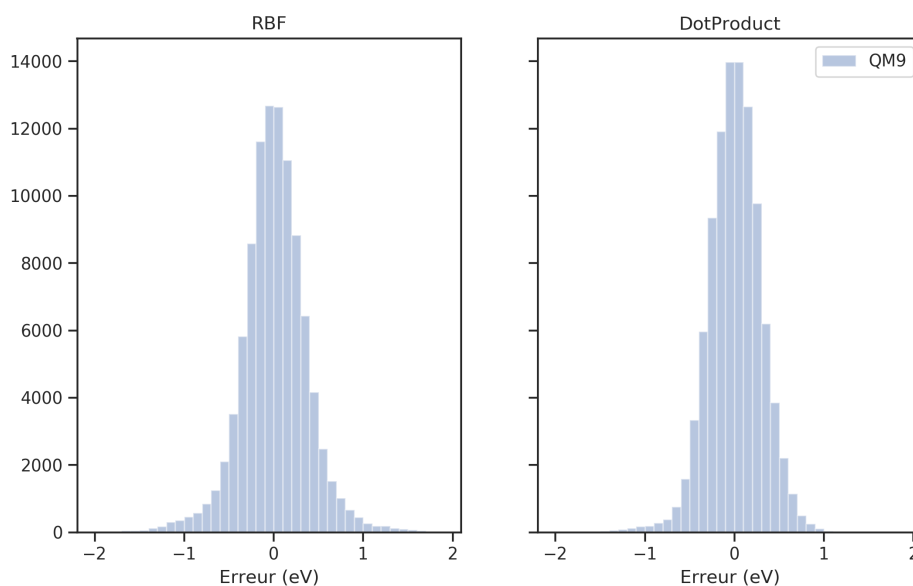


FIGURE 4.13 – Distribution des erreurs en eV du modèle ( $\sigma_n^2 = 0.1$ ) entraîné sur le premier pli pour la prédiction des valeurs d'énergie HOMO en fonction du jeu de données et de la fonction noyau.

évidente à saisir du point de vue de la propriété chimique estimée. En réalité, nous prédisons les valeurs d'un estimateur de cette propriété, pouvant lui-même être biaisé. Nous pouvons nous demander si l'apprentissage de cet estimateur avec une grande précision ne revient pas à apprendre une partie de ses éventuels biais. Finalement, il faut rappeler que notre objectif est de concevoir un modèle qui permettra de guider la recherche de solutions prometteuses. Ainsi, on ne cherche pas nécessairement à définir le modèle produisant les erreurs les plus faibles, mais plus généralement un modèle capable de prédire les valeurs de fonction objectif « suffisamment bien » pour proposer des candidats pertinents. Rappelons également que les modèles GPR permettent également une estimation de l'incertitude, qui est exploitée par notre approche d'optimisation pour une exploration plus pertinente de l'espace de recherche.

Pour la valeur de paramètre  $\sigma_n^2$  que nous avons sélectionnée (0.1), nous savons que l'erreur des modèles varie très peu en fonction des données d'entraînement. Nous souhaitons maintenant étudier les erreurs commises par un modèle sur son jeu de test. Nous représentons ainsi en Figure 4.13 la distribution des erreurs du modèle entraîné sur le premier pli et testé sur l'ensemble des autres plis. Nous observons que la distribution est

Fonction noyau	Erreur intra-modèle (eV)	
	Moyenne	Écart-type
$k_{\text{RBF}}$	0.01	0.37
$k_{\text{DOTPRODUCT}}$	0.01	0.30

TABLE 4.7 – Moyenne et écart-type en eV des erreurs enregistrées pour le modèle entraîné sur le premier pli pour la prédiction des valeurs de HOMO (paramètre  $\sigma_n^2 = 0.1$ ).

légèrement plus resserrée en son centre pour le modèle basé sur le noyau  $k_{\text{DOTPRODUCT}}$ , ce qui est cohérent avec les valeurs de MAE observées précédemment. Comme pour les résultats observés pour la prédiction des valeurs de QED, ces deux distributions semblent approximativement gaussiennes et semblent centrées sur la valeur 0. La Table 4.7 présente la moyenne et l’écart-type de ces distributions. En supposant qu’elles sont effectivement gaussiennes, elles sont paramétrées par une moyenne de 0.01 et un écart-type de 0.37 pour le modèle basé sur le noyau  $k_{\text{RBF}}$  et de 0.30 pour le modèle basé sur le noyau  $k_{\text{DOTPRODUCT}}$ .

Nous souhaitons désormais étudier l’évolution des erreurs et des temps d’entraînement et de prédiction de nos modèles, en fonction de la taille du jeu de données. Nous concevons pour cela une nouvelle expérience, très similaire à celle que nous avons effectuée pour la prédiction des valeurs de QED dans la section précédente. Nous séparons le jeu de données QM9 de façon aléatoire de sorte à obtenir un jeu de test de 50 000 molécules. Nous entraînons nos deux modèles GPR pour la prédiction de l’énergie HOMO sur des sous-ensembles de différentes tailles des données d’entraînement restantes. Les tailles varient parmi  $\{10, 100, 1000, 10000\}$ . Tous les modèles sont évalués sur le jeu de test.

La Figure 4.14 représente l’erreur moyenne absolue observée pour la prédiction de l’énergie HOMO en fonction de la taille du jeu de données d’entraînement. À nouveau, on constate que le modèle basé sur la fonction noyau  $k_{\text{DOTPRODUCT}}$  et le vecteur de *shingles* produit des erreurs plus faibles pour toutes les tailles de jeu de données d’entraînement. On constate que l’erreur moyenne absolue diminue efficacement lorsque la taille du jeu de données d’entraînement augmente, jusqu’à atteindre entre 0.25 et 0.30 eV environ, ce qui est cohérent avec les résultats obtenus précédemment.

Nous étudions ensuite l’évolution des temps d’exécution en fonction de la taille des jeux d’entraînement, dont les résultats sont représentés en Figure 4.15. Nous observons d’abord que les temps pour l’entraînement du modèle et la prédiction de 1000 molécules sont similaires aux temps observés pour les modèles de prédiction des valeurs de QED (voir la Figure 4.4). Cette observation est valable pour chacune des deux fonctions noyau, et s’explique par le fait que ces calculs sont indépendants du type de descripteur pour

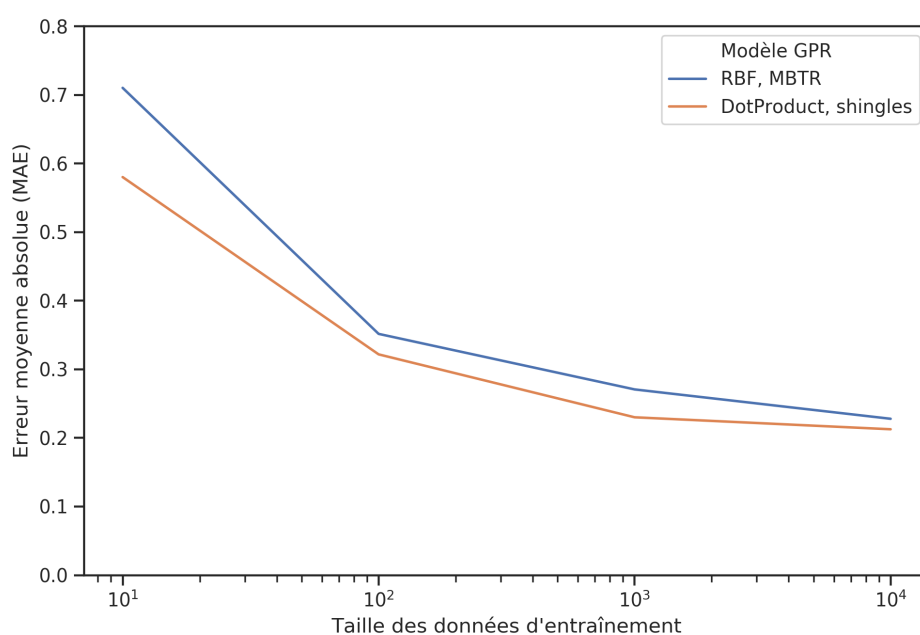


FIGURE 4.14 – Erreur moyenne absolue pour la prédiction de la valeur d'énergie HOMO en eV en fonction du modèle et de la taille du jeu de données d'entraînement, sur le jeu de données QM9. « RBF, MBTR » correspond à l'association de la fonction noyau  $k_{\text{RBF}}$  avec le descripteur MBTR. « DotProduct, shingles » correspond à l'association de la fonction noyau  $k_{\text{DOTPRODUCT}}$  avec le vecteur d'occurrences de *shingles*.

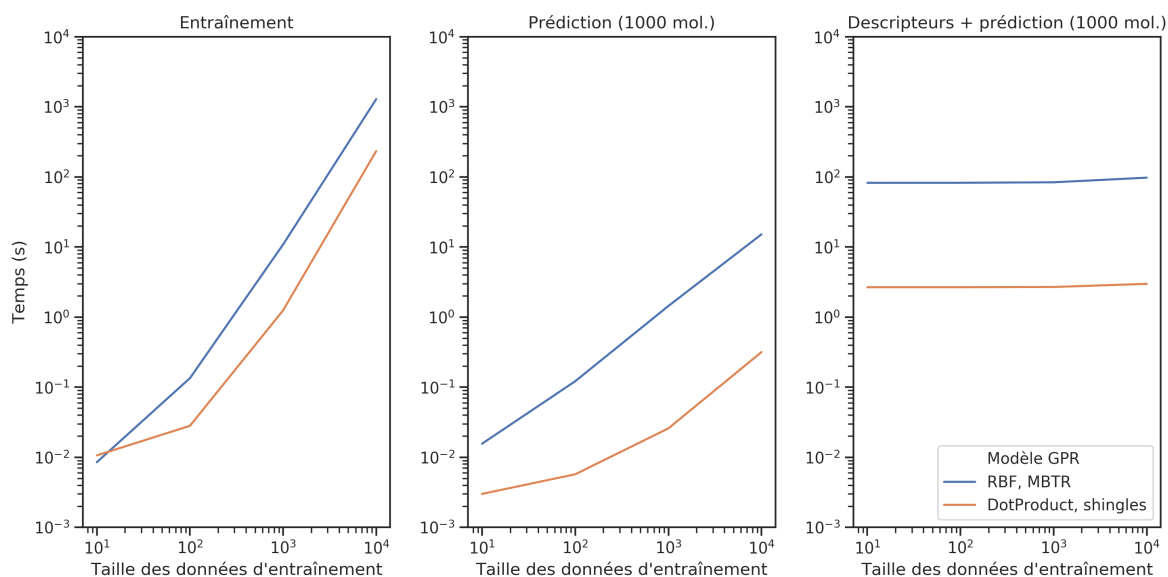


FIGURE 4.15 – À gauche : temps pour l’entraînement des modèles GPR de prédiction de l’énergie HOMO en fonction de la taille du jeu de données d’entraînement. Au centre : temps pour la prédiction de la valeur d’énergie HOMO de 1000 molécules en fonction de la taille du jeu de données d’entraînement à l’aide des modèles GPR. À droite : temps pour le calcul des descripteurs et la prédiction de la valeur d’énergie HOMO pour 1000 molécules en fonction de la taille du jeu de données d’entraînement à l’aide des modèles GPR. « RBF, MBTR » correspond à l’association de la fonction noyau  $k_{\text{RBF}}$  avec le descripteur MBTR. « DotProduct, shingles » correspond à l’association de la fonction noyau  $k_{\text{DOTPRODUCT}}$  avec le vecteur d’occurrences de *shingles*.

des descripteurs de tailles comparables. Nous observons toutefois que l’entraînement des modèles pour la prédiction de l’énergie HOMO est légèrement moins coûteux. Ce gain de temps s’explique par le fait que l’estimation du bruit des données d’apprentissage est effectuée dans le cas présent par un paramètre fixe, tandis qu’elle est effectuée de manière dynamique par la résolution d’un problème d’optimisation continue durant la phase d’apprentissage pour les modèles de prédiction des valeurs de QED.

Pour finir, nous étudions le temps pour l’estimation des valeurs d’énergie HOMO pour 1000 molécules, c’est-à-dire en considérant le calcul des descripteurs et la prédiction des valeurs (voir la partie droite de la Figure 4.15). On observe que le calcul est largement dominé par le temps de calcul des descripteurs, et que l’estimation des valeurs d’énergie HOMO de points de l’espace de recherche est donc quasiment indépendante de la taille des données d’entraînement.

## 4.5.2 Évaluation de notre méthode d'optimisation

Pour conclure cette série d'expériences, nous souhaitons maintenant étudier les performances de notre approche d'optimisation boîte-noire pour la maximisation de l'énergie HOMO. Nous souhaitons ainsi évaluer notre approche dans le cadre d'un problème d'optimisation plus réaliste que les expériences menées préalablement, dépendant d'une fonction d'évaluation réellement coûteuse. Nous cherchons à mesurer l'efficacité de notre méthode en termes de nombre d'appels à la fonction d'évaluation, ainsi qu'en termes de temps de calcul. Nous intégrons donc les deux modèles GPR étudiés précédemment en tant que modèles de substitution au sein de notre approche.

**Paramètres** En raison du coût de calcul de la propriété optimisée, nous limitons le nombre d'expériences différentes aux deux modèles GPR. Les autres paramètres ne varient pas. Nous choisissons ainsi de limiter cette étude à la fonction de mérite EI, qui est utilisée avec le paramètre d'exploration  $\xi$  fixé à 0.01. Les expériences sont effectuées dans l'espace de recherche correspondant à celui de QM9, c'est-à-dire l'espace des molécules contenant jusqu'à 9 atomes lourds parmi {C, N, O, F}. La géométrie initiale des optimisations DFT est obtenue par optimisation en mécanique moléculaire selon l'implémentation du champ de force MMFF94 du programme RDKit [Tosco et al. 2014].

**Sélection du jeu de données initial** Nous effectuons les expériences d'optimisation de l'énergie HOMO avec la stratégie qui consiste à utiliser comme jeu de données initial la seule molécule de méthane. La stratégie de sélection d'un sous-ensemble de ChEMBL ne serait pas adaptable directement pour ce problème, d'une part car la plupart des molécules de ChEMBL ne correspondent pas à l'espace de recherche que l'on considère ici, et d'autre part car cela nécessiterait le calcul en DFT de ces molécules. Il serait plus facilement envisageable d'utiliser un sous-ensemble de QM9, dont les molécules sont associées à leur calcul en DFT. Cependant, nous serions confrontés au même problème que pour l'étude menée précédemment pour l'optimisation des valeurs de QED à partir des données issues de ChEMBL, c'est-à-dire que le jeu de données contiendrait déjà des valeurs très hautes de la propriété cible. En effet, QM9 contient des molécules possédant des valeurs relativement élevées d'énergie HOMO (jusqu'à -3.61 eV, voir la distribution en Figure 4.12). Il serait envisageable d'utiliser comme nous l'avons fait pour la QED une stratégie de filtrage des solutions possédant des valeurs élevées de propriété cible, mais nous choisissons de limiter les expériences à la stratégie basée sur la molécule de méthane. Une première raison est

que nous cherchons à limiter les expériences effectuées pour limiter les coûts. Une seconde raison est que cette stratégie est néanmoins intéressante. La molécule de méthane possède en effet une énergie HOMO très faible de -10.6 eV. Il s'agit donc d'un point de départ très éloigné des valeurs que l'on souhaite atteindre, qui est susceptible de rendre le problème d'optimisation plus difficile et intéressant.

**Spécificité d'implémentation** Il existe une légère différence d'implémentation entre l'algorithme d'optimisation à l'aide d'un modèle de substitution utilisé pour l'optimisation des valeurs de QED dans la section précédente, et l'algorithme que nous utilisons ici pour l'optimisation de l'énergie HOMO. Dans le cas présent, les valeurs de fonction de mérite des individus de la population initiale lors des optimisations évolutionnaires de la fonction de mérite sont calculées comme nulles. Ainsi, tous les individus sélectionnés pour faire partie de la population initiale de l'algorithme évolutionnaire (voir la Figure 4.1) sont évalués comme ayant une valeur d'EI nulle. La différence pratique (mineure) que cela implique est que l'algorithme évolutionnaire ne favorisera pas pour produire des candidats les solutions connues possédant les valeurs d'EI les plus élevées au début de la procédure d'optimisation de la fonction de mérite.

