

MASTER I INFORMATIQUE  
TRAVAIL ENCADRÉ DE RECHERCHE  
JUIN 2018

---

Rapport QuChemPedia  
Sous titre

---

*Auteur*

Jules LEGUY

*Encadrants*

Benoit DA MOTA  
Thomas CAUCHY

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Contexte et objectifs</b>	<b>5</b>
2.1	Projet QuChemPedia . . . . .	5
2.2	Enjeux en chimie . . . . .	5
2.2.1	Prédiction de propriétés moléculaires . . . . .	5
2.2.2	Optimisation de la géométrie moléculaire . . . . .	6
2.3	Utilisation de modèles d'apprentissage automatique . . . . .	6
2.3.1	Principes fondamentaux . . . . .	6
2.3.1.1	Séparation des jeux de données . . . . .	6
2.3.1.2	Validation croisée . . . . .	6
2.3.1.3	Recherche des paramètres optimaux . . . . .	7
2.3.1.4	Prévention du sur-ajustement . . . . .	7
2.3.2	Entraînement de réseaux de neurones artificiels . . . . .	7
2.3.2.1	Principe . . . . .	7
2.3.2.2	Bibliothèque logicielles . . . . .	7
2.3.2.3	Hyper-paramètres . . . . .	8
<b>3</b>	<b>Représentations géométriques moléculaires</b>	<b>10</b>
3.1	Matrice des coordonnées atomiques . . . . .	10
3.2	Matrice réduite des distances inter-atomiques . . . . .	10
3.2.1	Motivation . . . . .	10
3.2.2	Formalisation . . . . .	11
3.2.3	Reconstruction des molécules . . . . .	12
3.2.3.1	Formalisation de la méthode de reconstruction . . . . .	12
3.2.3.2	Reconstruction automatique des positions en utilisant un solveur . . . . .	14
3.2.3.3	Reconstruction automatique des positions en utilisant des équations de trilatération	15
3.3	Matrice des distances à des points fixes . . . . .	17
3.3.1	Motivation . . . . .	17
3.3.2	Formalisation . . . . .	17
3.3.3	Reconstruction des molécules . . . . .	18
3.4	Représentation locale des liaisons covalentes . . . . .	18
3.4.1	Motivation . . . . .	18
3.4.2	Classes positionnelles . . . . .	18
3.4.3	Distances aux atomes de la liaison . . . . .	19
3.4.4	Restriction au voisinage le plus proche . . . . .	20
<b>4</b>	<b>Données</b>	<b>22</b>
<b>5</b>	<b>Prédiction de longueurs de liaisons convergées</b>	<b>23</b>
5.1	Introduction . . . . .	23
5.1.1	Motivation . . . . .	23
5.1.2	Représentation des données . . . . .	23
5.1.2.1	Données en entrée des modèles . . . . .	23

5.1.2.2	Homogénéisation de la taille des entrées . . . . .	24
5.1.2.3	Représentation d'une liaison en entrée d'un modèle . . . . .	24
5.1.3	Méthodologie . . . . .	24
5.1.3.1	Précision requise . . . . .	24
5.1.3.2	Classes de modèles . . . . .	25
5.1.4	Nomenclature . . . . .	25
5.2	Prédiction de longueurs de liaisons carbone-carbone . . . . .	26
5.2.1	Modèle naïf . . . . .	26
5.2.1.1	Préparation des données et paramètres . . . . .	26
5.2.1.2	Analyse statistique des erreurs . . . . .	27
5.2.1.3	Représentation graphique des résultats . . . . .	27
5.2.2	Restriction au voisinage le plus proche . . . . .	28
5.2.2.1	Préparation des données et paramètres . . . . .	28
5.2.2.2	Analyse statistique des erreurs . . . . .	29
5.2.2.3	Représentation graphique des résultats . . . . .	29
5.2.3	Application de fonctions aux distances . . . . .	29
5.2.3.1	Préparation des données et paramètres . . . . .	29
5.2.3.2	Analyse statistique des erreurs . . . . .	30
5.2.4	Réduction de la largeur du réseau et des entrées . . . . .	31
5.2.5	Recherche par quadrillage des paramètres du modèle naïf . . . . .	32
5.3	Généralisation de la méthode à d'autres liaisons . . . . .	33
5.3.1	Modèles naïfs . . . . .	34
5.3.1.1	Analyse statistique des erreurs . . . . .	34
5.3.1.2	Représentation graphique des résultats . . . . .	35
5.3.2	Restriction au voisinage le plus proche . . . . .	37
5.3.2.1	Analyse statistique des erreurs . . . . .	37
5.3.2.2	Analyse graphique des résultats . . . . .	38
5.3.3	Application de fonctions aux distances . . . . .	38
5.4	Ouverture à d'autres modèles d'apprentissage automatique . . . . .	40
5.4.1	Données d'entrée et complexité algorithmique . . . . .	41
5.4.2	Entraînement de modèles KRR . . . . .	41
5.4.2.1	Recherche par quadrillage des paramètres . . . . .	41
5.4.2.2	Entraînement d'un modèle et analyse des prédictions . . . . .	41
5.4.3	Entraînement de modèles SVM . . . . .	44
5.4.3.1	Recherche par quadrillage des paramètres (non aboutie) . . . . .	44
5.4.3.2	Entraînement d'un modèle et analyse des prédictions . . . . .	44
<b>6</b>	<b>Prédiction de géométries moléculaires convergées</b>	<b>47</b>
6.1	Introduction . . . . .	47
6.1.1	Motivation . . . . .	47
6.1.2	Méthodologie . . . . .	47
6.1.3	Nomenclature . . . . .	48
6.2	Données et paramètres des modèles . . . . .	48
6.2.1	Données . . . . .	48
6.2.1.1	Représentations géométriques . . . . .	48
6.2.1.2	Propriétés atomiques . . . . .	48
6.2.1.3	Bruit . . . . .	49
6.2.1.4	Homogénéisation des tailles de données . . . . .	49
6.2.1.5	Unités . . . . .	50
6.2.1.6	Synthèse du flux de données . . . . .	51
6.2.2	Fonctions d'évaluation . . . . .	51
6.2.2.1	Fonctions de coût . . . . .	51
6.2.2.2	Fonctions de validation . . . . .	52
6.2.2.3	Erreur introduite par le bruit . . . . .	52
6.2.3	Architectures . . . . .	52
6.2.4	Optimisation des paramètres . . . . .	52

<b>6.3 Résultats . . . . .</b>	<b>53</b>
6.3.1 Estimation des performances lors de l'entraînement . . . . .	53
6.3.2 Analyse détaillée d'un modèle . . . . .	53
6.3.2.1 Analyse statistique . . . . .	53
6.3.2.2 Distribution de l'erreur absolue . . . . .	54
6.3.2.3 Distribution de l'erreur absolue en fonction des cibles . . . . .	54
6.3.2.4 Distribution des prédictions en fonctions des cibles . . . . .	55
6.3.3 Abandon de la méthode . . . . .	56
<b>7 Conclusion</b>	<b>58</b>
<b>Appendices</b>	<b>59</b>
<b>A Représentations graphiques des prédictions des modèles <i>DIST_REL_C</i></b>	<b>61</b>
<b>B Représentations graphiques des prédictions des modèles <i>DIST_REL_XY</i></b>	<b>65</b>
<b>C Résultats de la recherche par quadrillage du modèle KRR</b>	<b>79</b>
<b>D Paramètres des modèles <i>DELTA_DIST_+H</i></b>	<b>80</b>

# Chapitre 1

## Introduction

# Chapitre 2

## Contexte et objectifs

### 2.1 Projet QuChemPedia

### 2.2 Enjeux en chimie

#### 2.2.1 Prédiction de propriétés moléculaires

Afin de pouvoir prédire les propriétés d'une molécule, les chimistes ont besoin de connaître avec une grande précision la position de son nuage électronique, défini par des orbitales moléculaires. La connaissance de cette information pour les différents niveaux d'énergie d'une molécule permet notamment de prédire les propriétés d'absorption et d'émission de lumière. Un axe de recherche fort également lié est l'étude de l'absorbance des molécules, permettant de prédire leur potentiel photovoltaïque.

Les orbitales moléculaires sont les solutions d'équations en chimie quantique dont la résolution analytique est impossible avec les outils mathématiques existants. Les chimistes en effectuent donc des approximations à l'aide de fonctions d'ondes, qui sont définies comme la combinaison linéaire de fonctions gaussiennes. Le calcul de ces fonctions d'ondes est par conséquent un enjeu fondamental en chimie, et est malheureusement très coûteux en termes de puissance et de temps de calcul. L'approximation de l'orbitale moléculaire d'une molécule de taille moyenne d'une cinquantaine d'atomes peut en effet prendre plusieurs semaines.

La figure ci-dessous représente les iso-valeurs de la fonction d'onde d'une molécule. Les lobes sont proportionnels à la probabilité de présence des électrons.

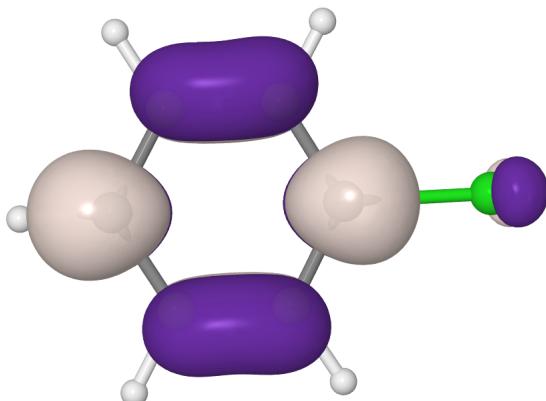


FIGURE 2.1 – Représentation des surfaces d'iso-valeurs de la fonction d'onde du chlorobenzene (Projet QuChemPedia)

## 2.2.2 Optimisation de la géométrie moléculaire

Le calcul des fonctions d'ondes dépend de la géométrie des molécules, qui est donc à la base de l'étude de toutes les propriétés électroniques moléculaires. Afin de prédire correctement ces propriétés, la géométrie doit être connue avec une précision que ne permettent pas la mesure expérimentale ou les résultats théoriques. Une phase d'optimisation géométrique est donc nécessaire.

L'approche communément utilisée en chimie quantique pour optimiser la géométrie moléculaire s'appuie sur l'optimisation itérative de la fonction électronique. Il s'agit d'une descente itérative du gradient énergétique des molécules, pour trouver un ensemble de positions atomiques tel que l'énergie totale est minimale, et donc tel que la molécule est la plus stable. Cette méthode est implémentée dans de nombreux programmes de chimie informatique, dont notamment Gaussian, NWChem et Gamess.

L'inconvénient principal de cette approche est le temps de calcul nécessaire, qui est exponentiellement proportionnel au nombre d'atomes et qui limite donc la possibilité de l'appliquer à des molécules de grandes tailles. Le développement d'une méthode alternative plus rapide et possédant le même niveau de précision ouvrirait donc des perspectives très intéressantes. Il pourrait par exemple permettre la découverte de nouveaux couples de molécules photovoltaïques.

Dans l'objectif de réduire le temps d'optimisation, nous tentons de développer une solution basée sur l'élaboration de modèles d'apprentissage automatique, qui remplaceraient partiellement ou en totalité l'optimisation géométrique quantique.

## 2.3 Utilisation de modèles d'apprentissage automatique

### 2.3.1 Principes fondamentaux

Le *machine learning* (ou apprentissage automatique) définit un certain nombre de méthodes permettant à un programme informatique d'effectuer des prédictions pour résoudre des problèmes. Ces méthodes sont appelées « modèles », et correspondent aux différents algorithmes utilisés pour extraire la connaissance d'un ensemble d'exemples, puis pour effectuer des prédictions sur des exemples inconnus. Avant d'effectuer des prédictions, les modèles doivent en effet d'abord suivre une phase d'apprentissage (ou d'entraînement), pendant laquelle leurs paramètres internes sont ajustés pour effectuer la bonne prédiction en fonction des données en entrée. Dans le cas qui nous intéresse, ces prédictions peuvent être une valeur ou un ensemble de valeurs, on parle alors d'une tâche de régression. Lors de la phase d'entraînement, la distance entre la prédiction d'un modèle et la valeur cible qui était attendue pour chaque exemple lui est donnée. Ce mécanisme lui permet d'ajuster itérativement ses paramètres internes, et l'on parle alors d'un apprentissage supervisé.

#### 2.3.1.1 Séparation des jeux de données

Lorsque l'on élabore un modèle prédictif, il n'est pas souhaitable d'évaluer ses performances sur les données qui ont servi à son entraînement. Cela induirait en effet un biais important, du fait qu'il est possible qu'il devienne très performant pour prédire les données qu'il connaît, mais qu'il ne soit pas capable de généraliser ses connaissances à des données qui lui sont inconnues (REF SURAJUSTEMENT). C'est pour cette raison que l'on sépare généralement les données en deux jeux disjoints : un jeu d'entraînement sur lequel le modèle effectue son apprentissage, et un jeu de test (ou de validation) sur lequel ses performances sont évaluées.

#### 2.3.1.2 Validation croisée

Pour évaluer les performances d'un modèle, il est intéressant d'évaluer ses performances lorsqu'il s'entraîne sur des jeux de données différents, la qualité des prédictions d'un modèle pouvant varier d'un entraînement à un autre. Une technique répandue pour évaluer la variance des performances d'un modèle est celle de la validation croisée à  $n$  entraînements. Elle consiste à séparer le jeu d'entraînement en  $n$  sous-ensembles disjoints nommés plis, puis à effectuer  $n$  entraînements sur toutes les combinaisons telles que  $n - 1$  plis constituent le jeu d'apprentissage, et le dernier pli forme le jeu de test à l'issue de chaque entraînement. On peut alors étudier la performance moyenne du modèle, ainsi que la dispersion de la qualité de ses prédictions.

### 2.3.1.3 Recherche des paramètres optimaux

En plus des paramètres internes, les modèles présentent également un certain nombre de paramètres permettant de régler leurs représentations internes ou leurs phases d'apprentissage. Ces paramètres sont parfois appelés hyper-paramètres. Lorsque l'on élabore un modèle, il faut alors également préciser leurs valeurs.

Pour déterminer un ensemble d'hyper-paramètres optimal ou du moins efficace, la technique la plus répandue est celle de la recherche par quadrillage, éventuellement avec validation croisée. Elle consiste à définir une grille de paramètres, soit un tableau à deux dimensions, contenant pour chaque paramètre (ligne) un ensemble de valeurs (colonnes). Pour chaque combinaison des valeurs, un modèle est entraîné, puis la performance relative des différents modèles est évaluée à la fin de la recherche. Dans le cas d'une recherche par quadrillage avec validation croisée à  $n$  entraînements, chaque combinaison de paramètres est testée  $n$  fois, ce qui permet en outre de tester la variance des performances de chaque combinaison.

### 2.3.1.4 Prévention du sur-ajustement

Lorsque les données représentent des connaissances trop simples pour un modèle, ou réciproquement que les paramètres internes d'un modèle permettent de représenter des connaissances plus complexes que celles des données, la phase d'entraînement du modèle risque de mener à sur-ajustement (ou sur-apprentissage) des données. Cela signifie que le modèle effectuera de bonnes prédictions sur les données d'entraînement, mais que les connaissances extraites par le modèle se généraliseront mal à de nouvelles données, du fait de leur trop grande spécificité aux données d'apprentissage.