**Comparaison à un algorithme évolutionnaire** Comme lors de l'évaluation de notre méthode pour l'optimisation des valeurs de QED, nous utilisons une optimisation évolutionnaire de l'énergie HOMO à l'aide d'EvoMol en tant qu'expérience de référence. Les mêmes paramètres sont utilisés, à l'exception de l'espace de recherche qui est limité aux solutions contenant jusqu'à 9 atomes lourds pour que les résultats soient comparables.

**Conditions expérimentales** Les trois expériences sont effectuées 10 fois chacune afin de pouvoir tirer des résultats moyens. Chaque exécution est limitée à un budget de 1000 appels à la fonction objectif. Toutes les exécutions sont effectuées sur des nœuds de calcul homogènes afin de permettre la comparaison des temps de calcul.

**Résultats** La Figure 4.16 représente graphiquement les résultats de nos expériences. La première observation que l'on peut faire est que les exécutions de notre algorithme sont effectivement plus efficaces que l'optimisation purement évolutionnaire. Elles semblent avoir convergé vers des valeurs supérieures à -3.0 eV, tandis que l'exécution d'EvoMol continue de progresser lorsque le budget d'appels est atteint. On peut également observer que l'exécution basée sur la fonction noyau  $k_{\text{DOTPRODUCT}}$  semble plus efficace au début de

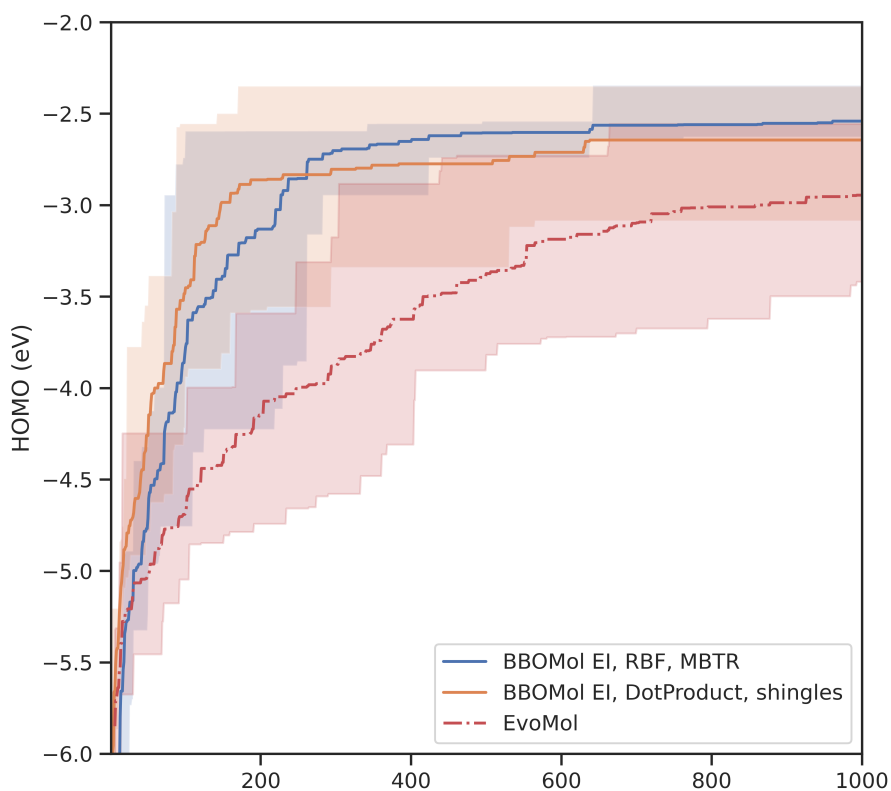


FIGURE 4.16 – Moyenne des valeurs d'énergie HOMO maximales obtenues parmi 10 exécutions en fonction du nombre d'appels à la fonction objectif. Les zones surlignées correspondent à l'étendue des valeurs maximales d'énergie HOMO. EI représente la fonction de mérite. RBF et DotProduct représentent la fonction noyau ( $k_{\text{RBF}}$  et  $k_{\text{DOTPRODUCT}}$  respectivement). Shingles et MBTR représentent le descripteur moléculaire.



Méthode	ERT (cible -3.0 eV)		Succès
	Appels à $f$	Temps de calcul (h)	
EI, $k_{\text{RBF}}$ , MBTR	177	11.3	10
EI, $k_{\text{DOTPRODUCT}}$ , <i>shingles</i>	377	27.9	9
EvoMol	1186	74.2	6

TABLE 4.8 – Mesure d’espérance du coût de l’exécution (ERT) pour l’obtention d’une solution possédant une énergie HOMO supérieure à -3.0 eV en nombre d’appels à la fonction objectif  $f$  ou en temps de calcul pour deux paramétrages de BBOMol et pour EvoMol. La colonne succès représente le nombre de fois que la cible a été atteinte parmi 10 exécutions. EI représente la fonction de mérite. RBF et DotProduct représentent la fonction noyau ( $k_{\text{RBF}}$  et  $k_{\text{DOTPRODUCT}}$  respectivement). *Shingles* et MBTR représentent le descripteur moléculaire.

la recherche, tandis que l’exécution basée sur la fonction  $k_{\text{RBF}}$  semble plus efficace à la fin de la recherche. Dans l’ensemble, ces deux exécutions obtiennent tout de même des performances très proches.

Nous étudions maintenant les résultats numériques présentés en Table 4.8. Cette dernière présente les valeurs d’ERT en nombre d’appels à la fonction objectif et en temps d’exécution pour obtenir une valeur cible élevée de -3.0 eV. On peut y observer que le gain de notre approche par rapport à EvoMol est d’un ordre de grandeur en termes de nombre d’appels selon cette métrique, ce qui nous semble être un résultat satisfaisant. Nous pouvons observer que ce résultat est en partie lié au fait que la cible n’est pas atteinte pour l’ensemble des exécutions, et qu’elle n’est en particulier atteinte que dans 6 des 10 exécutions d’EvoMol. En contradiction avec la lecture graphique que l’on aurait pu effectuer sur la Figure 4.16, le modèle basé sur le noyau  $k_{\text{RBF}}$  est le plus efficace selon cette métrique pour obtenir une solution possédant une valeur d’énergie HOMO au moins égale à -3.0 eV. Cela peut s’expliquer par le fait que l’obtention d’une cible correspond à un objectif binaire tandis que la moyenne représentée dans la Figure 4.16 prend en compte des valeurs proches mais inférieures à -3.0 eV dans le calcul de la moyenne affichée. Cela signifie de façon pratique qu’au moins une exécution du modèle basé sur  $k_{\text{DOTPRODUCT}}$  a tardé à atteindre la cible, tout en ayant proposé des solutions proches de la cible. Cela peut s’observer graphiquement dans les étendues de valeurs surlignées.

L’étude de l’efficacité en temps de calcul (Table 4.8) montre que notre approche peut obtenir selon cette métrique la valeur cible de -3.0 eV en environ un jour ou moins, tandis qu’il faut plus de trois jours pour l’optimisation évolutionnaire. Selon cette métrique également, le modèle basé sur le noyau  $k_{\text{RBF}}$  est le plus efficace. Il semble donc finalement

que le coût supplémentaire induit par le calcul du descripteur MBTR et de la fonction noyau  $k_{\text{RBF}}$  soit rentable en termes d'efficacité de la recherche. De manière plus générale, cela permet de vérifier que le coût induit par l'entraînement et l'utilisation d'un modèle de substitution dans notre approche d'optimisation est bien compensé par le gain d'efficacité qu'il permet.

### 4.5.3 Conclusion : apprentissage et optimisation de l'énergie HOMO

Dans cette section, nous avons proposé une étude de notre approche pour l'apprentissage et l'optimisation de l'énergie HOMO, une propriété électronique coûteuse. Cette étude peut être considérée comme une preuve de concept de notre approche pour l'optimisation efficace de propriétés moléculaires coûteuses en chimie des matériaux moléculaires organiques. Nous montrons que nos modèles d'apprentissage permettent d'avoir une estimation raisonnable des valeurs d'énergie HOMO, avec relativement peu de données d'entraînement. Nous intégrons ces modèles en tant que modèles de substitution au sein de notre approche d'optimisation, et montrons que cette dernière permet en effet une optimisation efficace de cette propriété moléculaire. Par rapport à une optimisation purement évolutionnaire, nous observons un gain d'un ordre de grandeur en appels à la fonction objectif pour obtenir une solution possédant une valeur d'énergie HOMO élevée selon la mesure ERT. En termes de temps de calcul, nous mesurons ce gain à plus de 48h, pour obtenir une solution possédant une valeur cible élevée en seulement 12h.

## 4.6 Conclusion et perspectives

Dans ce chapitre, nous proposons une méthode d'optimisation boîte-noire basée sur un modèle de substitution, dans l'objectif d'optimiser des propriétés moléculaires coûteuses de manière plus efficace que des approches d'optimisation traditionnelles. Notre approche est basée sur un modèle d'apprentissage artificiel, entraîné à prédire les valeurs de la fonction objectif coûteuse. Ce modèle est utilisé pour sélectionner des solutions candidates prometteuses, en substitution de la fonction objectif. Régulièrement, les candidats sélectionnés sont évalués par la fonction objectif, et insérés dans le jeu de données d'apprentissage du modèle de substitution. Ainsi, les procédures d'apprentissage et d'optimisation de la fonction objectif sont effectuées conjointement. Nous choisissons d'utiliser

un modèle GPR en tant que modèle de substitution, dont la prédiction probabiliste est mise à profit pour guider la recherche selon l'incertitude du modèle.

Nous montrons l'efficacité de notre approche par rapport à un algorithme évolutionnaire pour l'optimisation d'une propriété peu coûteuse issue de la chimie pharmaceutique, ainsi que pour l'optimisation d'une propriété dépendant de calculs coûteux en chimie quantique. Nous profitons du faible coût d'évaluation des valeurs de QED pour étudier de façon rigoureuse l'effet de différents paramètres de notre approche. Nous pouvons envisager plusieurs façons d'approfondir notre étude. Tout d'abord, nous avons utilisé comme méthode de référence l'algorithme évolutionnaire que nous avons proposé au Chapitre 2. Cela permet de comparer notre approche à une optimisation métaheuristique en termes d'efficacité de la recherche. Cependant, notre algorithme évolutionnaire n'est pas « universel ». Il intègre en effet des mécanismes qui lui sont propres, comme les stratégies de sélection des solutions mutées et remplacées. De plus, il est également utilisé au sein de l'approche que nous avons présentée dans ce chapitre pour permettre l'optimisation de la fonction de mérite dans l'espace moléculaire. Il pourrait être intéressant de chercher à découpler ces deux algorithmes, afin d'évaluer au mieux leurs performances respectives. Nous pourrions utiliser comme méthode de référence ou comme algorithme d'optimisation interne une métaheuristique plus simple, comme une recherche locale par *hill-climber*. Nous avons d'ailleurs montré au Chapitre 2 qu'il est possible de configurer notre algorithme évolutionnaire pour qu'il se comporte comme un *hill-climber*. Une stratégie d'exploration aléatoire de l'espace de recherche pourrait également être pertinente. Il est relativement difficile de définir ce genre de stratégie dans l'espace des graphes moléculaires car il est difficile de générer des graphes moléculaires valides de façon aléatoire. Nous pourrions cependant utiliser une heuristique simple qui s'y rapporterait comme le tirage aléatoire au sein d'un jeu de données moléculaires. Nous avons par ailleurs étudié les performances d'une stratégie qui se rapporte à la recherche aléatoire de solutions, lorsque le modèle de substitution est privé de sa capacité à prédire les valeurs de la propriété cible de façon pertinente.

Nous avons observé que la sélection du jeu de données initial est un facteur déterminant pour l'efficacité de la recherche. Dans les expériences que nous avons effectuées, cette connaissance n'a pas été un facteur limitant puisque de bonnes solutions aux problèmes d'optimisation ont pu être obtenues en temps raisonnable. Cependant, nous avons étudié l'optimisation de l'énergie HOMO, dépendant de calculs coûteux en DFT, dans le cadre d'un espace de recherche restreint à 9 atomes lourds parmi {C, N, O, F}. Or, nous savons

(voir la section 1.1 de ce mémoire) que le coût augmenterait drastiquement sans cette restriction. La taille de l'espace de recherche augmenterait également de fait. Il est donc probable qu'une stratégie efficace pour la sélection du jeu de données initial puisse être déterminante pour l'optimisation de propriétés électroniques dans un espace de recherche étendu.

Nous pouvons imaginer plusieurs stratégies pour la sélection pertinente d'un jeu de données initial. Les stratégies basées sur une mesure de distance pourraient être adaptées relativement facilement à l'espace moléculaire, puisqu'il est possible de définir une mesure de distance entre deux molécules. Il resterait alors à définir une façon de résoudre le problème d'optimisation *minimax* ou *maximin*, à l'aide d'une méthode d'optimisation moléculaire. Une difficulté est que dans ce cas la fonction objectif évalue l'ensemble du jeu de données et non des solutions individuellement. Une façon de résoudre ce problème pourrait être de définir une fonction objectif au niveau des solutions évaluant la contribution au score du jeu de données complet, de façon semblable à la mesure de contribution à la diversité que nous avons définie au Chapitre 3. Il serait également envisageable d'employer directement notre approche de maximisation de la diversité des descripteurs moléculaires en tant que stratégie d'initialisation du jeu de données. La diversité du jeu de données produit permettrait en effet d'obtenir la connaissance initiale de la fonction objectif sur des zones très variées de l'espace de recherche. Finalement, une sélection statistique peut également être envisagée, par maximisation de la mesure de *D-optimality* sur la matrice de covariance des modèles GPR que nous avons étudiés, par exemple. Dans ce cas également, l'objectif évalue l'ensemble du jeu de données, et il serait donc nécessaire de transformer le problème pour qu'une méthode d'optimisation dans l'espace moléculaire puisse le résoudre.

Une limitation potentielle de l'approche que nous avons développée provient du coût de calcul des modèles GPR. Ces derniers nécessitent en effet des calculs de complexité  $\mathcal{O}(n^3)$  avec  $n$  la taille du jeu de données d'entraînement. Ce coût limite l'utilisation de notre approche avec des jeux de données possédant une taille dans l'ordre de grandeur du millier. Pour traiter un éventuel problème nécessitant une quantité de données supérieure à l'aide de notre approche, il est envisageable d'utiliser une méthode d'approximation d'un modèle GPR permettant de limiter le coût de calcul total. Différentes approximations ont été définies dans la littérature [CANDELA et RASMUSSEN 2005], comme par exemple la méthode nommée « *projected process* » qui consiste à baser les calculs sur un sous-ensemble représentatif du jeu de données. Cette dernière a par ailleurs été utilisée par

[MUSIL et al. 2019] pour la prédiction de propriétés électroniques sur des sous-ensembles de QM9 de plus grandes tailles qu'au sein de nos travaux (20 000). Des travaux ont montré que des réseaux de neurones peuvent également être considérés comme une approximation d'un GPR [GAL et GHAHRAMANI 2016]. Cette technique a par ailleurs été utilisée par [SHAPEEV et al. 2020] pour obtenir une mesure d'incertitude pour la prédiction d'une propriété électronique sur QM9 à l'aide du modèle d'apprentissage profond SchNet. L'utilisation d'un tel modèle permettrait d'appliquer notre approche sur des jeux de données de grandes tailles. Cependant, cela pose la question du coût d'apprentissage du modèle initial, et nécessiterait que le modèle soit appris en ligne pour intégrer la connaissance obtenue au cours de la procédure d'optimisation. Ainsi, cela introduirait des problèmes supplémentaires qu'il ne semble pas trivial de résoudre actuellement.

Lors de notre étude, nous avons observé que notre approche est très efficace pour obtenir en peu d'appels à la fonction objectif des solutions possédant des valeurs de fonction objectif élevées. Nous avons également observé que notre algorithme évolutionnaire est plus efficace lorsque le jeu de données contient déjà des solutions très prometteuses. Nous pensons qu'il serait envisageable de combiner les deux approches pour maximiser l'efficacité tout au long de la recherche. Notre procédure d'optimisation basée sur le modèle de substitution pourrait en effet être utilisée en début de recherche pour mettre en évidence des candidats prometteurs, tant qu'elle permet des améliorations conséquentes. Au fil de la recherche, il est attendu que l'amplitude numérique de l'amélioration potentielle diminue par rapport à l'erreur du modèle. Lorsqu'elle devient plus faible, cela peut remettre en question la pertinence de l'utilisation d'un modèle de substitution. En fin de recherche, le passage à une méthode d'optimisation stochastique comme un algorithme évolutionnaire pourrait donc potentiellement améliorer l'efficacité globale de la recherche.

# CONCLUSION GÉNÉRALE ET PERSPECTIVES

---

Dans le cadre cette thèse, nous nous intéressons à la génération de molécules satisfaisant des propriétés moléculaires à l'aide de méthodes issues du domaine de l'intelligence artificielle. En particulier, nous cherchons à appliquer ces méthodes au domaine de la chimie des matériaux moléculaires organiques, qui correspond à la spécialité du laboratoire MOLTECH-Anjou avec lequel nous collaborons. Au sein de ce mémoire, nous proposons les contributions suivantes. Dans le Chapitre 2, nous proposons un algorithme évolutionnaire pour l'optimisation de propriétés moléculaires. Notre algorithme est conçu pour être générique, c'est-à-dire pour permettre l'optimisation de propriétés issues de différents domaines de la chimie. Pour cela, nous limitons au maximum les hypothèses sur l'espace de recherche intégrées dans la version de base de notre approche. Il est également conçu pour que ses résultats soient interprétables par des utilisateurs du domaine d'application. Nous proposons notamment une représentation sous forme d'un arbre d'exploration permettant à des spécialistes de la chimie d'étudier les liens entre structures et propriétés moléculaires lors de la recherche. Par sa conception, notre algorithme manifeste une prédisposition à l'intensification de l'espace de recherche par rapport à l'exploration. Il s'agit d'un comportement qui est avantageux dans de nombreux cas, car en chimie des molécules similaires ont souvent des propriétés similaires. L'intensification permet de déterminer la meilleure solution d'une zone de l'espace de recherche. Cependant, un compromis raisonnable entre intensification et exploration de l'espace de recherche doit être trouvé, au risque de pénaliser l'efficacité de la recherche. Dans nos expériences, nous observons également une tendance à la génération de molécules peu réalistes, ce qui est une caractéristique connue des algorithmes évolutionnaires. Nous étudions différentes approches pour améliorer le réalisme des molécules générées, notamment sous la forme d'une contrainte définie comme une liste blanche de caractéristiques locales qui permet de restreindre l'espace de recherche accessible. Nous observons empiriquement que cette contrainte permet d'améliorer sensiblement le réalisme des solutions. Toutefois, la localité du descripteur moléculaire considéré ne permet pas de prendre en compte certaines caractéristiques non

---

réalistes exprimées à l'échelle des cycles. Nous évoquons à la fin de cette conclusion des travaux menés très récemment au sein de notre équipe de recherche dans le but de prendre en compte ces caractéristiques.

Dans le Chapitre 3, nous proposons de générer un jeu de données moléculaires avec une forte diversité. Notre objectif initial est de produire un jeu de données qui pourra être utilisé en tant que jeu de données d'entraînement d'un modèle d'apprentissage de propriétés moléculaires. Pour cela, nous adaptons notre algorithme évolutionnaire à l'optimisation d'une mesure de la diversité moléculaire basée sur l'entropie de Shannon. Cette mesure étant définie à partir de l'ensemble des individus de la population, nous définissons un objectif de contribution à la diversité totale de la population qui permet d'évaluer l'apport des individus à l'objectif de diversité. Ce calcul est très coûteux. Nous en proposons un ensemble d'approximations qui rendent possible l'optimisation de l'objectif de diversité au sein d'une population de grande taille. Nous étudions également l'optimisation conjointe d'une propriété moléculaire cible et de l'objectif de contribution à la diversité moléculaire, dont le poids est pondéré par un paramètre. Nous montrons que cela peut permettre une optimisation plus efficace de la propriété cible en nombre d'appels à la fonction objectif. Cela est expliqué par le fait que l'objectif de diversité favorise l'exploration de l'espace de recherche. Le compromis entre intensification et exploration peut être réglé par le paramètre de pondération de l'objectif de diversité.

Nous abordons ensuite dans le Chapitre 4 le problème de l'optimisation de propriétés moléculaires coûteuses. Nous proposons une approche d'optimisation basée sur un modèle de substitution, qui correspond à un modèle d'apprentissage artificiel prédisant les valeurs de la propriété cible. L'objectif est d'utiliser le modèle de substitution pour sélectionner des candidats prometteurs dans l'espace de recherche à faible coût, selon une procédure d'optimisation pour laquelle nous proposons d'utiliser notre algorithme évolutionnaire. Nous utilisons un modèle de substitution probabiliste de régression par processus gaussien. La procédure d'optimisation consiste à maximiser une mesure dite d'espérance d'amélioration des solutions selon le modèle de substitution, ce qui permet de prendre en compte l'incertitude du modèle lors de la recherche. Régulièrement, les solutions candidates sont évaluées de manière exacte par la fonction objectif coûteuse, ce qui permet de mettre à jour les données d'entraînement du modèle de substitution. La procédure globale consiste à alterner des phases d'optimisation et d'apprentissage. Nous menons une étude approfondie de notre approche pour l'optimisation d'une propriété moléculaire peu coûteuse, ce qui nous permet d'étudier de nombreux paramètres. Le paramètre le plus déterminant est

---

la stratégie d’initialisation du jeu de données initial. Nous montrons également l’efficacité de notre approche pour l’optimisation d’une propriété électronique coûteuse, en nombre d’appels à la fonction objectif ainsi qu’en temps de calcul.

Les travaux présentés dans ce mémoire amènent un ensemble de perspectives. Nous en présentons plusieurs afin de conclure ce mémoire. Parmi les perspectives que nous pouvons envisager, nous avons eu l’occasion de mener des travaux préliminaires récents pour deux d’entre elles. Il s’agit de l’utilisation de notre algorithme évolutionnaire pour la génération d’explications de modèles d’apprentissage artificiel de propriétés moléculaires, et de la proposition d’un nouveau descripteur moléculaire adapté à la définition d’une contrainte permettant d’améliorer le réalisme des solutions générées par notre algorithme évolutionnaire. Nous présentons brièvement ces travaux, puis nous évoquons plusieurs perspectives supplémentaires qui peuvent être envisagées.

### **Explication contre-factuelle de modèles de classification moléculaire**

De nombreux modèles d’apprentissage artificiel ont été proposés pour la chimie moléculaire. Dans de nombreux cas, il s’agit de modèles d’apprentissage profond qui forment une boîte noire dont les sorties sont souvent très satisfaisantes, mais dont le comportement est très peu interprétable. Un enjeu important pour leur adoption par la communauté des chimistes reste la possibilité de les expliquer, ce qui permettrait de favoriser la confiance des utilisateurs humains pour ces modèles. Parmi les méthodes permettant de fournir des explications de modèles, il est envisageable de chercher à générer des explications contre-factuelles, qui permettent à des experts du domaine d’application d’étudier les frontières de décision des modèles sans nécessiter de compétences particulières en apprentissage artificiel [VERMA et al. 2020]. Cette approche est définie pour les problèmes de classification et en particulier pour les problèmes de classification binaire. En pratique, cela consiste à rechercher pour une instance donnée quelles modifications minimales de ses descripteurs conduirait à un changement de classe prédite. Dans le cas de notre application en chimie, on va chercher pour une molécule donnée un ensemble de transformations minimales du graphe moléculaire pour produire le changement de classe prédite. En partie (a) de la Figure 4.17, nous représentons schématiquement le lien entre des molécules dont on recherche des explications (ici de classe négative), leurs explications contre-factuelles (ici de classe positive) et la frontière de décision.

La génération d’explications contre-factuelles peut être définie comme un problème d’optimisation dont la fonction objectif prend en compte l’objectif de changement de



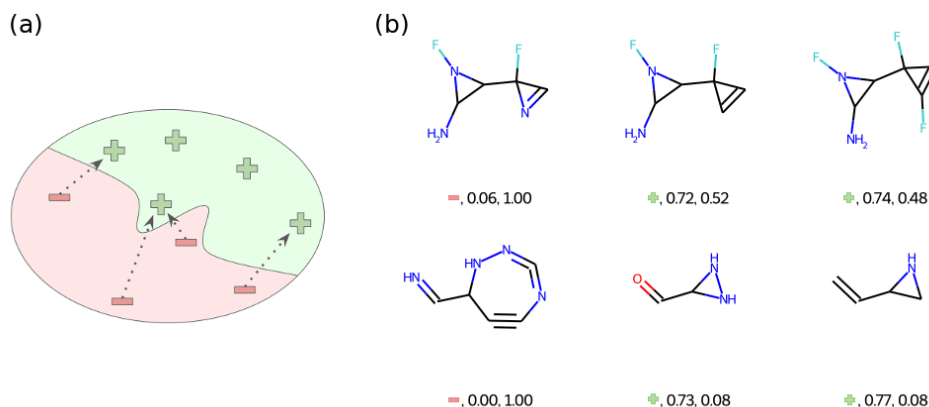


FIGURE 4.17 – (a) : Représentation schématique de la frontière de décision d’un modèle de classification binaire séparant un ensemble d’instances négatives et positives. Les flèches permettent d’identifier les explications contre-factuelles (instance de classe opposée la plus proche) des instances de classe négative. (b) : Exemples d’application de notre approche pour générer des explications d’un modèle prédictif de la stabilité moléculaire. La première colonne correspond à deux molécules de classe négative dont on cherche des explications. Les colonnes suivantes correspondent à des explications contre-factuelles de ces molécules, obtenues avec notre approche. Chaque molécule est étiquetée selon un triplet (classe prédite, valeur prédite par le modèle, similarité avec la molécule de départ).

classe et l’objectif de proximité avec la molécule de départ. Dans des travaux récents correspondant à une ouverture de fin de thèse vers de nouvelles applications de nos travaux, nous proposons de résoudre ce problème à l’aide de notre algorithme évolutionnaire. Ces travaux ont fait l’objet d’une présentation au sein de l’atelier EXPLAIN’AI à la conférence EGC en 2022 [LEGUY et al. 2022b]. En Annexe B, nous proposons une reproduction complète de l’article qui y a été présenté. Pour résumer brièvement ces travaux, nous proposons une fonction objectif qui considère la valeur prédite par le modèle de classification ainsi qu’une mesure de distance moléculaire. Nous supposons que le modèle de classification peut fournir une prédiction qui correspond à un nombre réel entre 0 (classe négative) et 1 (classe positive), avec la frontière de décision fixée à 0.5. La fonction objectif est composée d’un terme dont le but est de guider la recherche vers un changement de classe prédite, et d’un terme dont le but est de guider la recherche vers des solutions proches de la molécule de départ.

Nous appliquons notre approche pour l’explication d’un modèle de prédiction de la stabilité du calcul en chimie quantique. En partie (b) de la Figure 4.17, nous présentons des exemples d’explications contre-factuelles générées avec notre approche. Les molécules

---

en première colonne correspondent à des molécules prédites comme instables, tandis que les molécules des colonnes suivantes correspondent à leurs explications contre-factuelles. Il s’agit donc de molécules prédites comme stables et qui sont proches des molécules initiales. Cela démontre que notre approche peut identifier des transformations minimales qui conduisent à un changement de la classe prédite par le modèle. Par exemple, pour la première ligne, la classe prédite peut être transformée simplement en substituant un atome d’azote par un atome de carbone, ou en y liant en plus un atome de fluor. Les explications de la seconde ligne sont plus éloignées de la molécule de départ, ce qui laisse penser qu’il n’existe pas de voisins considérés stables par le modèle dans un voisinage proche.

Les explications contre-factuelles fournies par notre approche peuvent permettre aux utilisateurs du domaine d’application de comprendre quels facteurs influencent localement la propriété prédite par un modèle de classification. Il s’agit de travaux préliminaires amenant eux-mêmes un ensemble de perspectives. À titre d’exemple, nous pensons qu’il serait pertinent de proposer une visualisation des atomes et des liaisons d’une molécule qui sont le plus régulièrement altérés au sein des explications contre-factuelles. Cela permettrait d’identifier aisément quelles sont les parties des molécules qui sont responsables des changements de classe. Dans la littérature, des travaux proches ont été proposés peu après la présentation de nos travaux [WELLAWATTE et al. 2022].

## Descripteur moléculaire pour contraindre le réalisme des molécules

Afin de favoriser le réalisme des solutions générées par notre algorithme évolutionnaire, nous avons proposé en section 2.4 de ce mémoire une contrainte basée sur un descripteur moléculaire encodant les environnements locaux. Cette contrainte permet d’améliorer sensiblement le réalisme des solutions générées, mais ne permet pas de filtrer des caractéristiques indésirables définies à l’échelle des cycles. Dans des travaux très récents effectués au sein de notre équipe de recherche, nous proposons une nouvelle contrainte basée sur un descripteur moléculaire, que nous avons défini spécialement pour mettre en évidence ces caractéristiques.

Ce descripteur est nommé GCF (abréviation de « *generic cyclic feature* »). Il est défini à partir d’un ensemble de caractéristiques dont la procédure de génération est représentée en partie supérieure de la Figure 4.18. Elle consiste d’abord à extraire l’ensemble des composantes cycliques de la molécule qui sont représentées en partie (b). Les composantes cycliques sont obtenues en supprimant l’ensemble des liaisons correspondant à des ponts

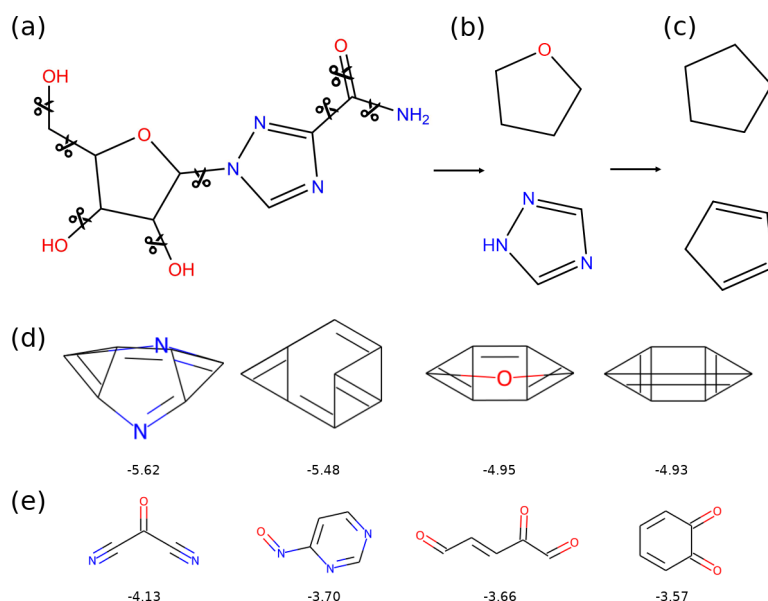


FIGURE 4.18 – Représentation du processus de génération des caractéristiques GCF (a, b, c) et étude de l'application de la contrainte GCF (d, e). La partie (a) représente la molécule de ribavirine sur laquelle l'action de suppression des ponts du graphe moléculaire est représentée par des paires de ciseaux. La partie (b) représente ses deux composantes cycliques, et la partie (c) représente ses deux caractéristiques GCF. La partie (d) correspond aux 4 meilleures solutions obtenues pour la minimisation de l'énergie LUMO avec la contrainte sillywalks, et la partie (e) correspond aux 4 meilleures solutions obtenues pour le même problème avec la contrainte sillywalks et la contrainte GCF. La légende des molécules correspond à la valeur d'énergie LUMO (eV).

du graphe moléculaire, c'est-à-dire dont la suppression augmente le nombre de composantes connexes. Ensuite, l'information des types d'atomes est retirée des composantes cycliques, ce qui permet d'obtenir les caractéristiques GCF représentées en partie (c). Pour que les caractéristiques GCF puissent être représentées comme des graphes moléculaires, tous les atomes sont transformés en atomes de carbone.