Pour parer à cela, les différents modèles proposent des techniques de régularisation qui leur sont propres, et qui vont permettre de limiter leur liberté à ajuster de trop près les données d'entraînement.

## 2.3.2 Entraînement de réseaux de neurones artificiels

### 2.3.2.1 Principe

Les réseaux de neurones artificiels sont des modèles prédictifs qui possèdent l'avantage d'être en général plus efficaces que les autres types de modèles sur des jeux de données de grandes taille. Ils ont également montré de très bonnes performances comparativement aux autres types de modèles sur des problèmes de traitement de signaux et de traitement d'images.

L'objectif de ces réseaux de neurones est d'approximer la fonction qui à chaque exemple donné en entrée associe l'ensemble de valeurs attendu en sortie. Ils sont pour cela composés d'un ensemble de neurones artificiels partageant des connexions. L'information circule de l'entrée à la sortie du modèle, en étant transformée successivement par les différents neurones jusqu'à idéalement prendre la valeur attendue en sortie.

Chaque neurone est défini par l'ensemble des neurones dont la sortie constitue son entrée, un ensemble de poids, un seuil, une fonction d'activation, ainsi que par l'ensemble des neurones dont l'entrée contient sa sortie. Nous formalisons le fonctionnement d'un neurone de la façon suivante.

Soit  $n$  la taille de l'entrée d'un neurone,  $x_i$  sa  $i^{\text{ème}}$  entrée,  $w_i$  le poids associé à sa  $i^{\text{ème}}$  entrée ( $i \in \{1, \dots, n\}$ ) ,  $\Theta$  son seuil et  $\varphi$  sa fonction d'activation. La somme de  $\sum_{i=1}^n w_i x_i$  et de  $\Theta$  est transmise à la fonction d'activation, dont la sortie  $O$  constitue la sortie du neurone. La valeur  $O$  est alors propagée aux neurones suivants. Ce mécanisme est schématisé dans la figure suivante.

La phase d'entraînement d'un réseau de neurones est un problème d'optimisation visant à trouver un ensemble de paramètres  $w_i$  et  $\Theta$  pour chaque neurone du réseau, qui minimise une fonction de coût évaluant la qualité des prédictions.

### 2.3.2.2 Bibliothèque logicielles

Pour l'implémentation des réseaux de neurones, nous utilisons la bibliothèque Tensorflow[1] développée par Google, par l'intermédiaire de la surcouche TFLearn<sup>1</sup>, qui offre une interface simplifiée pour créer des réseaux

---

1. <http://tflearn.org>

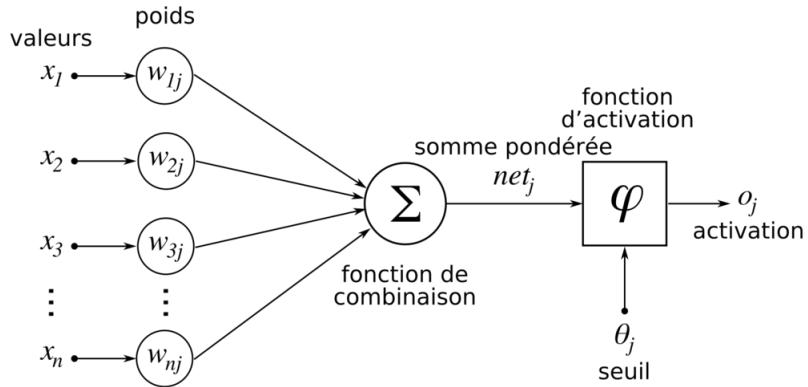


FIGURE 2.2 – Schématisation du fonctionnement d'un neurone artificiel (Wikimedia, Chrislb)

de neurones aux architectures communes.

Nous utilisons en outre l'interface web Tensorboard, incluse dans Tensorflow, qui permet de visualiser l'évolution des différentes métriques évaluant la qualité des modèles au cours de l'entraînement.

### 2.3.2.3 Hyper-paramètres

Les modèles créés par l'intermédiaire de TFLearn présentent un certain nombre d'hyper-paramètres (REF HYPERPARAMS), qui permettent de régler finement les modèles que l'on entraîne. Nous listons ci-dessous ces différents paramètres et leurs rôles.

**Optimiseur :** Définit la méthode d'optimisation des poids utilisée à chaque étape de l'entraînement. Nous utilisons pour tous les modèles l'optimiseur Adam, réputé pour être le plus efficace et éviter les minimums locaux.

**Taux d'apprentissage (learning rate) :** Définit la vitesse maximale à laquelle l'optimiseur va modifier les solutions pendant l'optimisation des poids lors de l'entraînement des modèles. Si la valeur est trop faible, l'entraînement mettra trop de temps à converger vers de bonnes solutions. Si elle est trop élevée, le modèle risque d'être bloqué dans des minimums locaux de la fonction de coût.

**Epsilon :**

**Taille de lot (batch size) :** L'apprentissage des réseaux de neurones artificiels ne s'effectue pas exemple par exemple mais lot d'exemples par lot d'exemples. Ce paramètre définit la taille de ces lots.

**Époques (epochs) :** Définit le nombre d'époques d'entraînement. Cela correspond au nombre de fois que le réseau de neurones va apprendre sur toutes les données du jeu d'entraînement.

**Initialisation des poids :** Les poids du réseau de neurones doivent être initialisés à des valeurs non nulles pour que l'information puisse se propager. Ce paramètre correspond à l'écart-type de la variable aléatoire gaussienne utilisée pour initialiser les poids.

**Fonctions d'activation :** Définit les fonctions d'activation utilisées par les neurones artificiels. On différencie la fonction utilisée par les neurones des couches internes de la fonction utilisée par les neurones de la couche de sortie.

**Dégénération des poids (weight decay) :** Il s'agit d'un paramètre de régularisation ajoutant un terme à la fonction de coût, qui va forcer les poids à ne pas prendre de valeurs trop élevées. La présence de poids possédant des valeurs élevées est en effet un facteur connu de sur-ajustement.

**Taux d'abandon (*dropout*) :** Technique de régularisation permettant d'éviter le sur-ajustement. À chaque époque d'entraînement, certains neurones sont désactivés aléatoirement afin de pousser le modèle à être résilient et à éviter qu'une co-dépendance forte entre certains neurones s'installe. Ce paramètre permet en réalité de définir la proportion de neurones qui resteront activés à chaque époque.

# Chapitre 3

# Représentations géométriques moléculaires

## 3.1 Matrice des coordonnées atomiques

La matrice des coordonnées atomiques est la façon la plus simple de représenter la géométrie d'une molécule. L'intérêt de cette représentation est qu'elle est utilisée par les chimistes (fichiers .mol, .xyz + utilisation dans les logiciels de calcul ?). Il s'agit donc pour nous d'une représentation d'entrée et de sortie. Nos données d'apprentissage contiennent pour chaque molécule une matrice des positions, en plus des numéros et masses atomiques, et nous devons être capables de fournir cette représentation en sortie de nos prédictions, pour que nos résultats soient utilisables par les chimistes.

Formellement, la matrice des coordonnées atomiques d'une molécule contient les coordonnées de chaque atome dans un repère cartésien orthonormé à trois dimensions.

$x_1$	$y_1$	$z_1$
$x_2$	$y_2$	$z_2$
$\vdots$	$\vdots$	$\vdots$
$x_n$	$y_n$	$z_n$

FIGURE 3.1 – Matrice des coordonnées atomiques (molécule de taille  $n$ )

Si cette représentation de la géométrie des molécules est très commode pour les chimistes, elle n'est pas utilisable telle quelle dans nos modèles prédictifs. Nous cherchons en effet à prédire des distances (ou des différences de distances, voir REF DELTA\_DIST...) entre des points. Donner les coordonnées brutes aux modèles implique qu'ils devraient *apprendre* les outils mathématiques permettant de calculer des distances entre des points, ce qui constitue en soi une tâche complexe. C'est pourquoi nous allons définir un ensemble de représentations géométriques, toutes basées sur les distances plutôt que les positions, et adaptées aux différentes prédictions que nous souhaitons effectuer.

## 3.2 Matrice réduite des distances inter-atomiques

### 3.2.1 Motivation

Cette représentation est issue du travail qui a été fait précédemment sur ce projet, et consiste à représenter une molécule par ses distances inter-atomiques. L'intérêt de cette représentation est que les réseaux de neurones qui l'utilisent travaillent dans des repères relatifs. Lorsqu'ils effectuent des prédictions, ils n'ont pas besoin de *comprendre* les notions mathématiques de géométrie permettant de déduire la position d'un point dans un repère à partir de ses distances à d'autres points, contrairement à la représentation décrite en (REF REPR ABS). Cette représentation est donc très commode pour les modèles prédictifs dont l'objectif est de corriger les distances entre deux atomes, puisqu'elle est basée sur les distances entre les paires d'atomes.

De plus, l'utilisation d'une représentation basée sur les distances relatives permet d'offrir une représentation unique pour les molécules ayant des ensembles d'atomes pouvant effectuer des rotations, contrairement aux

représentations basées sur les coordonnées (REF REPR COORDS) ou sur des distances à des points fixes (REF REPR DIST ABS).

Lorsque les modèles utilisent cette représentation en sortie, ou plus précisément que l'on déduit la matrice réduite des distances inter-atomiques de la sortie du modèle (voir REF SORTIE\_DELTA\_DIST+H), nous devons toutefois trouver une méthode (voir REF PRINC RECONSTRUCT) pour reconstruire les molécules sous la forme d'une matrice de coordonnées (REF REPR COORDS).

### 3.2.2 Formalisation

Pour ne pas surcharger les modèles d'information, nous ne travaillons pas sur la matrice de distances inter-atomiques complète, mais sur un sous-ensemble de cardinalité minimale de cette matrice telle que nous pouvons reconstruire sans ambiguïté un ensemble de coordonnées représentant les positions des atomes de la molécule. La matrice des distances étant symétrique et la diagonale étant nulle, toute l'information est contenue dans chaque demi-matrice triangulaire privée de la diagonale.

	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	...	$a_{n-4}$	$a_{n-3}$	$a_{n-2}$	$a_{n-1}$	$a_n$
$a_0$	$d_{0,0}$	$d_{0,1}$	$d_{0,2}$	$d_{0,3}$	$d_{0,4}$	...	$d_{0,n-4}$	$d_{0,n-3}$	$d_{0,n-2}$	$d_{0,n-1}$	$d_{0,n}$
$a_1$	$d_{1,0}$	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$d_{1,4}$	...	$d_{1,n-4}$	$d_{1,n-3}$	$d_{1,n-2}$	$d_{1,n-1}$	$d_{1,n}$
$a_2$	$d_{2,0}$	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$d_{2,4}$	...	$d_{2,n-4}$	$d_{2,n-3}$	$d_{2,n-2}$	$d_{2,n-1}$	$d_{2,n}$
$a_3$	$d_{3,0}$	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$	$d_{3,4}$	...	$d_{3,n-4}$	$d_{3,n-3}$	$d_{3,n-2}$	$d_{3,n-1}$	$d_{3,n}$
$a_4$	$d_{4,0}$	$d_{4,1}$	$d_{4,2}$	$d_{4,3}$	$d_{4,4}$	...	$d_{4,n-4}$	$d_{4,n-3}$	$d_{4,n-2}$	$d_{4,n-1}$	$d_{4,n}$
:	:	:	:	:	:	*	:	:	:	:	:
$a_{n-4}$	$d_{n-4,0}$	$d_{n-4,1}$	$d_{n-4,2}$	$d_{n-4,3}$	$d_{n-4,4}$	...	$d_{n-4,n-4}$	$d_{n-4,n-3}$	$d_{n-4,n-2}$	$d_{n-4,n-1}$	$d_{n-3,n}$
$a_{n-3}$	$d_{n-3,0}$	$d_{n-3,1}$	$d_{n-3,2}$	$d_{n-3,3}$	$d_{n-3,4}$	...	$d_{n-3,n-4}$	$d_{n-3,n-3}$	$d_{n-3,n-2}$	$d_{n-3,n-1}$	$d_{n-3,n}$
$a_{n-2}$	$d_{n-2,0}$	$d_{n-2,1}$	$d_{n-2,2}$	$d_{n-2,3}$	$d_{n-2,4}$	...	$d_{n-2,n-4}$	$d_{n-2,n-3}$	$d_{n-2,n-2}$	$d_{n-2,n-1}$	$d_{n-2,n}$
$a_{n-1}$	$d_{n-1,0}$	$d_{n-1,1}$	$d_{n-1,2}$	$d_{n-1,3}$	$d_{n-1,4}$	...	$d_{n-1,n-4}$	$d_{n-1,n-3}$	$d_{n-1,n-2}$	$d_{n-1,n-1}$	$d_{n-1,n}$
$a_n$	$d_{n,0}$	$d_{n,1}$	$d_{n,2}$	$d_{n,3}$	$d_{n,4}$	...	$d_{n,n-4}$	$d_{n,n-3}$	$d_{n,n-2}$	$d_{n,n-1}$	$d_{n,n}$

FIGURE 3.2 – Matrice complète des distances inter-atomiques d'une molécule

De plus, nous n'avons besoin que des distances à quatre points pour retrouver la position de chaque atome (voir REF RECONSTRUCT), nous nous contentons donc de garder les quatre premières distances de chaque ligne de la matrice triangulaire supérieure privée de la diagonale.