Nous définissons une contrainte à partir de ces caractéristiques. De façon très semblable à la contrainte sillywalks qui est basée sur la liste des caractéristiques ECFP4 qui existent au sein de ChEMBL, nous définissons la contrainte GCF qui est basée sur la liste des caractéristiques GCF qui existent au sein de ChEMBL. Cette liste est traitée comme une liste blanche de caractéristiques, ce qui signifie qu'une molécule est considérée valide selon la contrainte GCF si et seulement si toutes les caractéristiques GCF qu'elle contient existent dans ChEMBL. Cela permet de prendre en compte les caractéristiques liées aux

---

composantes cycliques, tout en permettant une certaine généralité puisque le type des atomes n'est pas considéré. Les types des liaisons sont en revanche bien pris en compte, car la stabilité d'une structure cyclique dépend souvent des contraintes géométriques imposées par les liaisons doubles ou triples.

En parties (d) et (e) de la Figure 4.18, nous représentons un exemple de l'effet de l'application de la contrainte GCF. Nous considérons le problème de la minimisation de l'énergie LUMO sur l'espace de recherche des molécules contenant jusqu'à 8 atomes lourds parmi {C, N, O, F}. En utilisant uniquement la contrainte sillywalks (partie d), nous pouvons obtenir des solutions possédant des valeurs d'énergie LUMO très basses mais qui sont composées d'assemblages de cycles qui sont très peu réalistes. En appliquant également la contrainte GCF (partie e), les solutions obtenues sont nettement plus réalistes. Bien que leurs valeurs d'énergies LUMO soient légèrement plus élevées, il s'agit de solutions qui sont plus pertinentes. Elles répondent en effet au problème d'optimisation et il peut être plus facilement envisagé de les synthétiser en pratique. Il s'agit d'ailleurs de molécules qui sont déjà connues dans la littérature en chimie, pour d'autres applications. Ces résultats n'ont pour le moment pas été publiés.

### Autres perspectives

Nous considérons désormais un ensemble de perspectives supplémentaires que nous pouvons envisager pour nos travaux et qui n'ont pas été explorées à l'heure actuelle. Pour l'optimisation évolutionnaire de propriétés moléculaires, nous utilisons actuellement une sélection aléatoire des opérateurs de mutation. Or, nous pouvons nous attendre à ce que certains opérateurs soient plus adaptés que d'autres à certains problèmes. Nous pouvons également nous attendre à ce que pour une propriété moléculaire donnée, certains opérateurs soient plus efficaces en fonction de l'état de la recherche. Nous pourrions considérer le problème de sélection des opérateurs comme un problème de bandit manchot à plusieurs bras, et appliquer une méthode d'apprentissage par renforcement pour leur sélection dynamique. Lors de l'optimisation conjointe de la diversité moléculaire et d'une propriété cible, le poids attribué à l'objectif de diversité qui permet de contrôler le compromis entre intensification et exploration doit actuellement être fixé avant le début de l'optimisation. Une procédure de sélection dynamique de ce paramètre au cours de la recherche pourrait être très bénéfique. Cela est d'autant plus vrai qu'il semble difficile de prévoir a priori les valeurs des paramètres qui seront efficaces pour un problème quelconque.

L'optimisation de la diversité moléculaire nous a permis de générer un jeu de données

---

avec une très forte diversité. Une partie importante des molécules générées n'est pas stable après calcul en mécanique quantique. Cela est expliqué simplement par le fait que l'objectif de diversité favorise la génération de caractéristiques instables et peu réalistes. Il serait très intéressant d'étudier la maximisation de la diversité en appliquant les contraintes de réalisme que nous avons définies. Cela pourrait produire un jeu de données de très grand intérêt en chimie-informatique, qui serait représentatif de la chimie réaliste tout en ayant un biais minimal vers un domaine particulier de la chimie. Les caractéristiques contenues dans les données résultantes seraient issues des jeux de données de référence utilisés pour définir les contraintes, mais leur distribution serait uniforme grâce à l'objectif de diversité.

Enfin, il serait intéressant d'appliquer notre approche d'optimisation basée sur un modèle de substitution à des problèmes plus réalistes en chimie des matériaux moléculaires. Cela impliquerait certainement de prendre en compte les valeurs de plusieurs propriétés moléculaires, mais également des contraintes géométriques telles que la symétrie ou la disposition planaire des atomes. Pour traiter ces problèmes plus complexes, nous pensons que l'axe majeur d'amélioration de notre approche concerne la procédure de sélection du jeu de données initial. Nous pensons que notre approche d'optimisation de la diversité moléculaire pourrait être utilisée dans ce contexte.

# BIBLIOGRAPHIE

---

- BARTÓK, Albert P., Sandip DE, Carl POELKING, Noam BERNSTEIN, James R. KERMODE, Gábor CSÁNYI et Michele CERIOTTI (déc. 2017), « Machine learning unifies the modeling of materials and molecules », in : *Science Advances* 3.12.
- BARTÓK, Albert P., Risi KONDOR et Gábor CSÁNYI (mai 2013), « On representing chemical environments », in : *Physical Review B* 87.
- BENHENDA, Mostapha (août 2017), « ChemGAN challenge for drug discovery : can AI reproduce natural chemical diversity ? », in : *arXiv :1708.08227 [cs, stat]*.
- BICKERTON, G. Richard, Gaia V. PAOLINI, Jérémy BESNARD, Sorel MURESAN et Andrew L. HOPKINS (fév. 2012), « Quantifying the chemical beauty of drugs », in : *Nature Chemistry* 4, p. 90-98.
- BOHACEK, Regine S., Colin McMARTIN et Wayne C. GUIDA (1996), « The art and practice of structure-based drug design : A molecular modeling perspective », in : *Medicinal Research Reviews* 16, p. 3-50.
- BOUSSAÏD, Ilhem, Julien LEPAGNOT et Patrick SIARRY (juill. 2013), « A survey on optimization metaheuristics », in : *Information Sciences, Prediction, Control and Diagnosis using Advanced Neural Computations* 237, p. 82-117.
- BROWN, Nathan, Marco FISCATO, Marwin H.S. SEGLER et Alain C. VAUCHER (mars 2019), « GuacaMol : Benchmarking Models for de Novo Molecular Design », in : *Journal of Chemical Information and Modeling* 59.3, p. 1096-1108.
- BROWNE, Cameron B, Edward POWLEY, Daniel WHITEHOUSE, Simon M LUCAS, Peter I COWLING, Philipp ROHLFSHAGEN, Stephen TAVENER, Diego PEREZ, Spyridon SAMOTHRAKIS et Simon COLTON (2012), « A survey of monte carlo tree search methods », in : *IEEE Transactions on Computational Intelligence and AI in games* 4, p. 1-43.
- BÜHLMANN, Sven et Jean-Louis REYMOND (fév. 2020), « ChEMBL-Likeness Score and Database GDBChEMBL », in : *Frontiers in Chemistry* 8.
- CANDELA, Joaquin Quiñonero et Carl Edward RASMUSSEN (2005), « A Unifying View of Sparse Approximate Gaussian Process Regression », in : *J. Mach. Learn. Res.* 6, p. 1939-1959.

- 
- CORNUÉJOLS, Antoine, Laurent MICLET et Vincent BARRA (2018), *Apprentissage artificiel*, 3e édition, Eyrolles.
- DE CAO, Nicola et Thomas KIPF (mai 2018), « MolGAN : An implicit generative model for small molecular graphs », in : *arXiv :1805.11973 [cs, stat]*.
- DERINGER, Volker L., Albert P. BARTÓK, Noam BERNSTEIN, David M. WILKINS, Michele CERIOTTI et Gábor CSÁNYI (août 2021), « Gaussian Process Regression for Materials and Molecules », in : *Chemical Reviews* 121, p. 10073-10141.
- DEVI, R. Vasundhara, S. Siva SATHYA et Mohane Selvaraj COUMAR (fév. 2015), « Evolutionary algorithms for de novo drug design – A survey », in : *Applied Soft Computing* 27, p. 543-552.
- ELTON, Daniel C., Zois BOUKOUVALAS, Mark D. FUGE et Peter W. CHUNG (2019), « Deep learning for molecular design—a review of the state of the art », in : *Mol. Syst. Des. Eng.* 4.4, p. 828-849.
- EMMERICH, M. T. M., K. C. GIANNAKOGLU et B. NAUJOKS (août 2006), « Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels », in : *IEEE Transactions on Evolutionary Computation* 10.4, p. 421-439.
- ERTL, Peter (juin 2017), « An algorithm to identify functional groups in organic molecules », in : *Journal of Cheminformatics* 9, p. 36.
- ERTL, Peter et Ansgar SCHUFFENHAUER (juin 2009), « Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions », in : *Journal of Cheminformatics* 1, p. 8.
- FABER, Felix A., Luke HUTCHISON, Bing HUANG, Justin GILMER, Samuel S. SCHOENHOLZ, George E. DAHL, Oriol VINYALS, Steven KEARNES, Patrick F. RILEY et O. Anatole von LILIENTHAL (nov. 2017), « Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error », in : *Journal of Chemical Theory and Computation* 13, p. 5255-5264.
- FREEDMAN, David H. (déc. 2019), « Hunting for New Drugs with AI », in : *Nature* 576, S49-S53.
- FRISCH, M. J., G. W. TRUCKS, H. B. SCHLEGEL, G. E. SCUSERIA, M. A. ROBB, J. R. CHEESEMAN, G. SCALMANI, V. BARONE, B. MENNUCCI, G. A. PETERSSON, H. NAKATSUJI, M. CARICATO, X. LI, H. P. HRATCHIAN, A. F. IZMAYLOV, J. BLOINO, G. ZHENG, J. L. SONNENBERG, M. HADA, M. EHARA, K. TOYOTA, R. FUKUDA, J. HASEGAWA, M. ISHIDA, T. NAKAJIMA, Y. HONDA, O. KITAO, H. NAKAI, T. VREVEN, J. A. MONTGOMERY Jr., J. E. PERALTA, F. OGLIARO, M. BEARPARK, J. J. HEYD,

- 
- E. BROTHERS, K. N. KUDIN, V. N. STAROVEROV, R. KOBAYASHI, J. NORMAND, K. RAGHAVACHARI, A. RENDELL, J. C. BURANT, S. S. IYENGAR, J. TOMASI, M. COSSI, N. REGA, J. M. MILLAM, M. KLENE, J. E. KNOX, J. B. CROSS, V. BAKKEN, C. ADAMO, J. JARAMILLO, R. GOMPERTS, R. E. STRATMANN, O. YAZYEV, A. J. AUSTIN, R. CAMMI, C. POMELLI, J. W. OCHTERSKI, R. L. MARTIN, K. MOROKUMA, V. G. ZAKRZEWSKI, G. A. VOTH, P. SALVADOR, J. J. DANNENBERG, S. DAPPRICH, A. D. DANIELS, Ö FARKAS, J. B. FORESMAN, J. V. ORTIZ, J. CIOSLOWSKI et D. J. FOX (2009), *Gaussian 09 Revision D.01*.
- FUSTER, Valentin et Joseph M. SWEENEY (fév. 2011), « Aspirin », in : *Circulation* 123, p. 768-778.
- GAL, Yarin et Zoubin GHAHRAMANI (juin 2016), « Dropout as a Bayesian approximation : representing model uncertainty in deep learning », in : *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, p. 1050-1059.
- GAULTON, Anna, Anne HERSEY, Michał NOWOTKA, A. Patrícia BENTO, Jon CHAMBERS, David MENDEZ, Prudence MUTOWO, Francis ATKINSON, Louisa J. BELLIS, Elena CIBRIÁN-UHALTE, Mark DAVIES, Nathan DEDMAN, Anneli KARLSSON, María Paula MAGARIÑOS, John P. OVERINGTON, George PAPADATOS, Ines SMIT et Andrew R. LEACH (jan. 2017), « The ChEMBL database in 2017 », in : *Nucleic Acids Research* 45.D1.
- GEBAUER, Niklas W. A., M. GASTEGGER et Kristof T. SCHÜTT (2019), « Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules », in : *NeurIPS*.
- GENHEDEN, Samuel, Amol THAKKAR, Veronika CHADIMOVÁ, Jean-Louis REYMOND, Ola ENKQVIST et Esben BJERRUM (nov. 2020), « AiZynthFinder : a fast, robust and flexible open-source software for retrosynthetic planning », in : *Journal of Cheminformatics* 12, p. 70.
- GLAVATSKIKH, Marta, Jules LEGUY, Gilles HUNAULT, Thomas CAUCHY et Benoit DAMOTA (déc. 2019), « Dataset's chemical diversity limits the generalizability of machine learning predictions », in : *Journal of Cheminformatics* 11.1, p. 69.
- GLOVER, Fred W. (1989), « Tabu Search - Part I », in : *INFORMS J. Comput.* 1, p. 190-206.
- GOODFELLOW, Ian, Jean POUGET-ABADIE, Mehdi MIRZA, Bing XU, David WARDEFARLEY, Sherjil OZAIR, Aaron COURVILLE et Yoshua BENGIO (2014), « Generative



- 
- Adversarial Nets », in : *Advances in Neural Information Processing Systems*, t. 27, Curran Associates, Inc.
- GUBAEV, Konstantin, Evgeny V. PODRYABINKIN et Alexander V. SHAPEEV (juin 2018), « Machine learning of molecular properties : Locality and active learning », in : *The Journal of Chemical Physics*.
- GUIMARAES, Gabriel Lima, Benjamin SANCHEZ-LENGELING, Carlos OUTEIRAL, Pedro Luis Cunha FARIAS et Alán ASPURU-GUZIŁ (fév. 2018), « Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models », in : *arXiv :1705.10843 [cs, stat]*.
- GUTMANN, H.-M. (mars 2001), « A Radial Basis Function Method for Global Optimization », in : *Journal of Global Optimization* 19.3.
- GÓMEZ-BOMBARELLI, Rafael, David DUVENAUD, José Miguel HERNÁNDEZ-LOBATO, Jorge AGUILERA-IPARRAGUIRRE, Timothy D. HIRZEL, Ryan P. ADAMS et Alán ASPURU-GUZIŁ (2016), « Automatic chemical design using a data-driven continuous representation of molecules », in : *arxiv abs/1610.02415v2*.
- GÓMEZ-BOMBARELLI, Rafael, Jennifer N. WEI, David DUVENAUD, José Miguel HERNÁNDEZ-LOBATO, Benjamín SÁNCHEZ-LENGELING, Dennis SHEBERLA, Jorge AGUILERA-IPARRAGUIRRE, Timothy D. HIRZEL, Ryan P. ADAMS et Alán ASPURU-GUZIŁ (fév. 2018), « Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules », in : *ACS Central Science* 4, p. 268-276, ISSN : 2374-7943.
- HAGBERG, Aric A., Daniel A. SCHULT et Pieter J. SWART (2008), « Exploring Network Structure, Dynamics, and Function using NetworkX », in : *Proceedings of the 7th Python in Science Conference*, sous la dir. de Gaël VAROQUAUX, Travis VAUGHT et Jarrod MILLMAN, Pasadena, CA USA, p. 11 -15.
- HAIDER, Norbert (août 2010), « Functionality Pattern Matching as an Efficient Complementary Structure/Reaction Search Tool : an Open-Source Approach », in : *Molecules* 15, p. 5079-5092.
- HALGREN, Thomas A. (1996), « Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94 », in : *Journal of Computational Chemistry* 17, p. 490-519.
- HANSEN, Nikolaus, Anne AUGER, Raymond ROS, Olaf MERSMANN, Tea TUŠAR et Dimo BROCKHOFF (jan. 2021), « COCO : A Platform for Comparing Continuous Optimizers in a Black-Box Setting », in : *Optimization Methods and Software* 36, p. 114-144.

- 
- HIMANEN, Lauri, Marc O. J. JÄGER, Eiaki V. MOROOKA, Filippo FEDERICI CANOVA, Yashasvi S. RANAWAT, David Z. GAO, Patrick RINKE et Adam S. FOSTER (fév. 2020), « Dscribe : Library of descriptors for machine learning in materials science », in : *Computer Physics Communications* 247.
- HOCHREITER, Sepp et Jürgen SCHMIDHUBER (nov. 1997), « Long Short-Term Memory », in : *Neural Computation* 9, p. 1735-1780.
- HOLLAND, John H (1992), *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*, MIT press.
- HOMMA, Kenji, Yu LIU, Masato SUMITA, Ryo TAMURA, Naoki FUSHIMI, Junichi IWATA, Koji TSUDA et Chioko KANETA (juin 2020), « Optimization of a Heterogeneous Ternary Li<sub>3</sub>PO<sub>4</sub>-Li<sub>3</sub>BO<sub>3</sub>-Li<sub>2</sub>SO<sub>4</sub> Mixture for Li-Ion Conductivity by Machine Learning », in : *The Journal of Physical Chemistry C* 124.24, Publisher : American Chemical Society, p. 12865-12870.
- HUO, Haoyan et Matthias RUPP (jan. 2018), « Unified Representation of Molecules and Crystals for Machine Learning », in.
- HÜCKEL, Erich (mars 1931), « Quantentheoretische Beiträge zum Benzolproblem », in : *Zeitschrift für Physik* 70, p. 204-286.
- IRWIN, John J., Khanh G. TANG, Jennifer YOUNG, Chinzorig DANDARCHULUUN, Benjamin R. WONG, Munkhzul KHURELBAATAR, Yurii S. MOROZ, John MAYFIELD et Roger A. SAYLE (déc. 2020), « ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery », in : *Journal of Chemical Information and Modeling* 60.12.
- JACCARD, P (1901), « Distribution de la flore alpine dans le Bassin des Dranses et dans quelques regions voisines », in : *Bull Soc Vaudoise Sci Nat* 37, p. 241-272.
- JENSEN, Jan H. (2019), « A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space », in : *Chemical Science* 10.12, p. 3567-3572.
- JOHNSON, M. E., L. M. MOORE et D. YLVIKAKER (oct. 1990), « Minimax and maximin distance designs », in : *Journal of Statistical Planning and Inference* 26.2, p. 131-148.
- JONES, Donald R. (déc. 2001), « A Taxonomy of Global Optimization Methods Based on Response Surfaces », in : *Journal of Global Optimization* 21.4, p. 345-383.
- JONES, Donald R., Matthias SCHONLAU et William J. WELCH (déc. 1998), « Efficient Global Optimization of Expensive Black-Box Functions », en, in : *Journal of Global Optimization* 4, p. 455-492.

- 
- JONES, Gareth (1998), « Genetic and evolutionary algorithms », in : *Encyclopedia of Computational Chemistry* 2, p. 1127-1136.
- JOUNG, InSuk, Jong Yun KIM, Steven P. GROSS, Keehyoung JOO et Jooyoung LEE (fév. 2018), « Conformational Space Annealing explained : A general optimization algorithm, with diverse applications », in : *Computer Physics Communications*, p. 28-33.
- KANG, Beomchang, Chaok SEOK et Juyong LEE (oct. 2021), « MOLGENGO : Finding Novel Molecules with Desired Electronic Properties by Capitalizing on Their Global Optimization », in : *ACS Omega*.
- KENNEDY, James et Russell EBERHART (1995), « Particle swarm optimization », in : *Proceedings of ICNN'95-international conference on neural networks*, t. 4, IEEE, p. 1942-1948.
- KERSTJENS, Alan et Hans DE WINTER (jan. 2022), « LEADD : Lamarckian evolutionary algorithm for de novo drug design », in : *Journal of Cheminformatics* 14.1, p. 3.
- KIM, Sunghwan, Jie CHEN, Tiejun CHENG, Asta GINDULYTE, Jia HE, Siqian HE, Qingliang LI, Benjamin A SHOEMAKER, Paul A THIESSEN, Bo YU, Leonid ZASLAVSKY, Jian ZHANG et Evan E BOLTON (jan. 2021), « PubChem in 2021 : new data content and improved web interfaces », in : *Nucleic Acids Research* 49.
- KINGMA, Diederik P. et Max WELING (mai 2014), *Auto-Encoding Variational Bayes*.
- KRENN, Mario, Florian HÄSE, AkshatKumar NIGAM, Pascal FRIEDERICH et Alan ASPURUGUZIK (nov. 2020), « Self-referencing embedded strings (SELFIES) : A 100% robust molecular string representation », in : *Machine Learning : Science and Technology* 1.
- KWON, Yongbeom et Juyong LEE (mars 2021), « MolFinder : an evolutionary algorithm for the global optimization of molecular properties and the extensive exploration of chemical space using SMILES », in : *Journal of Cheminformatics* 13, p. 24.
- LANDRUM, Greg (2010), *RDKit : Open-source cheminformatics*, URL : <http://www.rdkit.org>.
- LECUN, Yann, Yoshua BENGIO et Geoffrey HINTON (mai 2015), « Deep learning », in : *Nature* 521, p. 436-444.
- LEGUY, Jules, Thomas CAUCHY, Béatrice DUVAL et Benoit DA MOTA (jan. 2022a), « Chapter 2 - Goal-directed generation of new molecules by AI methods », in : *Computational and Data-Driven Chemistry Using Artificial Intelligence*, sous la dir. de Takashiro AKITSU, Elsevier, p. 39-67.

- 
- LEGUY, Jules, Thomas CAUCHY, Marta GLAVATSKIKH, Béatrice DUVAL et Benoit DA MOTA (sept. 2020), « EvoMol : a flexible and interpretable evolutionary algorithm for unbiased de novo molecular generation », in : *Journal of Cheminformatics* 12.1, p. 55.
- LEGUY, Jules, Béatrice DUVAL, Benoit DA MOTA et Thomas CAUCHY (nov. 2021a), « Surrogate-Based Black-Box Optimization Method for Costly Molecular Properties », in : *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, p. 780-785.
- (2021b), « Surrogate-Based Black-Box Optimization Method for Costly Molecular Properties (Poster) », in : *RSC-CICAG Artificial Intelligence in Chemistry. Diss, UK (Virtual)*.
- LEGUY, Jules, Bryan GARREAU, Thomas CAUCHY, Benoit DA MOTA et Béatrice DUVAL (jan. 2022b), « Génération d'explications contre-factuelles pour la chimie moléculaire », in : *Atelier EXPLAIN'AI hébergé à EGC 2022*.
- LEGUY, Jules, Marta GLAVATSKIKH, Thomas CAUCHY et Benoit DA MOTA (oct. 2021c), « Scalable estimator of the diversity for de novo molecular generation resulting in a more robust QM dataset (OD9) and a more efficient molecular optimization », in : *Journal of Cheminformatics* 13, p. 76.
- LI, Y., Oriol VINYALS, Chris DYER, Razvan PASCANU et P. BATTAGLIA (2018), « Learning Deep Generative Models of Graphs », in : *ICLR 2018*.
- LIPKUS, Alan H., Qiong YUAN, Karen A. LUCAS, Susan A. FUNK, William F. BARTELT, Roger J. SCHENCK et Anthony J. TRIPPE (juin 2008), « Structural Diversity of Organic Chemistry. A Scaffold Analysis of the CAS Registry », in : *The Journal of Organic Chemistry* 73, p. 4443-4451.
- LIU, Yue, Tianlu ZHAO, Wangwei JU et Siqi SHI (2017), « Materials discovery and design using machine learning », en, in : *Journal of Materiomics*, High-throughput Experimental and Modeling Research toward Advanced Batteries 3.3, p. 159-177.
- MCKAY, M., Richard BECKMAN et William CONOVER (mai 1979), « A Comparison of Three Methods for Selecting Vales of Input Variables in the Analysis of Output From a Computer Code », in : *Technometrics* 21.
- MUSIL, Félix, Michael J. WILLATT, Mikhail A. LANGOVOY et Michele CERIOTTI (fév. 2019), « Fast and Accurate Uncertainty Estimation in Chemical Machine Learning », in : *Journal of Chemical Theory and Computation* 15.2, p. 906-915.

- 
- NIGAM, AkshatKumar, Pascal FRIEDERICH, Mario KRENN et Alan ASPURU-GUZIK (avr. 2020), « Augmenting Genetic Algorithms with Deep Neural Networks for Exploring the Chemical Space », in : *International Conference on Learning Representations*.
- NOÉ, Frank, Alexandre TKATCHENKO, Klaus-Robert MÜLLER et Cecilia CLEMENTI (avr. 2020), « Machine Learning for Molecular Simulation », in : *Annual Review of Physical Chemistry* 71, p. 361-390.
- OLBOYLE, Noel M., Michael BANCK, Craig A. JAMES, Chris MORLEY, Tim VANDERMEERSCH et Geoffrey R. HUTCHISON (2011), « Open Babel : An open chemical toolbox », in : *Journal of Cheminformatics* 3, p. 33.
- OLIVECRONA, Marcus, Thomas BLASCHKE, Ola ENKVIST et Hongming CHEN (sept. 2017), « Molecular de-novo design through deep reinforcement learning », in : *Journal of Cheminformatics* 9.
- PAUL, Debleena, Gaurav SANAP, Snehal SHENOY, Dnyaneshwar KALYANE, Kiran KALIA et Rakesh K. TEKADE (jan. 2021), « Artificial intelligence in drug discovery and development », in : *Drug Discovery Today* 26, p. 80-93.
- PEDREGOSA, Fabian, Gael VAROQUAUX, Alexandre GRAMFORT, Vincent MICHEL, Bertrand THIRION, Olivier GRISEL, Mathieu BLONDEL, Peter PRETTENHOFER, Ron WEISS, Vincent DUBOURG, Jake VANDERPLAS, Alexandre PASSOS et David COURNAPEAU (2011), « Scikit-learn : Machine Learning in Python », in : *Journal of Machine Learning Research* 12, p. 2825-2830.
- POLISHCHUK, P. G., T. I. MADZHIDOV et A. VARNEK (août 2013), « Estimation of the size of drug-like chemical space based on GDB-17 data », in : *Journal of Computer-Aided Molecular Design* 27, p. 675-679.
- POLISHCHUK, Pavel (2020), « CReM : chemically reasonable mutations framework for structure generation », in : *Journal of Cheminformatics* 12.1, p. 28.
- POLYKOVSKIY, Daniil, Alexander ZHEBRAK, Benjamin SANCHEZ-LENGELING, Sergey GOLOVANOV, Oktai TATANOV, Stanislav BELYAEV, Rauf KURBANOV, Aleksey ARTAMONOV, Vladimir ALADINSKIY, Mark VESELOV, Artur KADURIN, Simon JOHANSSON, Hongming CHEN, Sergey NIKOLENKO, Alán ASPURU-GUZIK et Alex ZHAVORONKOV (2020), « Molecular Sets (MOSES) : A Benchmarking Platform for Molecular Generation Models », in : *Frontiers in Pharmacology* 11.
- PREUER, Kristina, Philipp RENZ, Thomas UNTERTHINER, Sepp HOCHREITER et Günter KLAMBAUER (sept. 2018), « Fréchet ChemNet Distance : A Metric for Generative

- 
- Models for Molecules in Drug Discovery », in : *Journal of Chemical Information and Modeling* 58, p. 1736-1741.
- PROBST, Daniel et Jean-Louis REYMOND (déc. 2018), « A probabilistic molecular fingerprint for big data settings », in : *Journal of Cheminformatics* 10.
- QUEIPO, Nestor V., Raphael T. HAFTKA, Wei SHYY, Tushar GOEL, Rajkumar VAIDYANATHAN et P. KEVIN TUCKER (jan. 2005), « Surrogate-based analysis and optimization », in : *Progress in Aerospace Sciences* 41, p. 1-28.
- RAMAKRISHNAN, Raghunathan, Pavlo O. DRAL, Matthias RUPP et O. Anatole von LILIENFELD (août 2014), « Quantum chemistry structures and properties of 134 kilo molecules », in : *Scientific Data* 1.
- RASMUSSEN, Carl Edward et Christopher K. I. WILLIAMS (2006), *Gaussian processes for machine learning*, Adaptive computation and machine learning, MIT Press.
- REGIS, Rommel G. et Christine A. SHOEMAKER (nov. 2007), « A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions », in : *INFORMS Journal on Computing* 19.4, p. 497-509.
- RIOS, Luis Miguel et Nikolaos V. SAHINIDIS (juill. 2013), « Derivative-free optimization : a review of algorithms and comparison of software implementations », in : *Journal of Global Optimization* 56, p. 1247-1293.
- ROGERS, David et Mathew HAHN (mai 2010), « Extended-Connectivity Fingerprints », in : *Journal of Chemical Information and Modeling* 50, Publisher : American Chemical Society, p. 742-754.
- RUDDIGKEIT, Lars, Ruud van DEURSEN, Lorenz C. BLUM et Jean-Louis REYMOND (nov. 2012), « Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17 », in : *Journal of Chemical Information and Modeling* 52, p. 2864-2875.
- RUPP, Matthias, Alexandre TKATCHENKO, Klaus-Robert MÜLLER et O. Anatole von LILIENFELD (jan. 2012), « Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning », in : *Physical Review Letters* 108.
- SCHMIDT, Jens M. (avr. 2013), « A simple test on 2-vertex- and 2-edge-connectivity », en, in : *Information Processing Letters* 113.7, p. 241-244.
- SCHNEIDER, Gisbert, Markus HARTENFELLER, Michael REUTLINGER, Yusuf TANRIKULU, Ewgenij PROSCHAK et Petra SCHNEIDER (jan. 2009), « Voyages to the (un)known : adaptive design of bioactive compounds », in : *Trends in Biotechnology* 27, p. 18-26.

- 
- SCHÜTT, K. T., H. E. SAUCEDA, P.-J. KINDERMANS, A. TKATCHENKO et K.-R. MÜLLER (juin 2018), « SchNet – A deep learning architecture for molecules and materials », en, in : *The Journal of Chemical Physics* 148.24, p. 241722.
- SEGLER, Marwin H. S., Thierry KOGEJ, Christian TYRCHAN et Mark P. WALLER (jan. 2018), « Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks », in : *ACS Central Science*, p. 120-131.
- SHAHRIARI, Bobak, Kevin SWERSKY, Ziyu WANG, Ryan P. ADAMS et Nando de FREITAS (jan. 2016), « Taking the Human Out of the Loop : A Review of Bayesian Optimization », en, in : *Proceedings of the IEEE* 104.1, p. 148-175.
- SHANNON, C. E. (juill. 1948), « A mathematical theory of communication », in : *The Bell System Technical Journal* 27, p. 379-423.
- SHAPEEV, Alexander, Konstantin GUBAEV, Evgenii TSYMBALOV et Evgeny PODRYABINKIN (2020), « Active Learning and Uncertainty Estimation », en, in : *Machine Learning Meets Quantum Physics*, sous la dir. de Kristof T. SCHÜTT, Stefan CHMIELA, O. Anatole von LILIENFELD, Alexandre TKATCHENKO, Koji TSUDA et Klaus-Robert MÜLLER, Lecture Notes in Physics, Cham : Springer International Publishing, p. 309-329.
- SOUSA, Tiago, João CORREIA, Vítor PEREIRA et Miguel ROCHA (nov. 2021), « Generative Deep Learning for Targeted Compound Design », in : *Journal of Chemical Information and Modeling* 61, p. 5343-5361.
- SUMITA, Masato, Xiufeng YANG, Shinsuke ISHIHARA, Ryo TAMURA et Koji TSUDA (2018), « Hunting for Organic Molecules with Artificial Intelligence : Molecules Optimized for Desired Excitation Energies », in : *ACS Central Science* 4.9, p. 1126-1133.
- SUTTON, Richard S. et Andrew G. BARTO (nov. 2018), *Reinforcement Learning, second edition : An Introduction*, MIT Press.
- TERAYAMA, Kei, Masato SUMITA, Ryo TAMURA, Daniel T. PAYNE, Mandeep K. CHAHAL, Shinsuke ISHIHARA et Koji TSUDA (juin 2020), « Pushing property limits in materials discovery via boundless objective-free exploration », in : *Chemical Science* 11.23, Publisher : The Royal Society of Chemistry, p. 5959-5968.
- TERAYAMA, Kei, Masato SUMITA, Ryo TAMURA et Koji TSUDA (mars 2021), « Black-Box Optimization for Automated Discovery », in : *Accounts of Chemical Research* 54.6, p. 1334-1346.