	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	...	$a_{n-4}$	$a_{n-3}$	$a_{n-2}$	$a_{n-1}$	$a_n$
$a_0$	$d_{0,0}$	$d_{0,1}$	$d_{0,2}$	$d_{0,3}$	$d_{0,4}$	...	$d_{0,n-4}$	$d_{0,n-3}$	$d_{0,n-2}$	$d_{0,n-1}$	$d_{0,n}$
$a_1$	$d_{1,0}$	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$d_{1,4}$	...	$d_{1,n-4}$	$d_{1,n-3}$	$d_{1,n-2}$	$d_{1,n-1}$	$d_{1,n}$
$a_2$	$d_{2,0}$	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$d_{2,4}$	...	$d_{2,n-4}$	$d_{2,n-3}$	$d_{2,n-2}$	$d_{2,n-1}$	$d_{2,n}$
$a_3$	$d_{3,0}$	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$	$d_{3,4}$	...	$d_{3,n-4}$	$d_{3,n-3}$	$d_{3,n-2}$	$d_{3,n-1}$	$d_{3,n}$
$a_4$	$d_{4,0}$	$d_{4,1}$	$d_{4,2}$	$d_{4,3}$	$d_{4,4}$	...	$d_{4,n-4}$	$d_{4,n-3}$	$d_{4,n-2}$	$d_{4,n-1}$	$d_{4,n}$
:	:	:	:	:	:	*	:	:	:	:	:
$a_{n-4}$	$d_{n-4,0}$	$d_{n-4,1}$	$d_{n-4,2}$	$d_{n-4,3}$	$d_{n-4,4}$	...	$d_{n-4,n-4}$	$d_{n-4,n-3}$	$d_{n-4,n-2}$	$d_{n-4,n-1}$	$d_{n-4,n}$
$a_{n-3}$	$d_{n-3,0}$	$d_{n-3,1}$	$d_{n-3,2}$	$d_{n-3,3}$	$d_{n-3,4}$	...	$d_{n-3,n-4}$	$d_{n-3,n-3}$	$d_{n-3,n-2}$	$d_{n-3,n-1}$	$d_{n-3,n}$
$a_{n-2}$	$d_{n-2,0}$	$d_{n-2,1}$	$d_{n-2,2}$	$d_{n-2,3}$	$d_{n-2,4}$	...	$d_{n-2,n-4}$	$d_{n-2,n-3}$	$d_{n-2,n-2}$	$d_{n-2,n-1}$	$d_{n-2,n}$
$a_{n-1}$	$d_{n-1,0}$	$d_{n-1,1}$	$d_{n-1,2}$	$d_{n-1,3}$	$d_{n-1,4}$	...	$d_{n-1,n-4}$	$d_{n-1,n-3}$	$d_{n-1,n-2}$	$d_{n-1,n-1}$	$d_{n-1,n}$
$a_n$	$d_{n,0}$	$d_{n,1}$	$d_{n,2}$	$d_{n,3}$	$d_{n,4}$	...	$d_{n,n-4}$	$d_{n,n-3}$	$d_{n,n-2}$	$d_{n,n-1}$	$d_{n,n}$

FIGURE 3.3 – Matrice réduite des distances inter-atomiques d'une molécule (en gras)

$d_{0,1}$	$d_{0,2}$	$d_{0,3}$	$d_{0,4}$
$d_{1,2}$	$d_{1,3}$	$d_{1,4}$	$d_{1,5}$
$d_{2,3}$	$d_{2,4}$	$d_{2,5}$	$d_{2,6}$
$d_{3,4}$	$d_{3,5}$	$d_{3,6}$	$d_{3,7}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$d_{n-4,n-3}$	$d_{n-4,n-2}$	$d_{n-4,n-1}$	$d_{n-4,n}$
$d_{n-3,n-2}$	$d_{n-3,n-1}$	$d_{n-3,n}$	0
$d_{n-2,n-1}$	$d_{n-2,n}$	0	0
$d_{n-1,n}$	0	0	0

FIGURE 3.4 – Matrice réduite des distances inter-atomiques d'une molécule

### 3.2.3 Reconstruction des molécules

Lorsqu'un modèle a pour sortie une matrice réduite des distances inter-atomiques lorsqu'il effectue des prédictions, il faut définir une méthode pour reconstruire une matrice des coordonnées (ref REPR MAT COORDS) de façon automatique à partir de cette sortie, la seule contrainte étant que la distance relative entre chaque paire d'atomes soit respectée. Il ne s'agit pour autant pas d'une tâche triviale, elle s'est en effet avérée impossible en pratique pour les grosses molécules à cause de la propagation des erreurs qu'elle induit (voir REF PROPAG ERREURS).

#### 3.2.3.1 Formalisation de la méthode de reconstruction

**Nécessité et limite de l'introduction d'un atome fictif** Notre méthode de reconstruction des atomes doit permettre de respecter la chiralité<sup>1</sup> des molécules. Or, en déduisant uniquement la position d'un atome de ses distances aux quatre atomes précédents, il existe des cas pour lesquels il existe plusieurs solutions pour la position de l'atome (deux si les quatre atomes précédents sur un même plan ou une infinité si les quatre atomes précédents appartiennent à une droite). Pour pallier ce problème, la méthode retenue précédemment a été d'introduire un nouveau point (que l'on nomme atome fictif) et que l'on place arbitrairement dans la molécule, à une position telle qu'il n'appartient pas au plan formé par les trois premiers atomes, ou à la droite formée par les trois premiers atomes s'ils sont alignés. De cette façon, les atomes suivants seront placés sans ambiguïté.

Cependant, on peut imaginer des cas pour lesquels la technique de l'introduction d'un atome fictif ne permet pas de lever l'ambiguïté, notamment pour les molécules possédant une chaîne de carbones liés par des doubles liaisons (et formant donc une droite). La méthode ne permettra pas dans ce cas de déterminer les positions des atomes en bout de chaîne tel que leurs distances relatives soient respectées, cette information étant perdue lors de la création de la matrice réduite des distances inter-atomiques.

Cette représentation n'est donc pas viable en pratique. Cela fait partie des raisons (voir également REF PB SQRT) pour lesquelles nous sommes passés à la représentation par matrice réduite des distances à des points fixes (REF MATR RED FIXES).

**Placement de l'atome fictif** Puisque l'on définit la position de chaque atome en fonction de ses distances aux quatre atomes précédents, on doit d'abord placer les quatre premiers atomes de façon partiellement arbitraire. Le premier atome de la molécule dans notre représentation étant l'atome fictif  $a_0$ , nous commençons par le placer à la position qui lui a été attribuée.

**Placement de l'atome  $a_1$**  Une fois l'atome  $a_0$  placé, il existe une infinité de solutions pour la position de l'atome  $a_1$ . On peut en effet le placer à tout point appartenant à la surface de la sphère de centre  $a_0$  et de rayon  $d_{0,1}$ .

**Placement de l'atome  $a_2$**  L'atome  $a_2$  appartient au cercle solution de l'intersection entre les sphères de centres  $a_0$  et  $a_1$  et de rayons  $d_{0,2}$  et  $d_{1,2}$ . On choisit donc arbitrairement une position appartenant à ce cercle.

1. Un composé chimique est dit chiral s'il n'est pas superposable à son image dans un miroir. ([https://fr.wikipedia.org/wiki/Chiralite\\_\(chimie\)](https://fr.wikipedia.org/wiki/Chiralite_(chimie)))

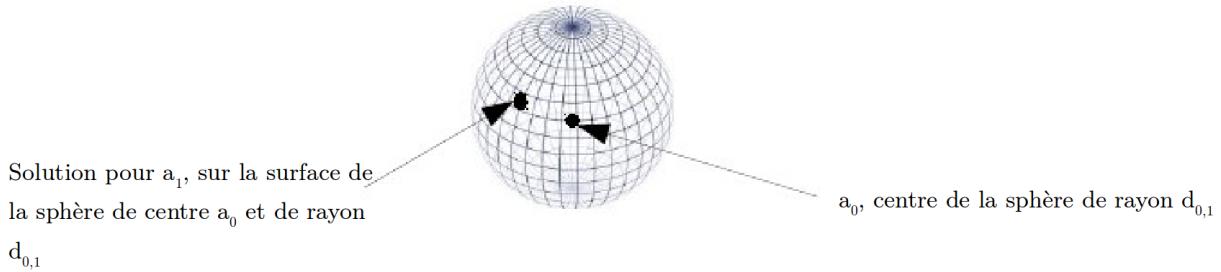


FIGURE 3.5 – Placement de l'atome  $a_1$  (image extraite du rapport de N.Roux)

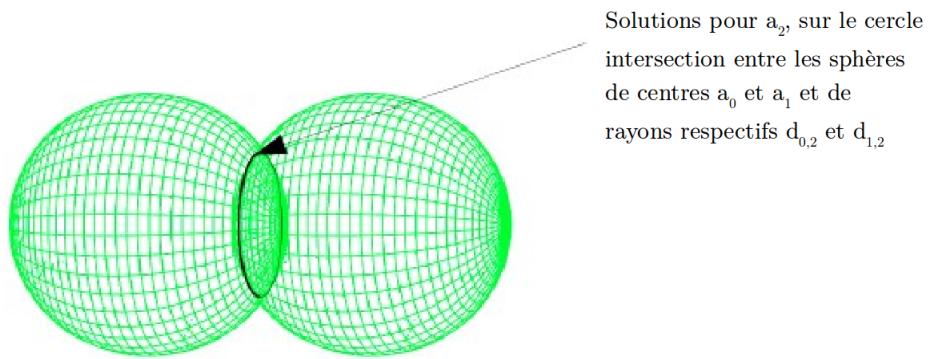


FIGURE 3.6 – Placement de l'atome  $a_2$  (image extraite du rapport de N.Roux)

**Placement de l'atome  $a_3$**  Dans le cas général, il existe deux solutions pour le placement de l'atome  $a_3$ , l'intersection non nulle de trois sphères étant deux points si tous les points ne sont pas sur un même plan ou une même droite. On choisit arbitrairement un point parmi ces deux solutions, car il n'y a pas à ce stade d'ambiguïté de chiralité de la molécule, une molécule composée de trois atomes ne possédant pas de chiralité (l'atome fictif  $a_0$  ne fait pas partie de la molécule).

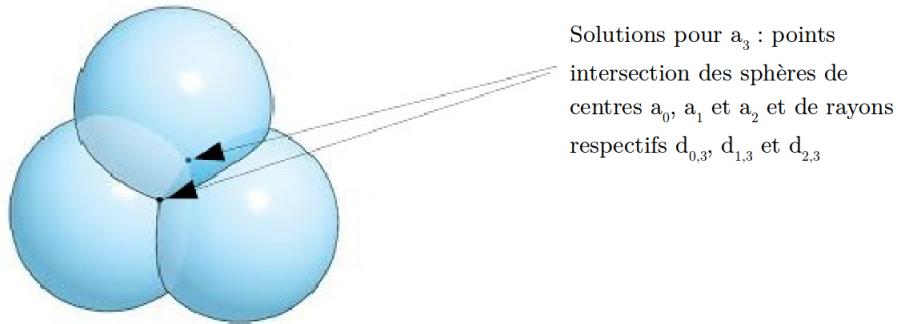


FIGURE 3.7 – Placement de l'atome  $a_3$  (image extraite du rapport de N.Roux)

**Placement de l'atome  $a_n$**  Pour placer l'atome  $a_n$  ( $n$  étant strictement inférieur à la taille de la molécule), nous généralisons la méthode de placement de l'atome  $a_3$ . Plutôt que de travailler sur l'intersection de quatre sphères, nous travaillons toujours sur l'intersection de trois sphères et nous utilisons la dernière distance pour discriminer les deux solutions obtenues. Cela facilite grandement la résolution des équations mathématiques associées et permet d'obtenir des solutions sensiblement équivalentes.

Formellement, nous calculons les positions des deux points solutions de l'intersection des trois sphères de centres  $a_{n-4}$ ,  $a_{n-3}$ , et  $a_{n-2}$  et de rayons  $d_{n-4,n}$ ,  $d_{n-3,n}$  et  $d_{n-2,n}$ , et nous discriminons les deux solutions selon la distance

$d_{n-1,n}$ .

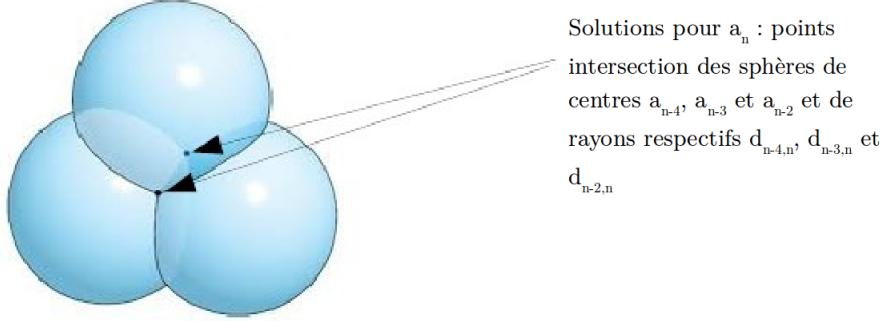


FIGURE 3.8 – Placement de l’atome  $a_n$  (image extraite du rapport de N.Roux)

### 3.2.3.2 Reconstruction automatique des positions en utilisant un solveur

Nous développons ici une méthode permettant de déterminer les coordonnées d’un atome quelconque en utilisant un solveur d’équations non linéaires<sup>2</sup>. Nous utilisons la bibliothèque Sympy<sup>3</sup>.

Tout d’abord, l’atome fictif  $a_0$  doit être placé à la position qui lui a été attribuée (REF PLACEMENT AT FICTIF). Nous plaçons ensuite arbitrairement les trois atomes suivants, de sorte que leurs distances relatives soient respectées. Pour simplifier le problème, nous effectuons une translation temporaire telle que  $a_0'$  est à l’origine du repère. Nous plaçons alors  $a_1'$  sur l’axe  $x$ , à une distance  $d_{0,1}$  de l’origine, et  $a_2'$  sur le plan tel que  $z = 0$ , à une position telle que les distances  $d_{0,2}$  et  $d_{1,2}$  sont respectées. Pour finir, nous plaçons  $a_3'$  à l’une des deux solutions de l’intersection des sphères associées au problème (REF PLACEMENT A3). Le choix de la solution est arbitraire car la reconstruction de la bonne chiralité de la molécule ne dépend pas du placement des trois premiers atomes non fictifs.

$$a'_0 \left\{ \begin{array}{l} x'_0 = 0 \\ y'_0 = 0 \\ z'_0 = 0 \end{array} \right. \quad a'_1 \left\{ \begin{array}{l} x'_1 = d_{0,1} \\ y'_1 = 0 \\ z'_1 = 0 \end{array} \right. \quad a'_2 \left\{ \begin{array}{l} x'_2 = \frac{d_{0,2}^2 - d_{1,2}^2 + x_1^2}{2x_1'} \\ y'_2 = \sqrt{d_{2,0}^2 - x_2'^2} \\ z'_2 = 0 \end{array} \right. \quad a'_3 \left\{ \begin{array}{l} x'_3 = \frac{d_{0,3}^2 + x_1'^2 - d_{1,3}^2}{2x_1'} \\ y'_3 = \frac{-2x'_3 x'_2 + d_{0,2}^2 - d_{2,3}^2 + d_{0,3}^2}{2y'_2} \\ z'_3 = \sqrt{-x_3'^2 - y_3'^2 + d_{0,3}^2} \end{array} \right.$$

FIGURE 3.9 – Placement des atomes  $a_0'$ ,  $a_1'$ ,  $a_2'$  et  $a_3'$

Une fois que les quatre premiers atomes sont placés, nous leur appliquons une translation selon le vecteur  $\vec{a}_0$ , de sorte que l’atome fictif soit à sa position originale, et que les distances relatives des atomes  $a_0$ ,  $a_1$ ,  $a_2$  et  $a_3$  soient toujours consistantes. Nous faisons alors appel au solveur pour résoudre les équations associées au placement des autres atomes de la molécule. Pour chaque atome, nous sélectionnons la solution respectant au mieux la distance  $d_{n-1,n}$  (REF PLACEMENT AN).

**Limites de l’approche par solveur** L’utilisation d’un solveur calculant les solutions au cas par cas pose deux problèmes importants. Le premier concerne les performances de la solution. En effet, la résolution des systèmes d’équations consomme beaucoup de ressources et prend donc un temps non négligeable si on souhaite appliquer la méthode à un grand nombre de molécules.

Le second problème est lié à la propagation des erreurs lors de la reconstruction (REF RECONSTRUCT TRI-LAT). À cause du manque de précision de certaines valeurs, certaines intersections de sphères sont vides. Le

2. [https://en.wikipedia.org/wiki/Nonlinear\\_system](https://en.wikipedia.org/wiki/Nonlinear_system)

3. <http://www.sympy.org/fr/>

$$\begin{cases} d_{n-4,n}^2 = (x_n - x_{n-4})^2 + (y_n - y_{n-4})^2 + (z_n - z_{n-4})^2 \\ d_{n-3,n}^2 = (x_n - x_{n-3})^2 + (y_n - y_{n-3})^2 + (z_n - z_{n-3})^2 \\ d_{n-2,n}^2 = (x_n - x_{n-2})^2 + (y_n - y_{n-2})^2 + (z_n - z_{n-2})^2 \end{cases}$$

FIGURE 3.10 – Équations de sphères permettant d’obtenir la position d’un atome quelconque de la molécule

solveur renvoie alors des solutions imaginaires que nous ne pouvons pas interpréter. Ce problème se manifeste avant tout sur les molécules de taille importante, mais il est impossible de déterminer une taille limite au delà de laquelle nous ne pouvons pas reconstruire les molécules. Cela implique qu’il existe des molécules que nous ne pouvons pas reconstruire, et que nous ne pouvons pas déterminer à l’avance si une molécule donnée peut être reconstruite.

### 3.2.3.3 Reconstruction automatique des positions en utilisant des équations de trilateration

Afin de pallier les problèmes liés à l’utilisation d’un solveur pour construire l’ensemble des positions des atomes d’une molécule à partir de la matrice réduite des distances inter-atomiques, nous utilisons une méthode permettant de calculer les positions de chaque point à partir d’un ensemble d’équations. Cette méthode est décrite sur Wikipédia<sup>4</sup>. Il s’agit d’une méthode de trilateration de points, c’est à dire que l’on cherche à déterminer la position d’un point en fonction de ses distances à trois points dont les positions sont connues, par opposition à la triangulation<sup>5</sup> pour laquelle on détermine la position d’un point en fonction de ses angles à des points dont les positions sont connues.

De même que pour la méthode utilisant un solveur (REF SOLV), nous commençons par placer l’atome fictif  $a_0$  à la position qui lui a été attribuée, puis les atomes  $a_1$ ,  $a_2$  et  $a_3$  de façon arbitraire telle que les distances relatives des atomes  $a_i$ ,  $i \in \{0, \dots, 3\}$  soient respectées. Nous utilisons pour cela les équations décrites en (REF FIG PLACEMENT).

Une fois les quatre premiers atomes placés, nous cherchons à placer l’atome  $a_n$  de la molécule en fonction de ses distances aux quatre atomes précédents. Nous calculons les solutions en considérant que  $a_{n-4}'$  est à l’origine du repère, que  $a_{n-3}'$  est sur l’axe  $x$ , et que  $a_{n-2}'$  est sur le plan tel que  $z = 0$ , puis nous effectuons une translation des solutions dans le système de coordonnées original. Pour cela, nous définissons les quantités et vecteurs suivants.

La notation  $\hat{u}$  indique un vecteur  $u$  de norme 1, et nous considérons que  $\overline{a_i}$  représente le vecteur allant de l’origine au point  $a_i$ , dans le but de simplifier l’écriture des équations.

Vecteur unitaire dans la direction de  $a_{n-4}$  à  $a_{n-3}$  :

$$\hat{e}_x = \frac{\overline{a_{n-3}} - \overline{a_{n-4}}}{d_{n-4,n-3}}$$

Ordre de grandeur signé de la composante  $x$  dans le nouveau système de coordonnées du vecteur  $\overline{a_{n-4}a_{n-2}}$  :

$$i = \hat{e}_x \cdot (\overline{a_{n-4}} - \overline{a_{n-2}})$$

Vecteur unitaire dans la direction  $y$  par rapport à  $\hat{e}_x$  :

$$\hat{e}_y = \frac{\overline{a_{n-2}} - \overline{a_{n-4}} - i\hat{e}_x}{\|\overline{a_{n-2}} - \overline{a_{n-4}} - i\hat{e}_x\|}$$

Vecteur unitaire dans la direction  $z$  par rapport à  $\hat{e}_x$  et  $\hat{e}_y$  :

---

4. <https://en.wikipedia.org/wiki/Trilateration>

5. <https://fr.wikipedia.org/wiki/Triangulation>

$$\hat{e}_z = \hat{e}_x \times \hat{e}_y$$

Ordre de grandeur signé de la composante  $y$  dans le nouveau système de coordonnées du vecteur  $\overrightarrow{a_{n-4}a_{n-2}}$  :

$$j = \hat{e}_y \cdot (\overrightarrow{a_{n-4}} - \overrightarrow{a_{n-2}})$$

On calcule alors les deux solutions pour  $a'_n$  selon les équations suivantes.

$$a'_n \left\{ \begin{array}{l} x'_n = \frac{d_{n-4,n}^2 - d_{n-3,n}^2 + d_{n-4,n-3}^2}{2d_{n-4,n-2}} \\ y'_n = \frac{d_{n-4,n}^2 - d_{n-2,n}^2 + i^2 + j^2}{2j} - \frac{i}{j} x'_n \\ z'_n = \pm \sqrt{d_{n-4,n}^2 - x'^2_n - y'^2_n} \end{array} \right.$$

Enfin, nous translatons les deux solutions  $a'_n$  dans le système de coordonnées original selon le vecteur suivant.

$$\bar{p} = \overrightarrow{a_{n-4}} + x'_n \hat{e}_x + y'_n \hat{e}_y + z'_n \hat{e}_z.$$

Nous obtenons alors deux solutions  $a_n$ , et nous sélectionnons celle telle que la distance  $d_{n-1,n}$  est la plus cohérente.

**Performances et limites (propagation des erreurs)** Les équations de trilatération permettent de calculer la matrice des coordonnées de façon très rapide. Néanmoins, de même que la méthode utilisant un solveur d'équations, cette méthode souffre d'un problème de propagation des erreurs intrinsèque à la représentation par matrice réduite des distances inter-atomiques. En effet, lorsque l'on calcule les coordonnées d'un atome à partir de ses distances aux quatre atomes précédents, et que l'on compare ces distances aux distances aux mêmes points de la position nouvellement calculée, on s'aperçoit qu'elles ne sont pas parfaitement identiques. L'erreur est très faible (de l'ordre de  $10^{-25}$  m) et est individuellement très au-delà de la précision requise en chimie quantique (environ  $10^{-15}$  m), mais elle finit par devenir trop importante du fait de sa propagation au fil des calculs, la position de chaque atome étant calculée à partir de ses distances aux quatre atomes précédents.

La présence d'une racine carrée dans les équations (calcul de  $z'_n$ ) accélère la propagation des erreurs. En effet, après quelques itérations et quelques faibles erreurs, les intersections de sphères deviennent vides, ce qui se traduit dans nos équations par le calcul de la racine d'un nombre négatif. Pour parer cela, nous considérons que le contenu de la racine vaut zéro lorsqu'il est négatif, mais cela introduit une erreur importante et augmente donc la fréquence des intersections vides dans le calcul de la position des atomes suivants.

Afin de retarder l'apparition des erreurs dépassant le seuil toléré, nous aurions pu ajuster les valeurs de  $x'_n$  et  $y'_n$  lorsque l'on considère que le contenu de la racine est nul selon l'équation ci-dessous. Toutefois, cela n'aurait pas constitué une solution viable car le problème aurait été simplement déplacé dans le temps, l'erreur se propageant tout de même.

$$x'^2_n = d_{n-4,n}^2 - y'^2_n$$

FIGURE 3.11 – Optimisation de  $x'_n$  et  $y'_n$  lorsque l'on considère que  $z'_n$  est nul

**Test de la reconstruction** Les tests ont montré que l'on pouvait reconstruire les positions des atomes des molécules avec cette méthode de façon fiable pour les molécules de taille inférieure ou égale à 20 atomes. La méthode de test est la suivante. On génère des coordonnées aléatoirement pour 100000 molécules de taille (nombre d'atomes) variable. On utilise la méthode de reconstruction puis on calcule la nouvelle matrice réduite des

distances inter-atomiques sur les nouvelles positions. On fait la différence des deux matrices de distances pour obtenir les erreurs et on considère que la reconstruction a été un succès si aucune composante de la matrice des erreurs n'est supérieure à un seuil. Ce seuil est choisi à  $10^{-15}$  m, car il correspond à la précision au delà de laquelle les chimistes considèrent qu'il ne s'agit plus d'information mais de bruit. Les résultats sont données dans le tableau suivant.

Taille des molécules	10	15	20	25	30	35	40
Molécules mal reconstruites	0	0	0	18	132	468	1752

FIGURE 3.12 – Test de la méthode de reconstruction pour la matrice des distances inter-atomiques

### 3.3 Matrice des distances à des points fixes

#### 3.3.1 Motivation

La matrice des distances à des points fixes a pour objectif de corriger les défauts de la représentation géométrique moléculaire par matrice réduite des distances inter-atomiques (REF MATR RED DIST REL). Cette dernière possédait en effet le défaut majeur de ne pas être systématiquement réversible en matrice des coordonnées atomiques (REF REPR MAT COORDS). Ce défaut était dû à la propagation des erreurs induite par le fait que les positions des atomes étaient calculées à partir du calcul de la position des atomes précédents (REF REPR DIST REL RECONSTRUCT). Pour parer cela, nous définissons une représentation telle que la position de chaque atome est définie à partir de distances à quatre points fixes du repère. Les erreurs, même si elles existent toujours à des valeurs minimes (autour de  $10^{-25}$  m), ne se propagent donc plus lors de la reconstruction des positions des atomes.

Un autre problème résolu par cette nouvelle représentation est qu'il n'existe plus de molécule dont on ne peut pas reconstruire les positions à cause d'une géométrie plane ou linéaire (REF AT FICTIF), le calcul de la position de chaque atome dépendant désormais de la distance à quatre points de l'espace que l'on choisit tels qu'ils n'appartiennent pas à un même plan.

#### 3.3.2 Formalisation

Formellement, la matrice contient donc les distances de chaque atome d'une molécule à quatre points fixes du repère. Nous choisissons arbitrairement comme points l'origine du repère, et le point sur chaque axe de distance 1 à l'origine. Ce choix est justifié par le fait que les points ont une distance à l'origine du même ordre de grandeur que les coordonnées des atomes dans les données ( $10^0$  à  $10^1$ ). Cela permet donc d'avoir suffisamment d'information pour calculer la position des atomes avec une précision suffisante lors de la reconstruction de la matrice des coordonnées atomiques.

$$p_0(0, 0, 0) \quad p_1(1, 0, 0) \quad p_2(0, 1, 0) \quad p_3(0, 0, 1)$$

FIGURE 3.13 – Points fixes

$d_{a_0, p_0}$	$d_{a_0, p_1}$	$d_{a_0, p_2}$	$d_{a_0, p_3}$
$d_{a_1, p_0}$	$d_{a_1, p_1}$	$d_{a_1, p_2}$	$d_{a_1, p_3}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$d_{a_n, p_0}$	$d_{a_n, p_1}$	$d_{a_n, p_2}$	$d_{a_n, p_3}$

FIGURE 3.14 – Matrice des distances à des points fixes (molécule de taille n)

### 3.3.3 Reconstruction des molécules

De même que pour la représentation par matrice réduite des distances inter-atomiques (REF MAT DIST REL), nous devons être capables de passer d'une matrice des distances à des points fixes à une matrice des coordonnées atomiques, afin que les résultats des modèles prédictifs puissent être utilisés par des chimistes.

La méthode de reconstruction des positions atomiques est très similaire pour les deux représentations. Nous utilisons également les équations de trilateration d'un point à partir des distances à trois points dont les positions sont connues, en utilisant la dernière distance comme un moyen de choisir la bonne solution (voir REF RECONSTRUCT MAT DIST REL). Du fait que la position des quatre points de référence soit fixe et qu'ils suivent les contraintes que nous imposons lors de la translation dans un système de coordonnées plus simple, les équations se trouvent néanmoins simplifiées. En effet,  $p_0$  est à l'origine du repère,  $p_1$  est sur l'axe  $x$  et  $p_2$  est sur le plan tel que  $z = 0$ . Pour rappel, nous résolvons le problème de placement de point dans le système de coordonnées simplifié, puis nous effectuons une translation des solutions dans le système de coordonnées original. Or, nos points de référence se trouvent être les vecteurs unitaires dans chaque direction des deux systèmes de coordonnées. Nous obtenons donc directement les solutions dans le système de coordonnées original. La méthode complète est décrite sur Wikipedia<sup>6</sup>. Nous en extrayons les équations suivantes pour le placement général d'un atome d'une molécule.

$$a_n \left\{ \begin{array}{l} x_n = \frac{d_{a_n, p_0}^2 - d_{a_n, p_1}^2 + 1}{2} \\ y_n = \frac{d_{a_n, p_0}^2 - d_{a_n, p_2}^2 + 1}{2} \\ z_n = \pm \sqrt{d_{a_n, p_0}^2 - x_n'^2 - y_n'^2} \end{array} \right.$$

Nous obtenons alors deux solutions  $a_n$ , et nous sélectionnons celle telle que la distance  $d_{a_n, p_3}$  est la plus cohérente.

## 3.4 Représentation locale des liaisons covalentes

### 3.4.1 Motivation

Cette représentation géométrique s'éloigne des représentations précédentes pour plusieurs raisons. Premièrement, elle s'inscrit dans l'idée de formuler des problèmes plus simples (REF MOD DIST REL) suite à l'échec des modèles utilisant les représentations précédentes (REF MOD DELTA DIST). Pour cette raison, nous n'allons plus chercher à représenter des molécules complètes mais uniquement des liaisons covalentes<sup>7</sup> entre des paires d'atomes au sein des molécules. Cette représentation doit contenir des informations permettant aux modèles l'utilisant de prédire la longueur de la liaison représentée, sans bien-sûr l'enregistrer directement.

En second lieu, la contrainte majeure de la nécessité d'être capable de reconstruire la matrice des coordonnées atomiques à l'issue des prédictions des modèles utilisant cette représentation disparaît. En effet, si l'on peut imaginer des représentations similaires (REF REPR ANGLES) et un assemblage de modèles (REF MODULES) qui permettraient de reconstruire la matrice de coordonnées atomiques d'une molécule convergée (REF DEF CONVERG), il s'agit d'objectifs hors de notre portée pour le moment, notre objectif étant dans un premier temps de valider notre capacité à prédire des géométries moléculaires.

### 3.4.2 Classes positionnelles

La longueur d'une liaison covalente entre deux atomes dépend du type des atomes formant la liaison, mais également de l'influence des atomes au voisinage de la liaison, qui dépend de leur position relativement aux atomes de la liaison. C'est pour cette raison que nous formalisons la notion de classe positionnelle qui va représenter de quel « côté » de la liaison chaque atome se trouve. Les atomes peuvent donc être « à gauche », « au centre » ou « à droite » de la liaison.

6. <https://en.wikipedia.org/wiki/Trilateration>

7. Une liaison covalente est une liaison chimique dans laquelle deux atomes se partagent deux électrons (un électron chacun ou deux électrons venant du même atome) d'une de leurs couches externes afin de former un doublet d'électrons liant les deux atomes. (Wikipédia)

Formellement, on compare la position des atomes aux deux plans normaux à la liaison et passant par les atomes de la liaison. Si un atome est entre les deux plans, il est de classe « centre », sinon il est de classe « gauche » ou « droite » en fonction du plan dont il est le plus proche. Puisque l'on se place dans le repère relatif de la liaison et qu'il n'y existe pas de notion absolue de gauche ou de droite, ces deux classes sont interchangeables à condition que les atomes appartenant à une classe soient tous à distance minimale du même plan.