- 
- TOSCO, Paolo, Nikolaus STIEFL et Gregory LANDRUM (juill. 2014), « Bringing the MMFF force field to the RDKit : implementation and validation », in : *Journal of Cheminformatics* 6, p. 37.
- TSUJIMURA, Y. et M. GEN (avr. 1998), « Entropy-based genetic algorithm for solving TSP », in : *1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No.98EX111)*, t. 2, 285-290 vol.2.
- UENO, Tsuyoshi, Trevor David RHONE, Zhufeng HOU, Teruyasu MIZOGUCHI et Koji TSUDA (juin 2016), « COMBO : An efficient Bayesian optimization library for materials science », in : *Materials Discovery* 4, p. 18-21.
- VAIDYANATHAN, Rajkumar, Kevin TUCKER, Nilay PAPILA et Wei SHYY (2003), « CFD-Based Design Optimization For Single Element Rocket Injector », in : *Aerospace Sciences Meetings*.
- VENEPALLI, Bhaskar Rao. et William C. AGOSTA (avr. 1987), « Fenestranes and the flattening of tetrahedral carbon », in : *Chemical Reviews* 87, Publisher : American Chemical Society, p. 399-410.
- VERMA, Sahil, John DICKERSON et Keegan HINES (oct. 2020), « Counterfactual Explanations for Machine Learning : A Review », in : *arXiv :2010.10596 [cs, stat]*.
- VORŠILÁK, Milan, Michal KOLÁŘ, Ivan ČMELO et Daniel SVOZIL (mai 2020), « SYBA : Bayesian estimation of synthetic accessibility of organic compounds », in : *Journal of Cheminformatics* 12, p. 35.
- VU, Ky Khac, Claudia D'AMBROSIO, Youssef HAMADI et Leo LIBERTI (2017), « Surrogate-based methods for black-box optimization », en, in : *International Transactions in Operational Research* 24.3, p. 393-424.
- WALTERS, Patrick (2018), *PatWalters/rd\_filters*, URL : [https://github.com/PatWalters/rd\\_filters](https://github.com/PatWalters/rd_filters).
- (2020), *PatWalters/sillywalks*, URL : [https://github.com/PatWalters/silly\\_walks](https://github.com/PatWalters/silly_walks).
- WEININGER, David (1988), « SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules », in : *Journal of Chemical Information and Computer Sciences* 28, p. 31-36.
- WEININGER, David, Arthur WEININGER et Joseph L. WEININGER (1989), « SMILES. 2. Algorithm for generation of unique SMILES notation », in : *Journal of Chemical Information and Computer Sciences* 29, p. 97-101.



- 
- WELLAWATTE, Geemi P., Aditi SESHADRI et Andrew D. WHITE (mars 2022), « Model agnostic generation of counterfactual explanations for molecules », in : *Chemical Science* 13, p. 3697-3705.
- WEYRICH, Laura S., Sebastian DUCHENE, Julien SOUBRIER, Luis ARRIOLA, Bastien LLAMAS, James BREEN, Alan G. MORRIS, Kurt W. ALT, David CARAMELLI, Veit DRESELY, Milly FARRELL, Andrew G. FARRER, Michael FRANCKEN, Neville GULLY, Wolfgang HAAK, Karen HARDY, Katerina HARVATI, Petra HELD, Edward C. HOLMES, John KAIDONIS, Carles LALUEZA-FOX, Marco de la RASILLA, Antonio ROSAS, Patrick SEMAL, Arkadiusz SOLTYSIAK, Grant TOWNSEND, Donatella USAI, Joachim WAHL, Daniel H. HUSON, Keith DOBNEY et Alan COOPER (avr. 2017), « Neanderthal behaviour, diet, and disease inferred from ancient DNA in dental calculus », in : *Nature*, p. 357-361.
- WILDMAN, Scott A. et Gordon M. CRIPPEN (sept. 1999), « Prediction of Physicochemical Parameters by Atomic Contributions », in : *Journal of Chemical Information and Computer Sciences* 39.5, p. 868-873.
- WILLIAMS, Ronald J. (mai 1992), « Simple statistical gradient-following algorithms for connectionist reinforcement learning », en, in : *Machine Learning* 8, p. 229-256.
- WINTER, Robin, Floriane MONTANARI, Andreas STEFFEN, Hans BRIEM, Frank NOÉ et Djork-Arné CLEVERT (2019), « Efficient multi-objective molecular optimization in a continuous latent space », in : *Chemical Science* 10.34.
- XIN, Jing-fan, Xiao-ru HAN, Fei-fei HE et Yi-hong DING (avr. 2019), « Global Isomeric Survey of Elusive Cyclopropanetrione : Unknown but Viable Isomers », in : *FRONTIERS IN CHEMISTRY* 7.
- YANG, Xin, Yifei WANG, Ryan BYRNE, Gisbert SCHNEIDER et Shengyong YANG (2019), « Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery », in : *Chemical Reviews* 119.18, p. 10520-10594.
- YANG, Xiufeng, Jinzhe ZHANG, Kazuki YOSHIZOE, Kei TERAYAMA et Koji TSUDA (nov. 2017), « ChemTS : an efficient python library for de novo molecular generation », in : *Science and Technology of Advanced Materials* 18, p. 972-976.
- YOSHIKAWA, Naruki et Geoffrey R. HUTCHISON (août 2019), « Fast, efficient fragment-based coordinate generation for Open Babel », in : *Journal of Cheminformatics* 11.1, p. 49.

- 
- YOSHIKAWA, Naruki, Kei TERAYAMA, Masato SUMITA, Teruki HOMMA, Kenta OONO et Koji TSUDA (nov. 2018), « Population-based De Novo Molecule Generation, Using Grammatical Evolution », in : *Chemistry Letters* 47.11, p. 1431-1434.
- YOU, Jiaxuan, Bowen LIU, Rex YING, Vijay PANDE et Jure LESKOVEC (juin 2018), « Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation », NeurIPS 2018, spotlight presentation.
- YUAN, Qi, Alejandro SANTANA-BONILLA, Martijn A. ZWIJNENBURG et Kim E. JELFS (mars 2020), « Molecular generation targeting desired electronic properties via deep generative models », in : *Nanoscale* 12, p. 6744-6758.
- ZHANG, Chenrui, Xiaoqing LYU, Yifeng HUANG, Zhi TANG et Zhenming LIU (nov. 2019), « Molecular Graph Generation with Deep Reinforced Multitask Network and Adversarial Imitation Learning », in : *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, p. 326-329.
- ZHANG, Gang et Charles B. MUSGRAVE (mars 2007), « Comparison of DFT Methods for Molecular Orbital Eigenvalue Calculations », in : *The Journal of Physical Chemistry A* 111.8, p. 1554-1561.
- ZHOU, Jie, Ganqu CUI, Shengding HU, Zhengyan ZHANG, Cheng YANG, Zhiyuan LIU, Lifeng WANG, Changcheng LI et Maosong SUN (jan. 2020), « Graph neural networks : A review of methods and applications », in : *AI Open* 1, p. 57-81.
- ZHOU, Zhenpeng, Steven KEARNES, Li LI, Richard N. ZARE et Patrick RILEY (juill. 2019), « Optimization of Molecules via Deep Reinforcement Learning », in : *Scientific Reports* 9.1, p. 10752.



# Annexes



# RÉSULTATS SUPPLÉMENTAIRES

---

## A.1 EvoMol : optimisation du benchmark GuacaMol

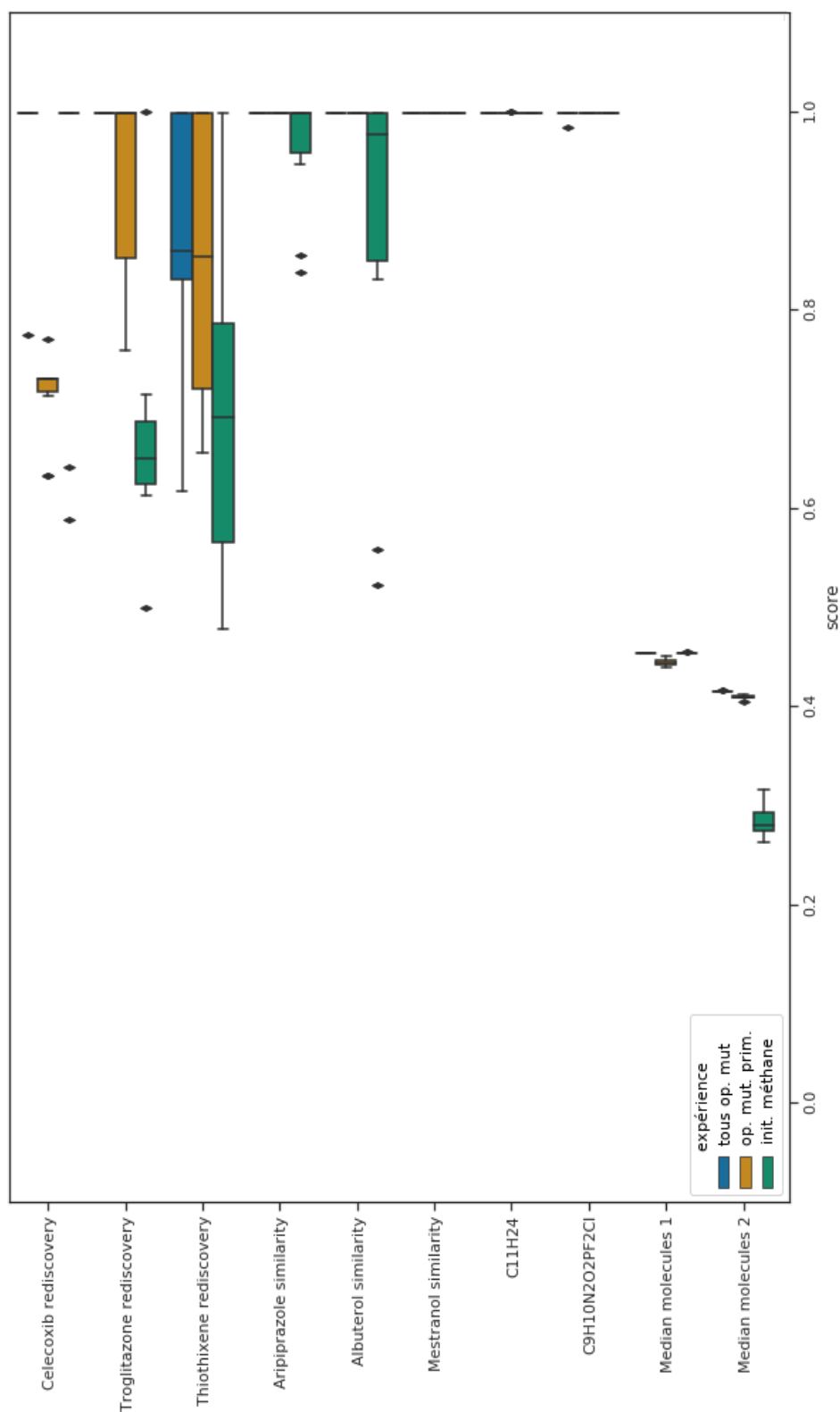


FIGURE A.1 – Diagramme en boîte de la distribution des scores obtenus par EvoMol au *benchmark* GuacaMol, pour les tâches Celecoxib rediscovery à  $C_9H_{10}N_2O_2PF_2Cl$ . Pour chaque tâche, la première ligne correspond à l'expérience de référence utilisant tous les opérateurs de mutation (bleu), la seconde ligne correspond à l'expérience n'utilisant que les opérateurs de mutation primaires (orange), et la troisième ligne correspond à l'expérience utilisant le méthane en tant que population initiale (vert).

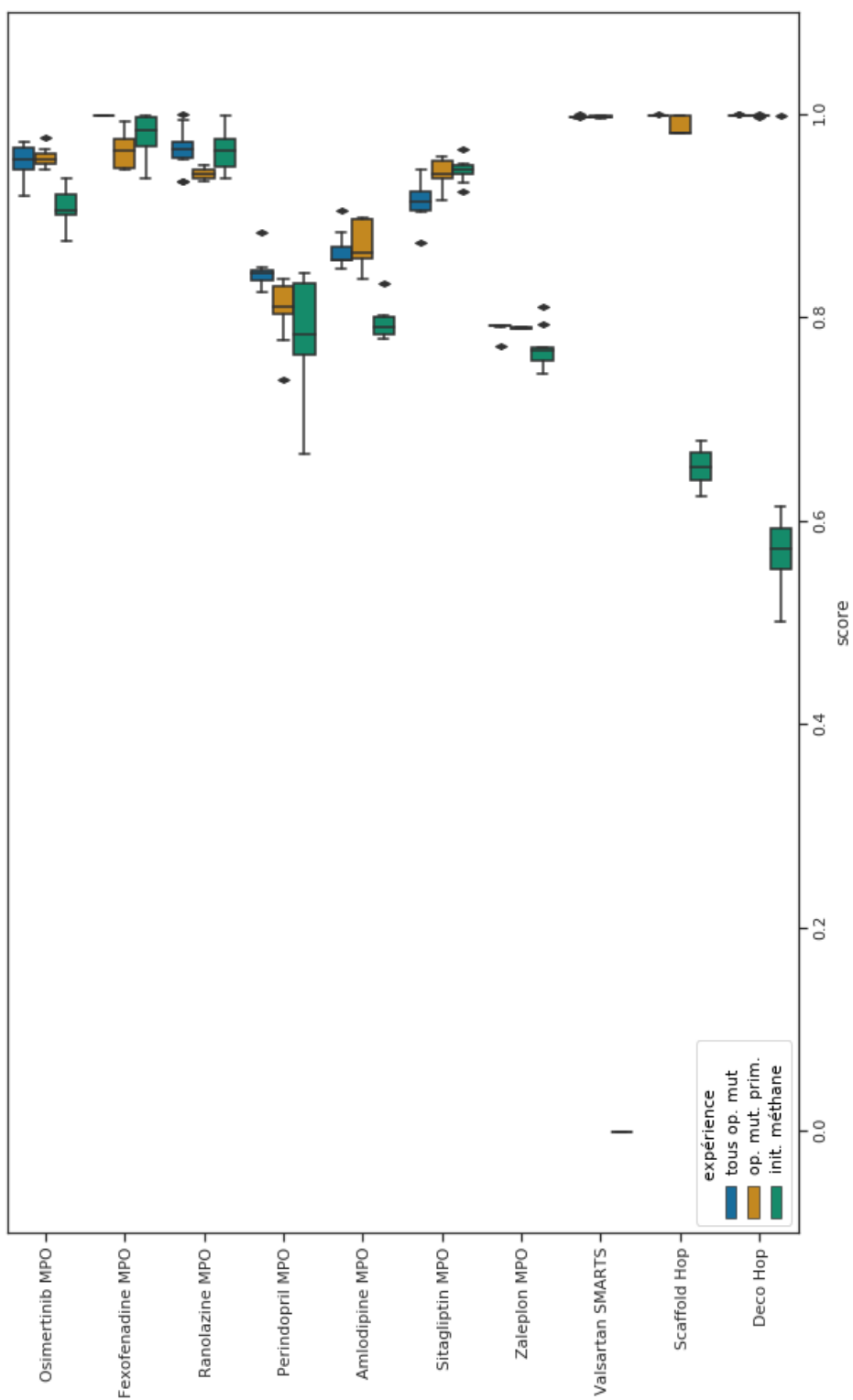


FIGURE A.2 – Diagramme en boîte de la distribution des scores obtenus par EvoMol au *benchmark* GuacaMol, pour les tâches Osimertinib MPO à Deco Hop. Pour chaque tâche, la première ligne correspond à l'expérience de référence utilisant tous les opérateurs de mutation (bleu), la seconde ligne correspond à l'expérience n'utilisant que les opérateurs de mutation primaires (orange), et la troisième ligne correspond à l'expérience utilisant le méthane en tant que population initiale (vert).