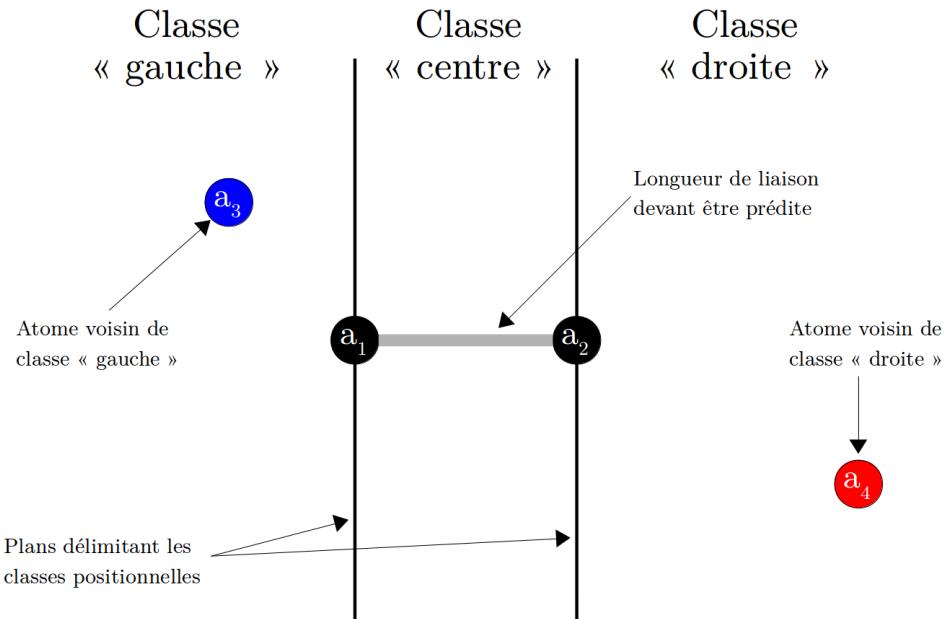


FIGURE 3.15 – Classes positionnelles au voisinage d'une liaison covalente

### 3.4.3 Distances aux atomes de la liaison

L'influence des atomes au voisinage de la liaison dépend également de leur distance à chacun des deux atomes de la liaison. Plus l'atome voisin est près, plus son influence est forte. C'est pourquoi notre représentation contient également cette information.

En fonction des modèles qui l'utilisent, on va éventuellement appliquer une fonction à ces distances, afin de mieux rendre compte de l'influence réelle des atomes au voisinage. Si les réseaux de neurones sont capables d'approximer ces fonctions lors de l'apprentissage, d'autres modèles comme les SVM (REF SVM) ne le sont pas et l'application de ces fonctions est donc nécessaire pour espérer obtenir de bons résultats. Ces fonctions sont les suivantes.

- Fonction identité : distance brute
- Fonction inverse : influence inversement proportionnelle à la distance, relation d'ordre identique à la réalité chimique.
- Fonction inverse du carré : influence inversement proportionnelle au carré de la distance, relation d'ordre identique à la réalité chimique et rend mieux compte de l'influence réelle des atomes qui est liée à la loi de Coulomb<sup>8</sup> en  $\frac{1}{d^2}$ .

Notons qu'aucune de ces fonctions ne représente parfaitement l'influence des atomes en fonction de leur distance, elles permettent cependant de s'approcher de la réalité.

8. [https://fr.wikipedia.org/wiki/Loi\\_de\\_Coulomb\\_\(électrostatique\)](https://fr.wikipedia.org/wiki/Loi_de_Coulomb_(électrostatique))

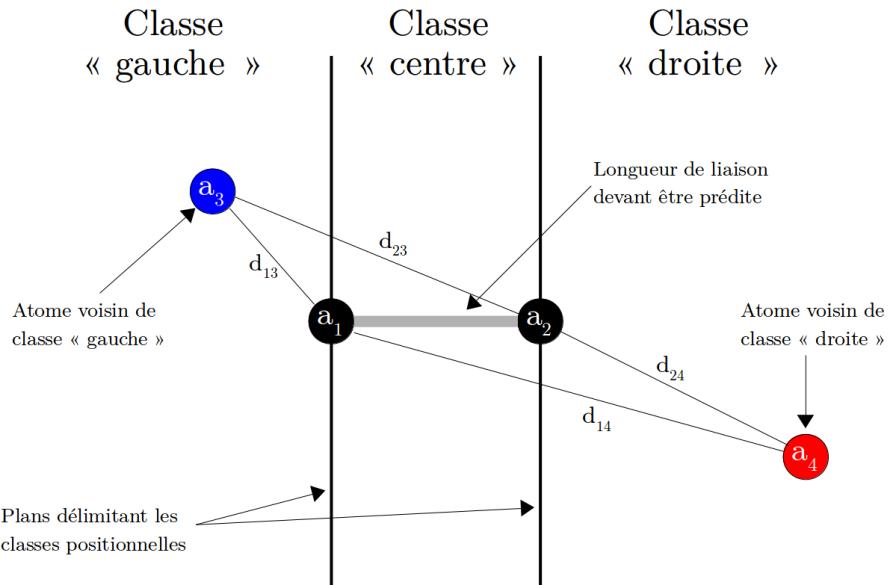


FIGURE 3.16 – Distances des atomes à chacun des atomes de la liaison covalente

### 3.4.4 Restriction au voisinage le plus proche

L'influence des atomes au voisinage de la liaison étant inversement proportionnelle à leur distances aux atomes de la liaison, elle décroît rapidement lorsque l'on s'éloigne de la liaison. L'influence des atomes n'étant pas au voisinage direct est ainsi négligeable. Dans le but de ne pas saturer l'entrée des modèles d'information inutile, nous n'enregistrons alors que les informations (classes positionnelles, distances et autres informations non géométriques spécifiques aux différents modèles) concernant les atomes au voisinage proche de la liaison. Formellement, nous enregistrons ces informations pour les atomes dont la distance à au moins un des atomes de la liaison est inférieure à un seuil  $\epsilon$  donné.

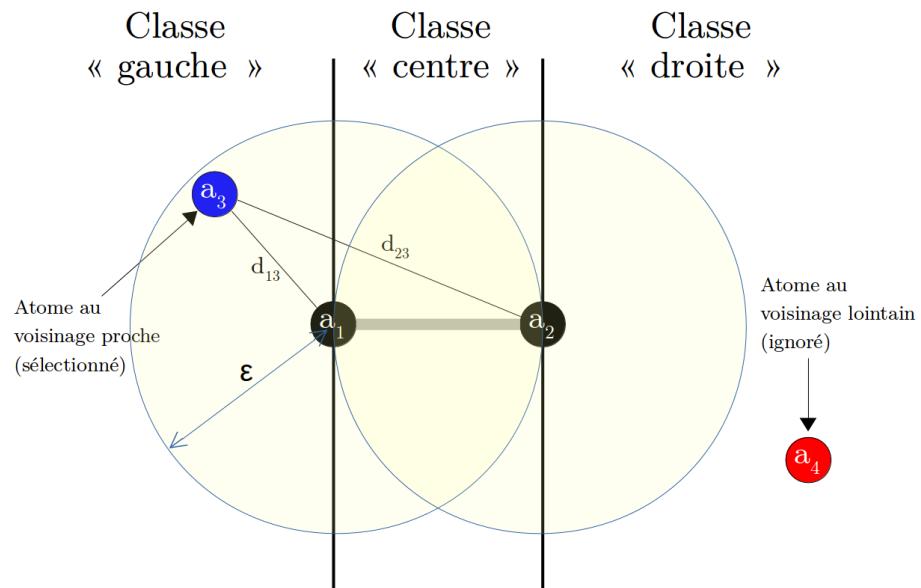


FIGURE 3.17 – Sélection des atomes au voisinage le plus proche

Un autre avantage de cette sélection est qu'il existe des molécules aux géométries particulières (repliées) telles que des atomes au voisinage d'une liaison ont très peu d'influence sur sa longueur (ne forment aucune liaison covalente avec les deux atomes de la liaison), et dont la proximité va induire les modèles en erreur. La sélection des atomes au voisinage le plus proche de la liaison avec un seuil  $\epsilon$  bien choisi va permettre de résoudre ces problèmes.

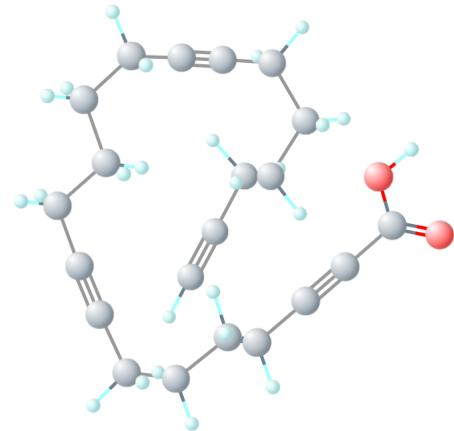


FIGURE 3.18 – Exemple de molécule repliée (CID Pubchem 328310)

## Chapitre 4

### Données

# Chapitre 5

## Prédiction de longueurs de liaisons convergées

### 5.1 Introduction

#### 5.1.1 Motivation

Les modèles décrits dans ce chapitre ont pour objectif de prédire la longueur de liaison optimisée entre des atomes partageant une liaison covalente au sein d'une molécule. L'objectif n'est donc pas de résoudre le problème de prédiction d'une géométrie moléculaire convergée complète, mais plutôt d'en résoudre une version locale simplifiée. Chronologiquement, cette classe de modèles est apparue après l'abandon des modèles tentant de prédire la géométrie optimisée complète d'une molécule (REF ABANDON DELTA\_DIST).

Puisque l'on résout le problème d'optimisation géométrique entre des couples d'atomes, la question de la façon d'utiliser cette méthode pour optimiser la géométrie complète d'une molécule se pose, nous n'y apportons cependant pas de réponse dans ce chapitre. L'objectif de ces modèles est en effet avant tout de valider notre capacité à effectuer des prédictions d'ordre géométrique de précision suffisante sur certains types de liaisons (REF GENERALISATION). L'élaboration d'une méthode d'optimisation géométrique moléculaire complète basée sur la résolution de sous-problèmes locaux est un problème très complexe, qui fait partie des nouveaux objectifs du projet QuChemPedia (REF PERSPECTIVES MODULES).

#### 5.1.2 Représentation des données

##### 5.1.2.1 Données en entrée des modèles

Les modèles décrits dans ce chapitre utilisent en entrée la représentation géométrique locale des liaisons covalentes (REF GEOM LOCALE), qui permet de représenter les atomes au voisinage d'une liaison. En plus des informations géométriques, on représente la masse et le numéro atomique de chaque atome au voisinage de la liaison. Le numéro atomique est encodé en *one-hot encoding*, c'est à dire de façon booléenne. La discréétisation des numéros atomiques a pour but de ne pas instaurer de relation d'ordre entre les différents atomes et donc a priori de mieux guider les modèles lors de l'apprentissage. Elle implique toutefois qu'il faut déterminer une limite aux numéros atomiques des atomes acceptés par un modèle. En effet, cet encodage coûte un attribut pour chaque numéro atomique accepté, pour chaque atome au voisinage de la liaison. Afin de travailler sur des modèles de taille raisonnable, ils acceptent les atomes de numéros atomiques inférieurs ou égaux à celui du fluor, ce qui correspond à neuf attributs encodant le numéro atomique pour chaque atome du voisinage.

De même la classe positionnelle (REF CLASSE POS) de chaque atome par rapport à la liaison est représenté en *one-hot encoding*, afin de ne pas représenter cette information sur un ensemble possédant une relation d'ordre.

Le tableau suivant présente le nombre d'attributs utilisés pour représenter chaque atome au voisinage d'une liaison.

Classe positionnelle	Distances	Masse atomique	Numéro atomique	Total
3	2	1	9	15

FIGURE 5.1 – Quantité d’attributs représentant chaque atome au voisinage d’une liaison

### 5.1.2.2 Homogénéisation de la taille des entrées

Les molécules possédant un nombre variable d’atomes et l’entrée des modèles étant de taille fixe, nous effectuons une procédure de *padding*<sup>1</sup> des données. Cela signifie que l’entrée des modèles est découpée en blocs, représentant chacun un atome au voisinage de la liaison. La taille des blocs dépend des attributs représentant chaque atome, et le nombre de blocs définit le nombre maximal d’atomes au voisinage des liaisons que les modèles peuvent traiter. Nous déduisons cette information de la taille des molécules que l’on choisit d’accepter en entrée des modèles. La grande majorité des molécules étant de taille inférieure à 60 (VOIR DONNEES DISTRIB TAILLES) et les deux atomes composant la liaison n’apparaissant pas dans les entrées, nous choisissons de limiter le voisinage de la liaison à 58 atomes.

La représentation d’une liaison en entrée des modèles est donc composée de 58 blocs de 15 attributs, soit 870 valeurs. Lorsqu’une liaison possède moins de 58 voisins, les blocs correspondant aux atomes non définis valent zéro.

### 5.1.2.3 Représentation d’une liaison en entrée d’un modèle

Nous détaillons la représentation en entrée d’un modèle prédictif de la liaison imaginaire utilisée comme exemple en (REF REPR LOCALE). On considère que l’atome  $a_3$  est un atome d’azote et que l’atome  $a_4$  est un atome d’oxygène.

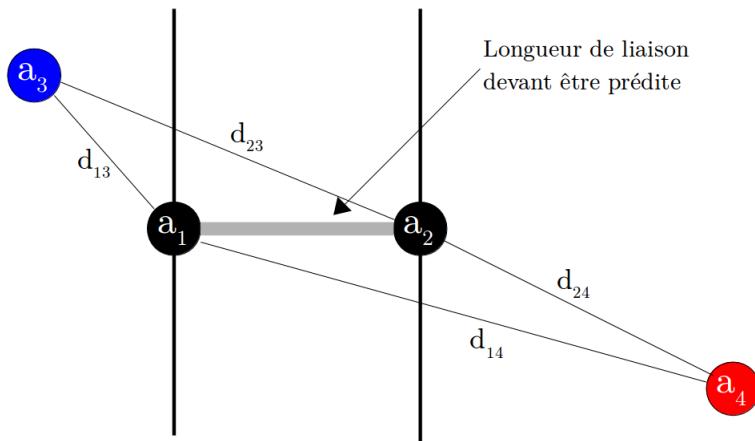


FIGURE 5.2 – Représentation schématique d’une liaison imaginaire

L’entrée correspondant à la liaison représentée ci-dessus est donnée dans le tableau suivant.

### 5.1.3 Méthodologie

#### 5.1.3.1 Précision requise

Les modèles décrits dans ce chapitre travaillent sur des données « parfaites », c’est à dire qu’ils prédisent des longueurs de liaisons dans des molécules dont la géométrie a déjà été optimisée. Cela nous permet de confirmer notre capacité à effectuer des prédictions d’ordre géométrique, mais pas de nous assurer que les modèles pourront effectuer de bonnes prédictions sur des données non optimisées issues de mesures ou de résultats théoriques.

1. Rembourrage

Classe pos. g ?   c ?   d ?			Distances		Masse atomique	Numéro atomique								
H ?	He ?	Li ?	Be ?	B ?	C ?	N ?	O ?	F ?						
1	0	0	$d_{13}$	$d_{23}$	14,007	0	0	0	0	0	1	0	0	
0	0	1	$d_{14}$	$d_{24}$	15,999	0	0	0	0	0	0	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	
:	:	:	:	:	:	:	:	:	:	:	:	:	:	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	

FIGURE 5.3 – Représentation des données d'une liaison en entrée d'un modèle prédictif

L'entraînement de modèles travaillant sur des données imparfaites fera l'objet de la suite du projet QuChemPedia (REF PERSPECTIVES). Pour pouvoir espérer obtenir de bonnes prédictions sur des données non optimisées, il faut obtenir de très bons résultats sur des données optimisées, comme on le montre en REF GENERALISATION.

La précision que l'on peut espérer atteindre avec les données sur lesquelles les modèles s'entraînent (REF PUBCHEM) est de l'ordre du picomètre (pm), soit  $10^{-12}$  m. Cette précision dépend des fonctions choisies lors de l'optimisation géométrique quantique des molécules (REF OPTI DFT). Les modèles effectuant des prédictions dont l'erreur est inférieure à 1 pm confirmeront donc notre capacité à effectuer des prédictions d'ordre géométrique de précision suffisante.

#### 5.1.3.2 Classes de modèles

Nous tentons de prédire les longueurs de liaisons entre plusieurs couples d'atomes, en entraînant un modèle par couple d'atomes formant une liaison. Les liaisons carbone-carbone ne seront alors pas prédites par le même modèle que les liaisons carbone-hydrogène. Cette séparation en sous-problèmes segmentés a pour objectif d'évaluer la précision que peuvent atteindre les modèles sur les problèmes les plus simples que l'on peut leur donner. L'évaluation de leur précision sur des problèmes plus complexes fait partie des futurs objectifs (REF MODULES PLUSIEURS LIAISONS). La prédiction des longueurs de liaisons d'un unique couple donné d'atomes par modèle n'en fait toutefois pas un problème trivial, car elles peuvent sensiblement varier en fonction des atomes impliqués (REF DISTRIB LONGUEURS). Si les liaisons oxygène-hydrogène ont une taille variant en général entre 96 pm et 106 pm soit avec une amplitude de 10 pm, la taille des liaisons carbone-carbone varie entre 120 pm et 160 pm, ce qui représente une amplitude de 40 pm.

L'entraînement des modèles est un processus qui prend un temps non négligeable (REF CONTRAINTES MATERIELLES). Pour cette raison, nous n'entraînons pas tous les modèles sur un grand nombre d'exemples et nous définissons deux classes de modèles ayant des objectifs différents.

La première classe de modèles a un objectif d'expérimentation. Les modèles sont entraînés sur un nombre relativement faible d'exemples différents sur 150 époques (REF DEF EPOCH), ce qui représente environ 2h de préparation de données et 6h d'entraînement avec le matériel disponible. Ces modèles sont entraînés dans le but d'expérimenter de nouveaux traitements des données d'entrée ou de nouveaux paramètres. Ils ont pour objectif de discriminer la qualité de ces entrées et paramètres, c'est pourquoi ils travaillent sur la prédiction difficile des distances de liaisons carbone-carbone. Ces modèles sont décrits en REF PRED C

La seconde classe de modèles a un objectif de validation des paramètres performants issus de l'entraînement des modèles de la première classe, ainsi qu'un objectif de généralisation des méthodes à différents types de liaisons. Ces modèles s'entraînent donc sur plus d'exemples et sur plusieurs liaisons différentes (carbone-carbone, carbone-hydrogène et oxygène-hydrogène). L'entraînement des trois modèles de cette classe pour un ensemble d'entrées et de paramètres donné prend environ deux jours. Ces modèles sont décrits en REF GENERALISATION.

#### 5.1.4 Nomenclature

Afin d'y faire référence simplement, nous nommons les différents modèles que l'on entraîne. Tous les modèles décrits dans ce chapitre ont pour préfixe *DIST\_REL*, issu de leur vocation à prédire la distance relative entre les atomes d'une liaison, et pour suffixe le numéro chronologique de leur entraînement au sein de leur classe. Les modèles de la première classe (resp. seconde) ont pour préfixe *DIST\_REL\_C* (resp. *DIST\_REL\_XY*, où

X et Y désignent les symboles des éléments formant la liaison prédictive). La différence de nomenclature entre les deux classes et notamment entre les modèles *DIST\_REL\_C* et *DIST\_REL\_CC* est discutable, mais a pour avantage de faire apparaître simplement la distinction.

Enfin, les modèles prédictifs n'étant pas des réseaux de neurones artificiels font apparaître leur type dans leur nom.

## 5.2 Prédiction de longueurs de liaisons carbone-carbone

Les modèles décrits dans cette section appartiennent à la première classe (REF CLASSES), et ont pour objectif de prédire les longueurs de liaisons entre des atomes de carbone. Ils sont tous des réseaux de neurones artificiels entraînés avec les paramètres suivants. Le rôle de chaque paramètre est décrit en REF NN PARAMS.

Paramètre	Valeur
Taille de lot ( <i>batch size</i> )	5000
Epsilon (Adam)	0.001
Initialisation des poids ( <i>stddev_init</i> )	0.001
Fonction d'activation couches cachées	elu
Fonction d'activation couche de sortie	linéaire
Abandon ( <i>dropout</i> )	0.98
Dégénération des coefficients ( <i>weight decay</i> )	0.001

FIGURE 5.4 – Paramètres d'entraînement des modèles *DIST\_REL\_C*

Les tailles des jeux de données utilisés par les modèles sont données dans le tableau suivant. Notons que tous les modèles apprennent sur les mêmes molécules et sont testés sur les mêmes molécules, même si la préparation des données diffère.

Jeu	Taille
Entraînement	2770924
Test	554434

FIGURE 5.5 – Taille des jeux de données pour les modèles *DIST\_REL\_C*

### 5.2.1 Modèle naïf

#### 5.2.1.1 Préparation des données et paramètres

Le premier modèle entraîné utilise la représentation locale des liaisons covalentes (REF REPR LOCALE + REF REPR DONNEES) simple, c'est à dire que l'on ne restreint pas la représentation aux atomes au voisinage le plus proche, et que l'on n'applique pas de fonction inversant l'ordre des distances. Les paramètres spécifiques d'entraînement et de préparation des données sont donnés dans le tableau suivant.

Paramètre	Valeur
Taux d'apprentissage ( <i>learning rate</i> )	0.01
Taille de l'entrée	870
Taille de la dernière couche cachée	870
Profondeur	3
Fonction appliquée aux distances	identité
Restriction du voisinage ( $\epsilon$ )	$\infty$

FIGURE 5.6 – Paramètres d'entraînement et de préparation des données spécifiques au modèle *DIST\_REL\_C\_01*

### 5.2.1.2 Analyse statistique des erreurs

Dans la figure suivante, nous présentons les valeurs d'un certain nombre de métriques statistiques permettant d'évaluer les erreurs du modèle. On remarque que les prédictions du modèle sont assez bonnes, la médiane des erreurs étant de l'ordre du demi picomètre. L'erreur moyenne est toutefois au dessus de la barre du picomètre et l'erreur maximale est élevée.

Métrique	Valeur
Moyenne	1,3301
Médiane	0,6881
Écart-type	1,9810
Minimum	0,0000
Maximum	28,7268
Erreur relative moyenne	0,9126%

FIGURE 5.7 – Analyse statistique des erreurs du modèle *DIST\_REL\_C\_01* (en pm)

### 5.2.1.3 Représentation graphique des résultats

La représentation graphique de la distribution des erreurs montre qu'un grand nombre d'erreurs ont des valeurs au delà du seuil du picomètre de précision nécessaire.

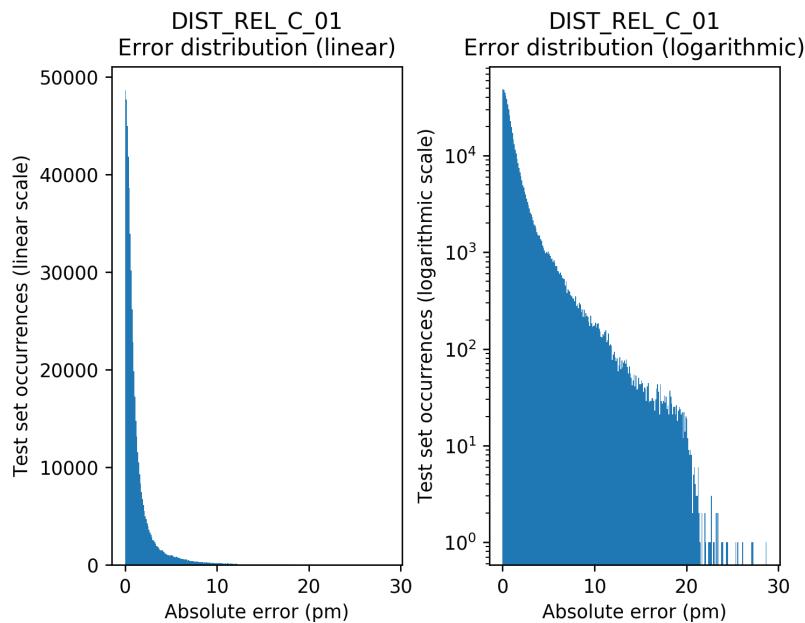


FIGURE 5.8 – Distribution des erreurs du modèle *DIST\_REL\_C\_01*

La représentation de l'erreur relative à la distance devant être prédite, exprimée en fonction des distances cibles montre que les plus grosses erreurs sont effectuées lors de la prédiction des longueurs de liaisons les moins représentées dans les données, mais que les longueurs de liaisons les plus représentées sont en revanche très bien prédites. L'histogramme de la représentation des liaisons situé dans la partie inférieure permet de s'en rendre compte.

Enfin, la représentation des prédictions en fonction des distances cibles permet de préciser quels types d'erreurs sont commises en fonction des différentes classes de liaisons à prédire. Les liaisons doubles et triples sont très largement surestimées, même si une partie non négligeable de ces liaisons est correctement prédite. La grande majorité des liaisons simples et aromatiques sont très bien prédites, même si une partie des liaisons aromatiques est légèrement sur-estimée et une partie des liaisons simples est légèrement sous-estimée.

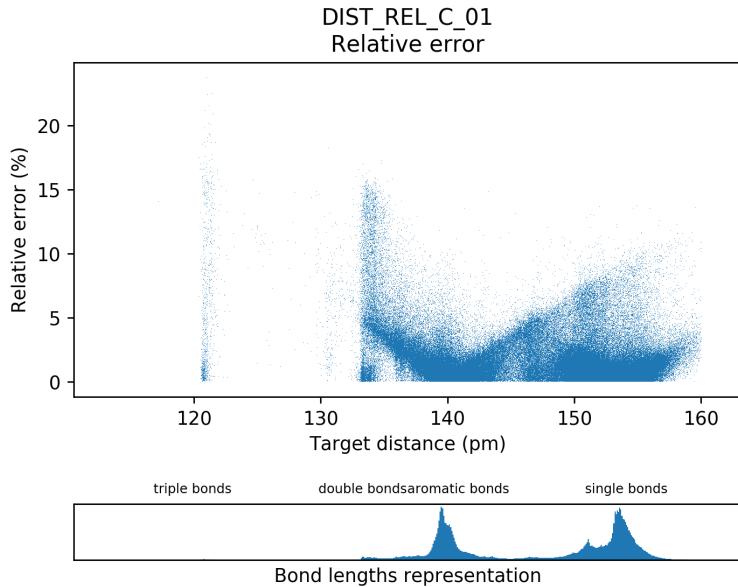


FIGURE 5.9 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_C\_01*

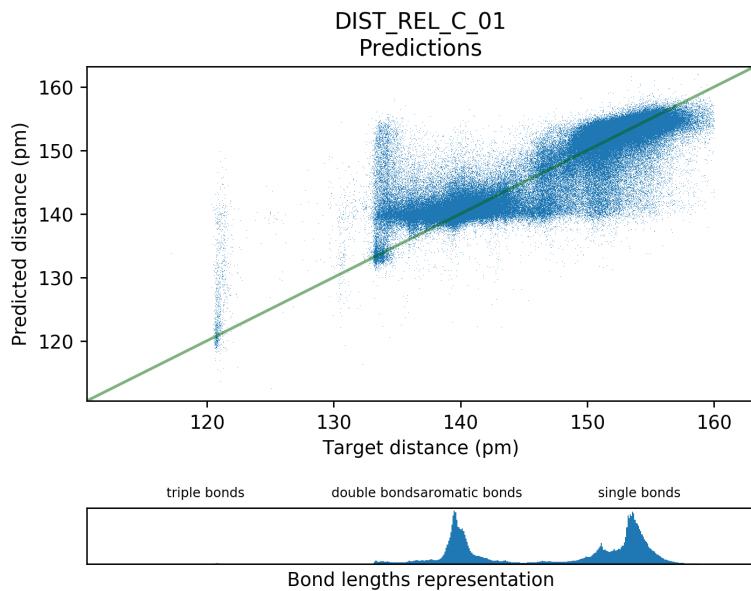


FIGURE 5.10 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_C\_01*

## 5.2.2 Restriction au voisinage le plus proche

### 5.2.2.1 Préparation des données et paramètres

À la différence du modèle naïf qui a en entrée tous les atomes d'une molécule (à l'exception des deux atomes formant la liaison étudiée), nous limitons l'entrée de ce modèle aux atomes au voisinage proche de la liaison (REF RESTRICTION VOISINAGE). Cela a pour objectif de donner uniquement l'information la plus pertinente, et donc de guider le modèle vers de meilleures solutions. Les paramètres spécifiques de préparation des données et d'entraînement sont donnés dans le tableau suivant.

Paramètre	Valeur
Taux d'apprentissage ( <i>learning rate</i> )	0.01
Taille de l'entrée	870
Taille de la dernière couche cachée	870
Profondeur	3
Fonction appliquée aux distances	identité
Restriction du voisinage ( $\epsilon$ )	200 pm

FIGURE 5.11 – Paramètres d'entraînement et de préparation des données spécifiques au modèle *DIST\_REL\_C\_01*

### 5.2.2.2 Analyse statistique des erreurs

Le tableau suivant présente les différentes valeurs des métriques statistiques utilisées pour évaluer les erreurs des modèles. Ses performances sont très intéressantes puisque la moyenne comme la médiane sont en dessous du picomètre. Le modèle semble toutefois toujours faire de grosses erreurs, même si l'erreur maximale a diminué de moitié.

Métrique	Valeur
Moyenne	0,5146
Médiane	0,4225
Écart-type	0,5022
Minimum	0,0000
Maximum	102,3328
Erreur relative moyenne	0,3472%

FIGURE 5.12 – Analyse statistique des erreurs du modèle *DIST\_REL\_C\_02* (en pm)

### 5.2.2.3 Représentation graphique des résultats

La représentation graphique de la distribution des erreurs montre que le modèle est très intéressant, les erreurs au-delà de 10 pm étant marginales.