---

## A.2 EvoMol : maximisation de l'énergie HOMO



### A.3 BBOMol : optimisation des valeurs de QED

Init.	Expérience	ERT		
		0.9	0.94	0.948
Méthane	BBOMol EI, $k_{\text{RBF}}$ , $\xi = 0.01$	482 (10)	1078 (10)	- (-)
	BBOMol POI, $k_{\text{RBF}}$ , $\xi = 0.01$	958 (9)	1894 (7)	- (-)
	BBOMol EI, $k_{\text{RBF}}$ , $\xi = 0.1$	366 (10)	843 (10)	9425 (2)
	BBOMol POI, $k_{\text{RBF}}$ , $\xi = 0.1$	<b>321</b> (10)	<b>786</b> (10)	4176 (4)
	BBOMol EI, $k_{\text{RBF}}$ , $\xi = 0$	386 (10)	919 (10)	<b>2829</b> (6)
	BBOMol POI, $k_{\text{RBF}}$ , $\xi = 0$	5688 (3)	19646 (1)	- (-)
	EvoMol	18583 (1)	19417 (1)	19625 (1)
QM9	BBOMol EI, $k_{\text{RBF}}$ , $\xi = 0.01$	162 (10)	825 (10)	<b>3495</b> (5)
	BBOMol POI, $k_{\text{RBF}}$ , $\xi = 0.01$	161 (10)	904 (10)	9512 (2)
	BBOMol EI, $k_{\text{RBF}}$ , $\xi = 0.1$	168 (10)	660 (10)	- (-)
	BBOMol POI, $k_{\text{RBF}}$ , $\xi = 0.1$	<b>139</b> (10)	<b>653</b> (10)	5570 (3)
	BBOMol EI, $k_{\text{RBF}}$ , $\xi = 0$	164 (10)	1330 (8)	5795 (3)
	BBOMol POI, $k_{\text{RBF}}$ , $\xi = 0$	193 (10)	2294 (6)	19856 (1)
	EvoMol	527 (9)	3099 (5)	4360 (4)
ChEMBL	BBOMol EI, $k_{\text{RBF}}$ , $\xi = 0.01$	79 (10)	369 (10)	3414 (5)
	BBOMol POI, $k_{\text{RBF}}$ , $\xi = 0.01$	49 (10)	481 (10)	2608 (6)
	BBOMol EI, $k_{\text{RBF}}$ , $\xi = 0.1$	72 (10)	516 (9)	9405 (2)
	BBOMol POI, $k_{\text{RBF}}$ , $\xi = 0.1$	67 (10)	432 (10)	9780 (2)
	BBOMol EI, $k_{\text{RBF}}$ , $\xi = 0$	45 (10)	486 (10)	4070 (4)
	BBOMol POI, $k_{\text{RBF}}$ , $\xi = 0$	<b>37</b> (10)	340 (10)	2515 (6)
	EvoMol	78 (10)	<b>241</b> (10)	<b>592</b> (10)

TABLE A.1 – Étude de l’influence du paramètre d’exploration  $\xi$ . Mesure d’espérance du coût de l’exécution en nombre d’appels à la fonction objectif (ERT) pour obtenir une solution ayant une QED au moins égale aux cibles 0.9, 0.94 et 0.948 pour différents paramétrages de BBOMol et pour EvoMol. Le nombre de fois que la cible a été atteinte parmi les 10 exécutions est indiqué entre parenthèses. Pour chaque stratégie d’initialisation et pour chaque colonne de résultats, la valeur d’ERT la plus faible est mise en évidence en gras. Lorsqu’un tiret (-) est indiqué à la place d’une valeur numérique, cela signifie que la cible n’a pas été atteinte.

Init.	Expérience	ERT		
		0.9	0.94	0.948
Méthane	BBOMol EI, $k_{\text{RBF}}$ , shingles	<b>482</b> (10)	<b>1078</b> (10)	- (-)
	BBOMol EI, $k_{\text{RBF}}$ , aléatoire	- (-)	- (-)	- (-)
	BBOMol POI, $k_{\text{RBF}}$ , shingles	958 (9)	1894 (7)	- (-)
	BBOMol POI, $k_{\text{RBF}}$ , aléatoire	- (-)	- (-)	- (-)
	BBOMol id, $k_{\text{RBF}}$ , shingles	- (-)	- (-)	- (-)
	BBOMol id, $k_{\text{RBF}}$ , aléatoire	- (-)	- (-)	- (-)
	EvoMol	18583 (1)	19417 (1)	<b>19625</b> (1)
QM9	BBOMol EI, $k_{\text{RBF}}$ , shingles	162 (10)	<b>825</b> (10)	<b>3495</b> (5)
	BBOMol EI, $k_{\text{RBF}}$ , aléatoire	- (-)	- (-)	- (-)
	BBOMol POI, $k_{\text{RBF}}$ , shingles	<b>161</b> (10)	905 (10)	9512 (2)
	BBOMol POI, $k_{\text{RBF}}$ , aléatoire	- (-)	- (-)	- (-)
	BBOMol id, $k_{\text{RBF}}$ , shingles	361 (10)	- (-)	- (-)
	BBOMol id, $k_{\text{RBF}}$ , aléatoire	- (-)	- (-)	- (-)
	EvoMol	527 (9)	3099 (5)	4361 (4)
ChEMBL	BBOMol EI, $k_{\text{RBF}}$ , shingles	79 (10)	369 (10)	3414 (5)
	BBOMol EI, $k_{\text{RBF}}$ , aléatoire	227 (10)	3475 (4)	- (-)
	BBOMol POI, $k_{\text{RBF}}$ , shingles	49 (10)	481 (10)	2608 (6)
	BBOMol POI, $k_{\text{RBF}}$ , aléatoire	202 (10)	1899 (6)	- (-)
	BBOMol id, $k_{\text{RBF}}$ , shingles	<b>44</b> (10)	658 (9)	3876 (4)
	BBOMol id, $k_{\text{RBF}}$ , aléatoire	340 (10)	4010 (4)	- (-)
	EvoMol	78 (10)	<b>241</b> (10)	<b>592</b> (10)

TABLE A.2 – Étude de BBOMol lorsque le modèle de substitution est privé de sa capacité d’apprentissage (descripteur moléculaire aléatoire). Mesure d’espérance du coût de l’exécution en nombre d’appels à la fonction objectif (ERT) pour obtenir une solution ayant une QED au moins égale aux cibles 0.9, 0.94 et 0.948 pour différents paramétrages de BBOMol et pour EvoMol. Le nombre de fois que la cible a été atteinte parmi les 10 exécutions est indiqué entre parenthèses. Pour chaque stratégie d’initialisation et pour chaque colonne de résultats, la valeur d’ERT la plus faible est mise en évidence en gras. Lorsqu’un tiret (-) est indiqué à la place d’une valeur numérique, cela signifie que la cible n’a pas été atteinte.



# **ARTICLE : GÉNÉRATION D'EXPLICATIONS CONTRE-FACTUELLES**

---

Reproduction complète de l'article nommé « Génération d'explications contre-factuelles pour la chimie moléculaire », présenté à l'atelier EXPLAIN'AI de la conférence EGC en 2022 [LEGUY et al. 2022b].



# Génération d’explications contre-factuelles pour la chimie moléculaire

Jules Leguy\*, Bryan Garreau\*, Thomas Cauchy\*\*  
Benoit Da Mota\*, Beatrice Duval\*

\*Univ Angers, LERIA, SFR MATHSTIC,  
F-49000 Angers, France  
benoit.damota@univ-angers.fr

\*\*Univ Angers, CNRS, MOLTECH-ANJOU,  
SFR MATRIX, F-49000 Angers, France

**Résumé.** De nombreux modèles d’apprentissage artificiel ont été proposés récemment en chimie moléculaire, afin d’accélérer l’estimation de propriétés moléculaires coûteuses. Le développement de méthodes d’explication de ces modèles est un enjeu important pour leur adoption par la communauté des chimistes. Nous proposons une approche pour générer des explications contre-factuelles pour tout modèle de classification moléculaire binaire. Notre méthode est basée sur une recherche par voisinage par un algorithme évolutionnaire. Nous discutons de l’importance de la mesure de similarité et des opérateurs de voisinage, et nous proposons de restreindre les opérateurs de mutation pour améliorer la pertinence des explications générées.

## 1 Introduction

En chimie moléculaire, de nombreux modèles d’apprentissage artificiel sont étudiés dans l’objectif de favoriser la découverte de molécules prometteuses, dont l’évaluation des propriétés dépend souvent de calculs coûteux (Elton et al., 2019; Von Lilienfeld et al., 2020). Ces travaux sont très prometteurs, mais sont principalement définis comme des boîtes noires. Un enjeu important pour leur adoption par la communauté des chimistes reste la possibilité de les expliquer, afin de permettre une validation expérimentale et d’envisager une interaction entre l’utilisateur final et le modèle.

La génération d’explications contre-factuelles (*counterfactual explanations*) est un thème de recherche porteur pour l’interprétation et la validation de modèles d’apprentissage artificiel pour la classification (Verma et al., 2020). Ces explications permettent à l’utilisateur final du modèle d’étudier les frontières de décision locales sur des exemples concrets dans le domaine d’application. En pratique, cela consiste à rechercher pour une instance donnée un ensemble de transformations minimales résultant en un changement de classe prédite. Pour résoudre ce problème, il est possible de le considérer comme un problème d’optimisation qui consiste à appliquer des perturbations sur l’ins-



tance dans l’objectif de transformer sa classe prédite et de minimiser la distance entre l’instance et son explication contre-factuelle (Wachter et al., 2018).

Des travaux récents envisagent différentes approches pour expliquer les modèles d’apprentissage appliqués à la chimie moléculaire, mais le domaine est encore peu développé (Jiménez-Luna et al., 2020). Une difficulté provient du fait que la représentation des molécules (graphe moléculaire) diffère généralement des descripteurs qui sont utilisés en entrée des modèles d’apprentissage (Von Lilienfeld et al., 2020). Il est possible de convertir la représentation vers le descripteur mais généralement impossible d’effectuer l’opération inverse. Par conséquent, la procédure de génération d’explications doit manipuler différents espaces. Une exception notable est celle des réseaux de neurones pour graphes (*graph neural network*), qui peuvent utiliser en entrée la représentation sous forme de graphe moléculaire.

Nous n’avons connaissance que d’une unique proposition pour la génération d’explications contre-factuelles dans le domaine de la chimie-informatique (Numeroso et Bacchi, 2020). Cette dernière permet d’expliquer un réseau de neurones pour graphes générique à partir d’une méthode d’apprentissage par renforcement.

Nous proposons dans cet article une méthode basée sur un algorithme évolutionnaire pour générer des explications contre-factuelles pour tout type de modèle de classification appliqué à la chimie moléculaire. Nous effectuons une preuve de concept de notre approche pour l’explication de la prédiction de la stabilité moléculaire.

## 2 Méthode

On considère un modèle de classification  $f$  permettant de prédire une propriété moléculaire binaire (par exemple la stabilité ou la toxicité). La prédiction de  $f$  est une valeur réelle comprise entre 0 (classe négative) et 1 (classe positive). Nous considérons ici que la frontière de décision entre les deux classes est située à 0.5. Soit  $y$  une molécule prédite comme étant de classe positive ou négative, dont nous cherchons une explication contre-factuelle  $c_y$  de classe opposée, proche de  $y$  dans l’espace de recherche. Nous proposons d’effectuer cette recherche d’explications par un algorithme évolutionnaire, maximisant la fonction objectif que l’on définit ci-dessous.

### 2.1 Objectif

**Fonction objectif** La fonction objectif que l’on utilise pour guider la procédure d’optimisation vers des solutions répondant au problème de recherche d’explications est définie en équation (1).

$$\max_{c_y} (\text{sim}(y, c_y) + \min(2 \times g(f(c_y)), 1)) \quad (1)$$

Le premier terme est une mesure de similarité dans l’intervalle  $[0, 1]$  visant à orienter la recherche vers des solutions à proximité de  $y$ . Le second terme utilise la valeur réelle prédite par  $f$  afin de guider la recherche vers des solutions de classe désirée. Nous utilisons un seuil de sorte que ce terme soit identique pour toute solution prédite de classe désirée. Dans le cas contraire, la recherche risque d’être orientée vers des

solutions distantes de la frontière de décision, rentrant en contradiction avec l’objectif de similarité (Mothilal et al., 2020).  $g$  est une fonction inversant le sens de la prédiction en fonction de la classe cible, de sorte que  $g(x)$  vaut  $x$  si  $c_y$  doit être positif et  $1 - x$  sinon.

**Similarité** Pour évaluer la similarité entre deux solutions, nous utilisons la distance de Tanimoto sur des empreintes moléculaires (*molecular fingerprints*). Il s’agit d’une mesure communément utilisée, y compris pour la conception de fonctions objectif (Brown et al., 2019). Les empreintes moléculaires sont des vecteurs binaires obtenus à partir d’une fonction de hachage qui assigne un bit à des structures locales extraites dynamiquement depuis le graphe moléculaire (Bajusz et al., 2017). Nous utilisons ici des empreintes de type ECFP4, qui sont basées sur des environnements chimiques de rayon 2. La distance de Tanimoto  $d(x, x')$  étant comprise entre 0 et 1, la fonction de similarité est définie comme  $\text{sim}(x, x') = 1 - d(x, x')$

## 2.2 Optimisation évolutionnaire

Afin d’optimiser la fonction objectif, nous utilisons EvoMol, un algorithme évolutionnaire dédié à l’optimisation de propriétés moléculaires (Leguy et al., 2020). EvoMol est adapté à la recherche de solutions contre-factuelles, puisqu’il est conçu sur un ensemble de mutations locales du graphe moléculaire et permet d’intensifier efficacement le voisinage d’une solution. Parmi ces mutations, sont définies par exemple l’ajout, la suppression ou le changement de type d’un atome ou d’une liaison, l’insertion d’un atome dans une chaîne d’atomes ou encore le déplacement d’un groupe d’atomes. Lors de l’optimisation, les mutations sont sélectionnées aléatoirement après un filtrage garantissant la validité des graphes moléculaires résultants. La validité correspond ici à la garantie que les atomes forment un nombre de liaisons qui est compatible avec leur couche électronique de valence. Notons qu’il s’agit d’un concept moins strict que celui de la stabilité moléculaire (une molécule invalide est instable mais une molécule valide peut être stable ou instable). Dans le cadre de cet article, la population de l’algorithme évolutionnaire est initialisée avec un unique individu correspondant à la molécule dont on cherche une explication contre-factuelle.

## 2.3 Descripteur moléculaire

Le graphe moléculaire n’est généralement pas adapté pour être utilisé en entrée des modèles d’apprentissage. Cela est dû au fait qu’il n’est pas invariant à l’ordre des atomes qui le composent, et qu’il s’agit d’une représentation neutre qui ne met pas en évidence des caractéristiques pertinentes pour un problème d’apprentissage donné. Dans le cadre de cet article, nous utilisons un descripteur moléculaire basé sur les *shingles*. Un *shingle* correspond à un sous-graphe moléculaire de rayon  $r$  et permet d’encoder des caractéristiques structurelles locales (Probst et Reymond, 2018). Pour une molécule donnée, la présence ou l’absence de l’ensemble de ces *shingles* est agrégée dans un vecteur binaire qui est utilisé comme entrée des modèles de classification. La taille de ce vecteur est fixée à 1000, ce qui est suffisant pour encoder toutes les molécules du jeu de données utilisé dans le cadre de ce travail.