La représentation graphique de l'erreur en fonction des cibles montre également nettement l'intérêt du modèle. En effet, il ne fait plus d'erreurs importantes sur les liaisons aux tailles limites entre les liaisons aromatiques et simples.

Enfin, la représentation des prédictions en fonction des cibles confirme la continuité des prédictions du modèle entre les différents types de liaisons, ce qui confirme qu'il effectue des prédictions de l'ordre de la chimie quantique, et qu'il ne se contente pas de faire des prédictions basées sur des longueurs de liaisons types.

## 5.2.3 Application de fonctions aux distances

### 5.2.3.1 Préparation des données et paramètres

Les deux modèles décrits dans cette section sont identiques à *DIST\_REL\_C\_02*, à la différence qu'une fonction inversant l'ordre des distances aux atomes de la liaison est appliquée aux données d'entrée (REF FONCT). Cette fonction a pour objectif de guider les modèles vers de meilleures solutions, l'influence des atomes sur la longueur de la liaison étant inversement proportionnelle à leur distance à la liaison. Les résultats des modèles s'en voient améliorés, mais nous montrons en REF XY FONCTIONS que les modèles s'entraînant sur plus d'exemples sont en général plus performants lorsque aucune fonction n'est appliquée aux distances.

Les paramètres d'entraînement spécifiques des deux modèles sont donnés dans la figure suivante.

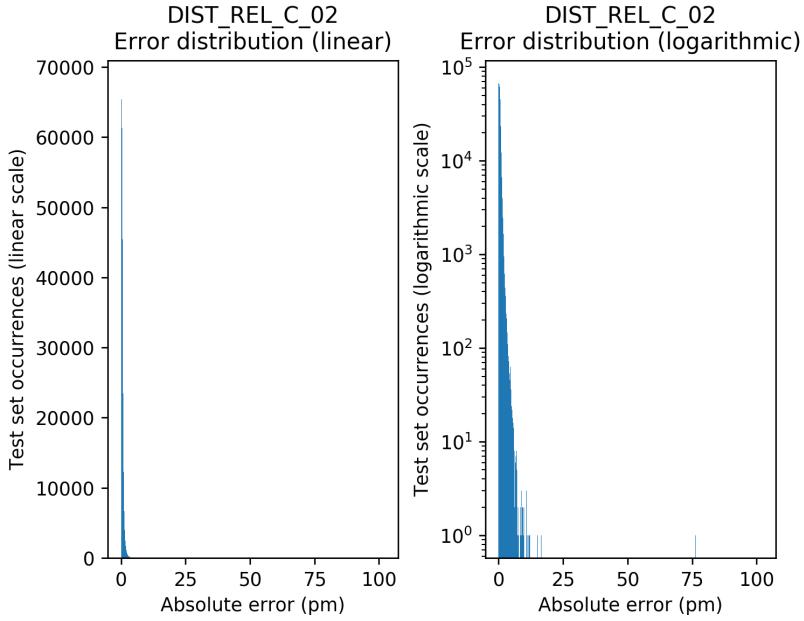


FIGURE 5.13 – Distribution des erreurs du modèle *DIST\_REL\_C\_02*

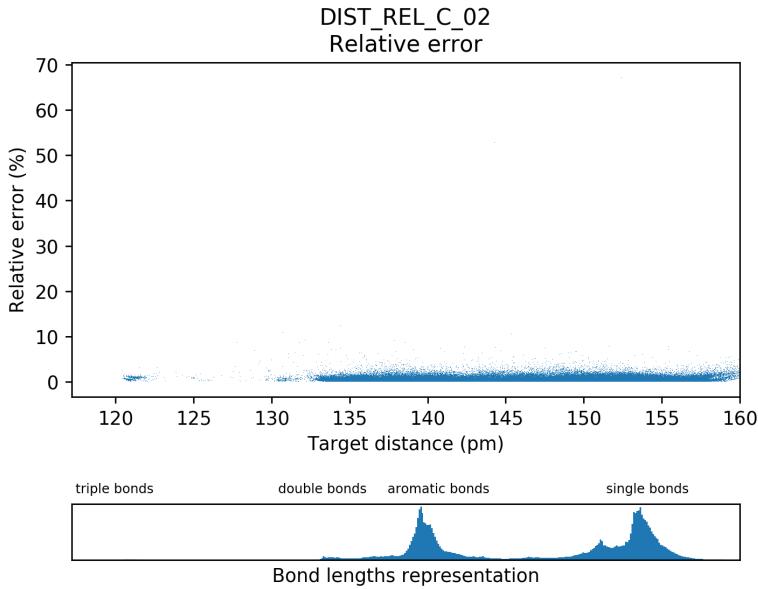


FIGURE 5.14 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_C\_02*

### 5.2.3.2 Analyse statistique des erreurs

Le tableau suivant présente les valeurs des métriques statistiques permettant d'évaluer les erreurs pour les deux modèles. En comparaison au modèle *DIST\_REL\_C\_02*, les deux modèles ont des erreurs maximales largement inférieures. L'utilisation de la fonction inverse fait en revanche augmenter la valeur de toutes les autres métriques. La fonction inverse du carré semble plus prometteuse, car elle fait baisser la moyenne et la médiane. L'écart-type est en revanche légèrement supérieur. Ces différences statistiques étant subtiles, elles sont à prendre avec le recul nécessaire lié au fait que l'on compare des exécutions uniques. Pour s'assurer que ces différences sont constantes, il faudrait entraîner chaque modèle une dizaine de fois et étudier les différentes valeurs de chaque

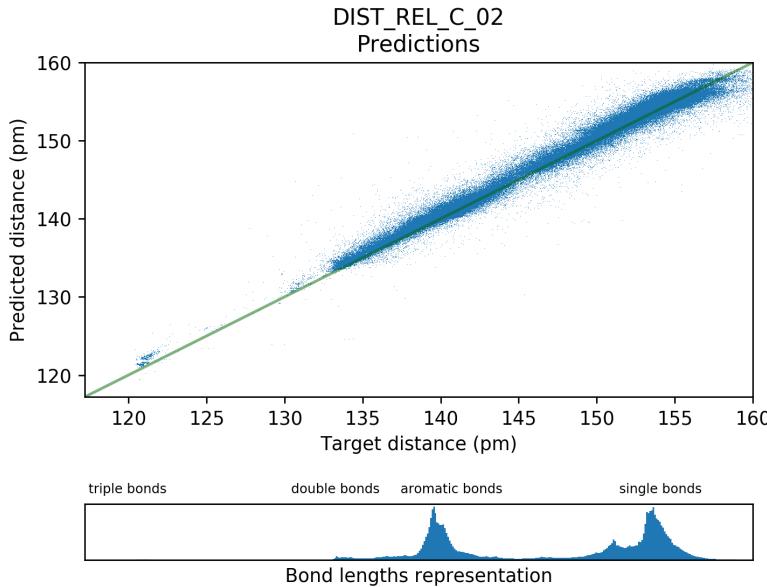


FIGURE 5.15 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_C\_02*

Paramètre	Valeur	
	<i>DIST_REL_C_03</i>	<i>DIST_REL_C_04</i>
Taux d'apprentissage ( <i>learning rate</i> )	0.01	0.01
Taille de l'entrée	870	870
Taille de la dernière couche cachée	870	870
Profondeur	3	3
Fonction appliquée aux distances	inverse	inverse du carré
Restriction du voisinage ( $\epsilon$ )	2 pm	2 pm

FIGURE 5.16 – Paramètres d'entraînement et de préparation des données spécifiques aux modèles

métrique, ce qui représenterait quelques jours d'exécution.

Les représentations graphiques étant très similaires à celles du modèle *DIST\_REL\_C\_02*, nous ne les affichons pas ici. Elles sont néanmoins disponibles en annexes (REF ANNEXES *DIST\_REL\_C\_034*).

Métrique	<i>DIST_REL_C_03</i>	<i>DIST_REL_C_04</i>
Moyenne	0,6255	0,4743
Médiane	0,6255	0,3086
Écart-type	0,6434	0,5521
Minimum	0,0000	0,0000
Maximum	15,479431	17,5180
Erreur relative moyenne	0,4226%	0,3198%

FIGURE 5.17 – Analyse statistique des erreurs des modèles (en pm)

#### 5.2.4 Réduction de la largeur du réseau et des entrées

Les modèles précédents sont très performants, mais ont l'inconvénient de nécessiter un temps d'entraînement élevé. C'est d'autant plus vrai pour les modèles de la seconde classe équivalents (REF INTRO CLASSES ET REF GENERALISATION). Pour tenter d'accélérer les processus d'entraînement, nous modifions la topologie des

réseaux de neurones et nous réduisons la taille des entrées.

Dans les modèles précédents, l'entrée est composée de 58 blocs représentant chacun un atome au voisinage de la liaison (REF REPR DONNEES INTRO). Or depuis le modèle *DIST\_REL\_C\_02*, les entrées ne sont plus composées que des atomes au voisinage le plus proche, tous les blocs ne sont donc plus nécessaires. Une recherche rapide par dichotomie du nombre de blocs tel que tous les exemples des jeux de données ont un nombre d'atomes au voisinage proche qui lui est inférieur ou égal donne une valeur de 15. L'entrée du modèle a alors une nouvelle taille de 15 blocs par 15 attributs, soit 225 attributs contre 870 auparavant.

La topologie des modèles précédents est simple : ils sont composés de trois couches cachées de 870 neurones artificiels. Le modèle que l'on entraîne ici est composé de trois couches cachées de tailles décroissantes. La taille des couches de l'entrée à la dernière couche décroît linéairement de 225 à 15 neurones. La valeur de la taille de cette dernière couche est choisie pour correspondre au nombre maximum d'atomes donnés en entrée, dans l'idée que les modèles pourront extraire couche par couche l'influence de chaque atome. La comparaison entre les deux topologies est donnée dans le tableau suivant.

	<i>DIST_REL_C_X, X&lt;5</i>	<i>DIST_REL_C_05</i>
<b>Taille couche entrée</b>	870	225
<b>Taille première couche cachée</b>	870	120
<b>Taille seconde couche cachée</b>	870	67
<b>Taille troisième couche cachée</b>	870	15
<b>Taille couche sortie</b>	1	1
<b>Nombre de neurones artificiels</b>	901	428
<b>Nombre de connexions</b>	152100	36060

FIGURE 5.18 – Comparaison de la topologie du modèle *DIST\_REL\_C\_05* avec celle des modèles précédents

En dehors de la topologie, les paramètres d'entraînement sont identiques à ceux de *DIST\_REL\_C\_02* (REF MODELE RESTRICTION).

### 5.2.5 Recherche par quadrillage des paramètres du modèle naïf

Afin d'en optimiser les performances, nous effectuons une recherche par quadrillage des paramètres du modèle *DIST\_REL\_C\_01* (REF MODELE NAIF). L'entraînement de chaque modèle prenant plusieurs heures, relativement peu de paramètres différents sont testés. Les paramètres qui varient sont limités à la profondeur du réseau de neurones et au taux d'apprentissage. Chaque entraînement est effectué deux fois dans le cadre d'une validation croisée des résultats. Une validation croisée à deux entraînements ne permet pas de s'assurer de la constance des résultats, mais limite le nombre d'entraînements nécessaires et augmente donc le nombre d'ensembles de paramètres différents pouvant être testés. La performance relative des différents paramètres est évaluée à l'aide de la sortie de tensorboard (REF Tensorboard).

La recherche par quadrillage est en réalité constituée de trois sous-recherches. La grille de la première recherche est choisie arbitrairement, tandis que les grilles des autres recherches dépendent des meilleurs paramètres des recherches précédentes.

Paramètres	Valeurs
Taux d'apprentissage ( <i>learning rate</i> )	0.01, 0.02
Profondeur	1, 2, 3

FIGURE 5.19 – Première grille de recherche par quadrillage pour le modèle *DIST\_REL\_C\_01*

La sortie de tensorboard associée à la première recherche est donnée dans la figure suivante. Elle représente l'erreur moyenne de tous les modèles sur les données de validation au fil de l'entraînement. Les données d'entraînement et de validation forment deux ensembles disjoints. Pour donner un ordre de grandeur des résultats, la différence entre les performances des deux premiers modèles est de l'ordre de 0.3 pm, et la différence entre le meilleur et le pire modèle est de l'ordre de 4 pm. Les meilleurs résultats étant obtenus avec le taux d'apprentissage le plus faible et le réseau le plus profond, on définit une nouvelle grille à partir de cette nouvelle connaissance. À l'issue de la seconde recherche, on utilise la même méthode pour définir une troisième grille de recherche.

Notons les résultats des modèles sur les jeux de test sont meilleurs que pendant l'entraînement, à cause de la désactivation de certains neurones à chaque époque d'entraînement (REF DROPOUT).

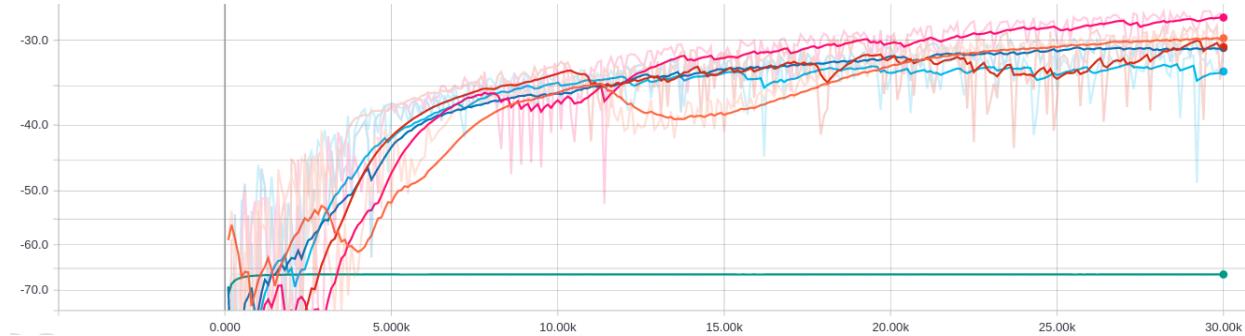


FIGURE 5.20 – Sortie de tensorboard pour la première recherche par quadrillage du modèle *DIST\_REL\_C\_01*

Paramètres	Valeurs
Taux d'apprentissage ( <i>learning rate</i> )	0.01
Profondeur	4, 5

Paramètres	Valeurs
Taux d'apprentissage ( <i>learning rate</i> )	0.005, 0.001
Profondeur	3

Paramètres	Valeurs
Taux d'apprentissage ( <i>learning rate</i> )	0.005, 0.001, 0.008
Profondeur	4, 5, 6

FIGURE 5.21 – Seconde grille de recherche par quadrillage pour le modèle *DIST\_REL\_C\_01*

Paramètres	Valeurs
Taux d'apprentissage ( <i>learning rate</i> )	0.001, 0.0005
Profondeur	7, 8, 9

FIGURE 5.22 – Troisième grille de recherche par quadrillage pour le modèle *DIST\_REL\_C\_01*

La dernière recherche par quadrillage montre que les meilleures performances sont obtenues par le modèle ayant un taux d'apprentissage de 0.001 et une profondeur de 9. Pour donner un ordre de grandeur, la différence entre les performances du meilleur modèle de la première recherche par quadrillage et de la dernière est de 0,5 pm.