### 3 Explication de la prédiction de la stabilité moléculaire

En chimie des matériaux moléculaires, l'estimation *in silico* des propriétés électroniques dépend de calculs coûteux en chimie quantique. En fonction des paramètres, le temps d'exécution peut varier de l'ordre de la minute à l'ordre de la journée pour estimer les propriétés d'une seule molécule. Ce coût de calcul est un frein important à l'optimisation de ces propriétés dans l'espace moléculaire. De plus, la convergence de ces calculs n'est pas garantie et ils peuvent donc échouer, y compris après plusieurs heures. Ces échecs sont dus à l'impossibilité d'obtenir une conformation géométrique stable de la molécule.

Par conséquent, un modèle d'apprentissage artificiel prédisant l'instabilité des molécules et appliqué en tant que filtre avant le calcul en chimie quantique pourrait permettre un gain de temps important en limitant le temps dédié au calcul de solutions dont le résultat ne sera pas utilisable. Dans cette section, nous entraînons un tel modèle, puis nous utilisons notre méthodologie afin de fournir des explications sur un ensemble d'instances prédites comme instables.

#### 3.1 Prédiction de la stabilité moléculaire

Afin de prédire la stabilité des molécules, nous utilisons une forêt aléatoire contenant 100 arbres, suivant l'implémentation de Scikit-Learn (Pedregosa et al., 2011). Nous choisissons ce modèle pour sa simplicité et sa généralité. Rappelons que pour la méthode que nous proposons dans cet article, le modèle d'apprentissage est traité comme une boîte noire et tout autre type de modèle pourrait également être employé.

Le modèle d'apprentissage est utilisé en association avec un descripteur binaire représentant la présence des *shingles* de rayon 1. Nous effectuons une validation croisée du modèle avec recherche par quadrillage interne de la profondeur maximale des arbres sur le jeu de données d'entraînement. Le meilleur paramètre (90) est utilisé pour entraîner un modèle sur la totalité du jeu d'entraînement. Ce modèle est évalué sur le jeu de test, et obtient une aire sous la courbe ROC de 0.925. Toutes les expériences sont effectuées sur le jeu de données OD9, qui contient le calcul en chimie quantique de 394 427 molécules, obtenues par maximisation de la diversité moléculaire (Leguy et al., 2021). Les données sont étiquetées pour représenter le succès ou l'échec de chaque calcul.

#### 3.2 Recherche d'explications contre-factuelles

Nous appliquons notre méthodologie pour générer des explications contre-factuelles à un ensemble de solutions instables (classe négative) du jeu de test.

Pour l'ensemble des expériences, des solutions sont obtenues avec succès par la procédure d'optimisation. Une sélection de ces résultats est présentée en Figure 1, qui représente les résultats obtenus pour trois molécules de classe négative (colonne a) dont on cherche des explications contre-factuelles (colonnes b, c et d). Les résultats de la molécule 1 sont convaincants, puisque les solutions contre-factuelles proposées sont très

proches du point de départ (un changement de type d'atome pour 1.b, un changement de type d'atome et l'ajout d'un atome pour 1.c et 1.d). Pour les deux molécules suivantes en revanche, les explications semblent plus éloignées de la molécule de départ, et il est plus difficile de se représenter les transformations qui ont été effectuées.

Il est intéressant de remarquer que la valeur de similarité peut diminuer de moitié même pour des structures très proches (passage de 1.a à 1.b), et que l'optimisation par exploration du voisinage peut mener à des solutions dont la structure semble très différente du point de départ (passage de 3.a à ses trois explications). Cela peut s'expliquer par le fait que la mesure de similarité utilisée est basée sur des environnements chimiques locaux, et ne prend pas en compte la structure globale de la molécule.

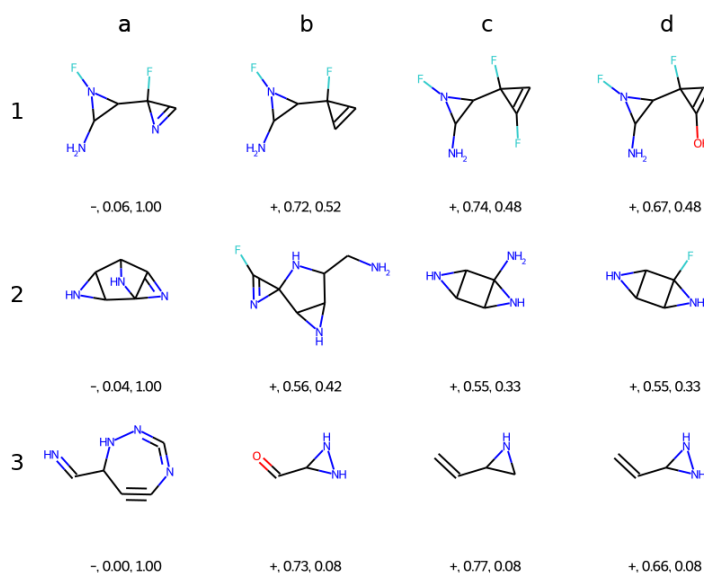


FIG. 1 – Représentation des explications contre-factuelles obtenues en utilisant l'ensemble des opérateurs de mutation. Chaque ligne correspond à une expérience, et est triée par similarité décroissante. La colonne (a) représente les molécules dont on cherche une explication. Chaque molécule est étiquetée selon un triplet (classe, prédiction du modèle de classification, similarité avec la molécule de départ).

### 3.3 Restriction des opérateurs de voisinage

Afin d'obtenir des explications contre-factuelles plus facilement interprétables, nous limitons les opérateurs de mutation pouvant être utilisés lors de l'optimisation évolutionnaire. Ces opérateurs sont restreints au changement du type des atomes, et au

Génération d'explications contre-factuelles pour la chimie moléculaire

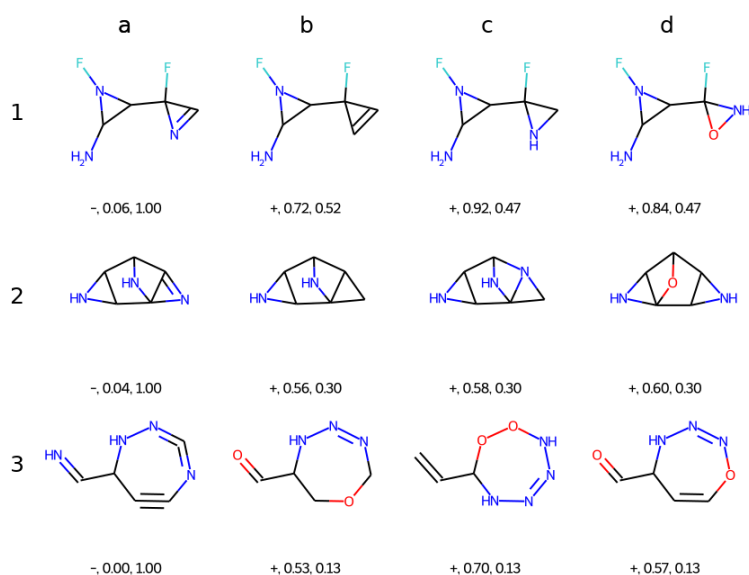


FIG. 2 – Représentation des explications contre-factuelles obtenues en restreignant les opérateurs de mutation. Chaque ligne correspond à une expérience, et est triée par similarité décroissante. La colonne (a) représente les molécules dont on cherche une explication. Chaque molécule est étiquetée selon un triplet (classe, prédiction du modèle de classification, similarité avec la molécule de départ).

changement du type de liaison. Pour ce dernier, seules les mutations qui ne créent ou ne suppriment pas de liaisons sont autorisées. Ainsi, la connectivité du graphe moléculaire ne peut pas être altérée lors de l'optimisation, et les solutions résultantes sont plus facilement comparables à la molécule de départ.

Les résultats de cette nouvelle expérience sont présentés en Figure 2. Les explications des molécules 2 et 3 sont plus pertinentes que dans l'expérience précédente. Il apparaît désormais qu'il suffit de deux actions (changement du type d'atome et de liaison) pour passer de la molécule 2.a à l'explication 2.b. D'un point de vue de chimiste, il semble peu probable que la molécule 3.c soit stable. Elle contient en effet une chaîne de six atomes composés uniquement d'oxygène et d'azote, ce qui ne correspond à aucune fonction chimique connue. Cette expérience permet de mettre en évidence une faille du modèle de classification, qui peut être expliquée par la nature locale du descripteur utilisé.

Il est intéressant de remarquer que les valeurs de similarité des explications 1 et 2 sont plus faibles que dans l'expérience autorisant l'ensemble des mutations. Cela laisse penser que la distance de Tanimoto sur les empreintes moléculaires n'est pas idéale dans

le cadre de ce problème, car elle est basée sur des environnements chimiques locaux mais ne prend pas en compte la structure globale des molécules. Or, pour l'interprétabilité des explications par les chimistes, la structure des solutions est probablement aussi importante que les environnements locaux. Pour les explications de la molécule 3, les valeurs de similarité sont au contraire supérieures à celles de l'expérience précédente. Cela accrédite l'idée que les explications 3.b, 3.c et 3.d précédemment obtenues sont des maximums locaux attractifs de la fonction objectif, mais dont la pertinence est discutable par rapport au problème que l'on souhaite résoudre.

## 4 Perspectives directes

Un ensemble de perspectives peuvent être envisagées afin d'étendre le présent travail. En premier lieu, le voisinage et la mesure de similarité moléculaire semblent être des éléments déterminants pour la génération de bonnes explications contre-factuelles pour la chimie moléculaire. Il serait envisageable d'intégrer la similarité de la structure moléculaire complète (connectivité du graphe moléculaire) au sein de la fonction objectif. Une alternative serait de définir une mesure de distance basée sur le nombre de mutations du graphe moléculaire lors de l'optimisation. Cela permettrait de favoriser les explications proches de la molécule de départ, sans nécessairement restreindre les mutations autorisées.

Afin de rendre l'étude plus robuste, certaines métriques pourraient être étudiées, telle que la diversité des explications générée (Mothilal et al., 2020). De plus, les résultats pourraient être comparés sur les mêmes problèmes de classification moléculaire qu'étudiés par Numeroso et Bacciu (2020).

## 5 Conclusion

Dans cet article, nous présentons une méthode pour la génération d'explications contre-factuelles à tout modèle de classification moléculaire binaire. Les explications sont générées par un algorithme évolutionnaire qui effectue des mutations du graphe moléculaire pour maximiser une fonction objectif qui prend en compte la prédiction du modèle d'apprentissage et la similarité entre la solution proposée et la molécule de départ.

Nous effectuons une preuve de concept pour l'explication d'un modèle de prédiction de la stabilité moléculaire. Notre approche permet d'obtenir facilement des résultats prometteurs. Nous observons qu'à travers la choix des opérateurs de mutation, il est possible de contrôler le type d'explications fournies à l'utilisateur final du modèle.

## Références

- Bajusz, D., A. Rácz, et K. Héberger (2017). 3.14 - Chemical Data Formats, Fingerprints, and Other Molecular Descriptions for Database Analysis and Searching. In S. Chackalamannil, D. Rotella, et S. E. Ward (Eds.), *Comprehensive Medicinal Chemistry III*, pp. 329–378. Oxford : Elsevier.

- Brown, N., M. Fiscato, M. H. Segler, et A. C. Vaucher (2019). GuacaMol : Benchmarking Models for de Novo Molecular Design. *Journal of Chemical Information and Modeling* 59(3), 1096–1108.
- Elton, D. C., Z. Boukouvalas, M. D. Fuge, et P. W. Chung (2019). Deep learning for molecular design—a review of the state of the art. *Mol. Syst. Des. Eng.* 4(4).
- Jiménez-Luna, J., F. Grisoni, et G. Schneider (2020). Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence* 2(10), 573–584.
- Leguy, J., T. Cauchy, M. Glavatskikh, B. Duval, et B. Da Mota (2020). EvoMol : a flexible and interpretable evolutionary algorithm for unbiased de novo molecular generation. *Journal of Cheminformatics* 12(1), 55.
- Leguy, J., M. Glavatskikh, T. Cauchy, et B. Da Mota (2021). Scalable estimator of the diversity for de novo molecular generation resulting in a more robust QM dataset (OD9) and a more efficient molecular optimization. *Journal of Cheminformatics* 13(1), 76.
- Mothilal, R. K., A. Sharma, et C. Tan (2020). Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 607–617. arXiv : 1905.07697.
- Numeroso, D. et D. Bacciu (2020). Explaining Deep Graph Networks with Molecular Counterfactuals. *arXiv :2011.05134 [cs, q-bio]*. arXiv : 2011.05134.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, et D. Cournapeau (2011). Scikit-learn : Machine Learning in Python. *Journal of Machine Learning Research* 12, 2825–2830.
- Probst, D. et J.-L. Reymond (2018). A probabilistic molecular fingerprint for big data settings. *Journal of Cheminformatics* 10(1).
- Verma, S., J. Dickerson, et K. Hines (2020). Counterfactual Explanations for Machine Learning : A Review. *arXiv :2010.10596 [cs, stat]*. arXiv : 2010.10596.
- Von Lilienfeld, O. A., K.-R. Müller, et A. Tkatchenko (2020). Exploring chemical compound space with quantum-based machine learning. *Nature Reviews Chemistry* 4(7).
- Wachter, S., B. Mittelstadt, et C. Russell (2018). Counterfactual Explanations without Opening the Black Box : Automated Decisions and the GDPR. arXiv : 1711.00399.

## Summary

Many machine learning models were recently proposed for molecular chemistry, to quicken the estimation of costly molecular properties. The developpement of explanation methods for these models is a challenge for the their adoption by the community of chemists. We propose an approach to generate counterfactual explanations for any molecular binary classification model, which is based on a neighbourhood search by an evolutionary algorithm. We discuss the significance of the similarity measure and the neighbourhood operators, and we propose to restrict the mutation operators to improve the relevance of the generated explanations.





---

**Titre :** Recherche combinatoire guidée par apprentissage artificiel en chimie moléculaire

**Mot clés :** Optimisation combinatoire, apprentissage artificiel, optimisation moléculaire

**Résumé :** La recherche de molécules satisfaisant des propriétés moléculaires cibles est un enjeu majeur en chimie moléculaire. Dans le cadre de cette thèse, nous cherchons à aborder en particulier des problèmes liés au domaine de la chimie des matériaux moléculaires organiques. Ces problèmes dépendent de fonctions d'évaluation des propriétés cibles dont le calcul est coûteux. Dans nos travaux, nous considérons la recherche de molécules satisfaisant des propriétés cibles comme un problème d'optimisation combinatoire de graphes moléculaires. Nous proposons un algorithme évolutionnaire générique et interprétable pour l'optimisation de propriétés moléculaires, et nous montrons qu'il per-

met d'optimiser avec succès de nombreuses propriétés. Nous proposons une approche par contrainte pour favoriser le réalisme des solutions générées. Nous définissons une méthode d'optimisation efficace pour la maximisation de la diversité moléculaire, basée sur notre algorithme évolutionnaire. Cela nous permet de générer un jeu de molécules très divers. Nous montrons également que l'objectif de diversité peut apporter un gain d'efficacité pour l'optimisation d'une propriété moléculaire cible. Finalement, nous proposons une approche d'optimisation boîte-noire basée sur un modèle de substitution qui est définie pour l'optimisation de propriétés coûteuses, et nous en menons une étude approfondie.

---

**Title:** Combinatorial search lead by machine learning for molecular chemistry

**Keywords:** Combinatorial optimization, machine learning, molecular optimization

**Abstract:** The search for molecules that satisfy target molecular properties is an important issue for molecular chemistry. In this thesis, we aim to tackle in particular problems related to the domain of organic molecular materials. These problems depend on costly evaluation functions. In our work, we consider the search of molecules satisfying molecular properties as a combinatorial optimization problem of molecular graphs. We propose an evolutionary algorithm for the optimization of molecular properties that is designed to be generic and interpretable, and we show that it can successfully optimize numerous target properties. We propose an approach that is

based on binary constraints to favor the generation of realistic molecules. We define an efficient maximization approach of molecular diversity, based on our evolutionary algorithm. It allows for the generation of a dataset of molecules with high molecular diversity, and we also show that the diversity objective can improve the efficiency of the search for the optimization of a target molecular property. Finally, we propose a surrogate-based black-box optimization method that is designed for the optimization of costly molecular properties, and we perform a detailed study of our approach.