### 5.3 Généralisation de la méthode à d'autres liaisons

Les modèles décrits dans cette section appartiennent à la seconde classe (REF CLASSES). Leur objectif est de valider les paramètres et les entrées des modèles de la première classe, et de généraliser la méthode à plusieurs liaisons différentes. Pour cela, nous préparons des jeux de données de grandes tailles, et nous entraînons les modèles sur 300 époques, pour les liaisons carbone-carbone, carbone-hydrogène et oxygène-hydrogène. Les tailles des jeux de données pour chacun des modèles sont données dans le tableau suivant. Si seulement une partie des molécules est explorée pour générer les jeux de données concernant les liaisons carbone-carbone et carbone-hydrogène du fait de leur grande représentation dans les données, toutes les molécules sont explorées pour générer les exemples concernant les liaisons oxygène-hydrogène, qui sont moins nombreuses dans les molécules de notre jeu de données (REF PUBCHEM QC).

Notons que les modèles *DIST\_REL\_C* et *DIST\_REL\_CC* prédisent les mêmes longueurs de liaisons (carbone-carbone), la différence entre les deux étant la taille des jeux de données utilisés et la durée d'entraînement.

Jeu	Modèles <i>DIST_REL_CC</i>	Modèles <i>DIST_REL_CH</i>	Modèles <i>DIST_REL_OH</i>
Entraînement	5541305	3636608	1293551
Test	1106823	1158251	143588

FIGURE 5.23 – Tailles des jeux de données des modèles *DIST\_REL\_XY*

Les paramètres d'entraînement communs utilisés par tous les modèles *DIST\_REL\_XY* sont donnés dans la table suivante.

Paramètre	Valeur
Taille de lot ( <i>batch size</i> )	10000
Epsilon (Adam)	0.001
Initialisation des poids ( <i>stddev_init</i> )	0.001
Fonction d'activation couches cachées	elu
Fonction d'activation couche de sortie	linéaire
Abandon ( <i>dropout</i> )	0.98
Dégénération des coefficients ( <i>weight decay</i> )	0.001

FIGURE 5.24 – Paramètres d'entraînement des modèles *DIST\_REL\_XY*

### 5.3.1 Modèles naïfs

Les trois modèles naïfs sont entraînés avec les mêmes paramètres et le même traitement des données d'entrée que leur modèle équivalent de la première classe (REF DIST REL C 01), c'est à dire qu'aucune restriction au voisinage le plus proche de la liaison n'est appliquée, et qu'aucune fonction n'est appliquée aux distances.

#### 5.3.1.1 Analyse statistique des erreurs

Dans la figure suivante, nous présentons les valeurs des différentes métriques permettant d'évaluer les erreurs pour les modèles prédisant chaque type de liaison. Afin de comparer les performances du modèle prédisant les longueurs de liaisons carbone-carbone avec le modèle de la première classe équivalent, nous rappelons également la valeur des métriques du modèle *DIST\_REL\_C\_01*.

L'entraînement sur un nombre plus élevé d'exemples fait apparaître un gain non négligeable pour la prédiction de longueurs de liaisons entre des atomes de carbone. La moyenne passe en effet en dessous de la barre cible de 1 pm. L'écart-type est cependant toujours au-dessus du pm, même si sa valeur est peu interprétable du fait de la distribution non gaussienne des erreurs.

La prédiction des autres types de liaisons montre de très bons résultats, la plupart des métriques étant à des valeurs proches du dixième de picomètre. Les erreurs maximales ont toutefois des valeurs relativement élevées, notamment dans le cas de la prédiction des liaisons carbone-hydrogène.

Métrique	<i>D_REL_C_01</i>	<i>D_REL_CC_01</i>	<i>D_REL_CH_01</i>	<i>D_REL_OH_01</i>
Moyenne	1,3301	0,8334	0,1753	0,1947
Médiane	0,6881	0,4604	0,1132	0,1153
Ecart-type	1,9810	1,2066	0,1961	0,2519
Minimum	0,0000	0,0000	0,0000	0,0000
Maximum	28,7268	30,1138	22,1473	7,2530
Erreur rel.	0,9126%	0,5709%	0,1599%	0,1986%

FIGURE 5.25 – Analyse statistique des erreurs des modèles (en pm)

### 5.3.1.2 Représentation graphique des résultats

La distribution des erreurs des modèles prédisant les longueurs de liaisons entre les atomes de carbone et d'hydrogène et entre les atomes d'hydrogène et d'oxygène montre que les plus grosses erreurs sont marginales en représentation.

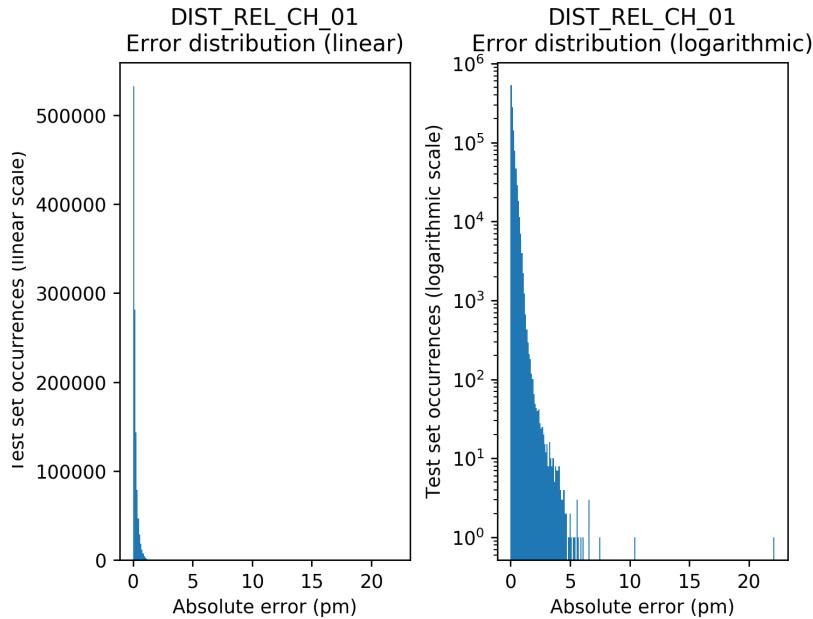


FIGURE 5.26 – Distribution des erreurs du modèle *DIST\_REL\_CH\_01*

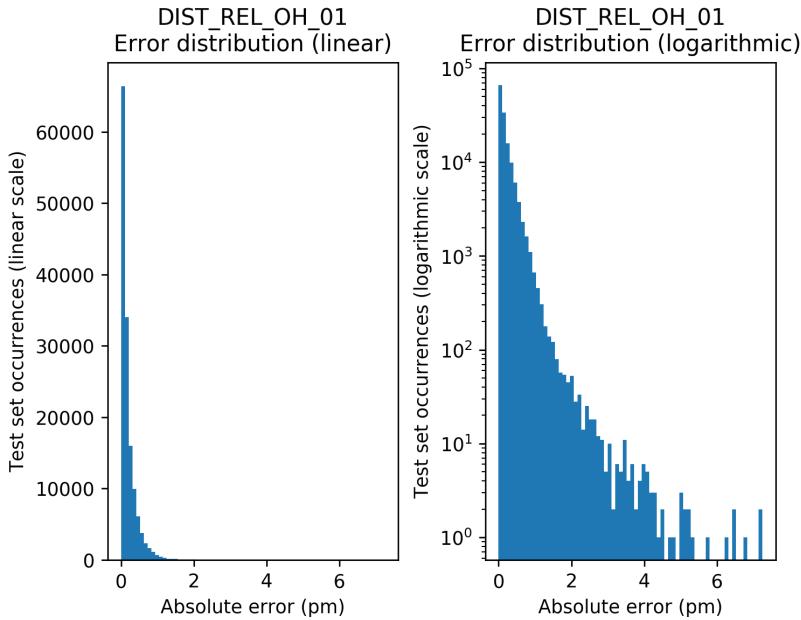


FIGURE 5.27 – Distribution des erreurs du modèle *DIST\_REL\_OH\_01*

La représentation graphique des erreurs relatives et des prédictions en fonction des cibles montre un phénomène intéressant pour le modèle prédisant les longueurs de liaisons entre les atomes d'oxygène et d'hydrogène. La

représentation des longueurs de liaisons OH chutant brusquement en deçà de 97 pm, le modèle s'adapte à cette particularité en ne prédisant jamais des valeurs inférieures à 97 pm.

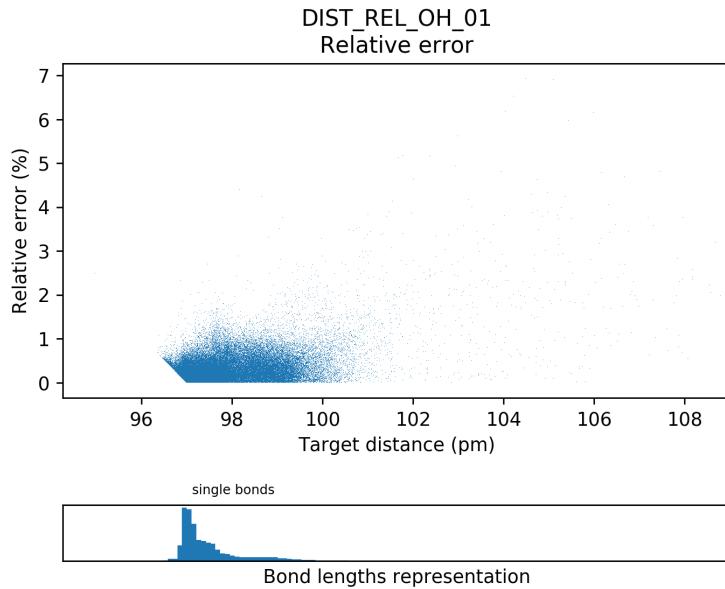


FIGURE 5.28 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_OH\_01*

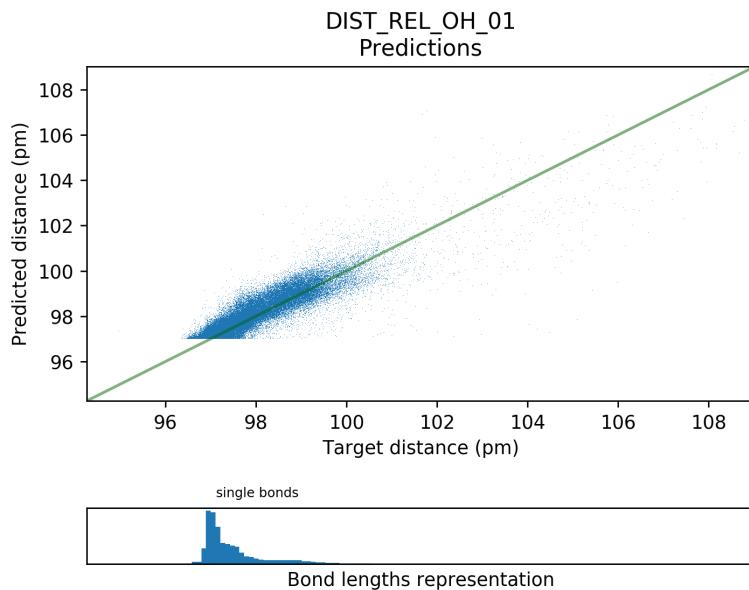


FIGURE 5.29 – Prédictions en fonction des cibles pour le modèle *DIST\_REL\_OH\_01*

Enfin, la représentation graphique des erreurs relatives et des prédictions du modèle prédisant les longueurs de liaison carbone-hydrogène montre que les très bons résultats du modèle s'expliquent en partie par la faible dispersion des longueurs cibles.

Les représentations graphiques concernant le modèle prédisant les liaisons carbone-carbone étant très semblables à celles du modèle *DIST\_REL\_C\_01*, elles sont uniquement visibles en annexes (REF ANNEXES DIST REL XY).

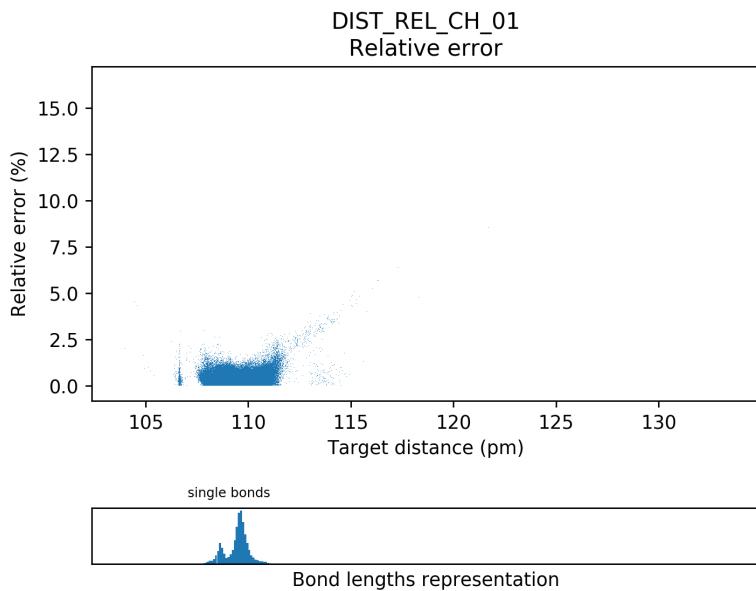


FIGURE 5.30 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_CH\_01*

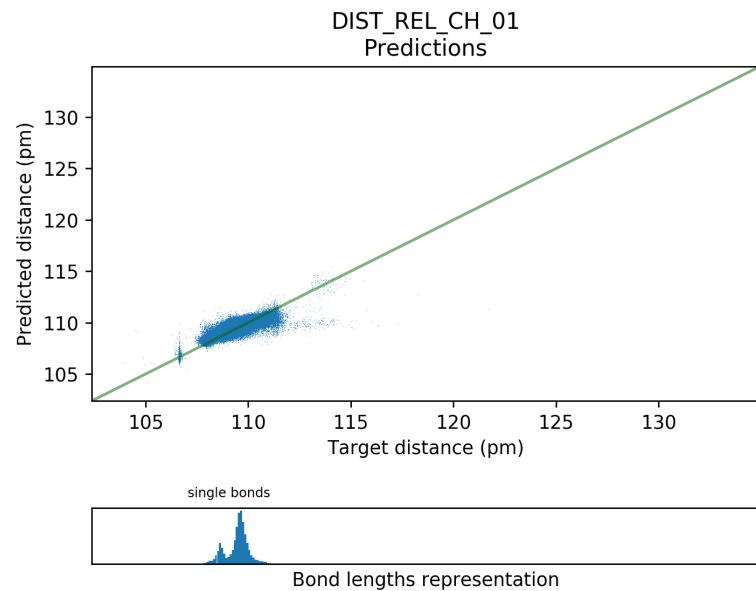


FIGURE 5.31 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_CH\_01*

### 5.3.2 Restriction au voisinage le plus proche

Les modèles décrits dans cette partie sont les modèles de la seconde classe équivalents aux modèles décrits en REF DIST REL C RESTRICT.

#### 5.3.2.1 Analyse statistique des erreurs

Le tableau suivant donne les valeurs des différentes métriques évaluant les erreurs des modèles. Comme attendu, la restriction aux atomes au plus proche voisinage améliore en général les performances des modèles, notamment dans le cas du modèle prédisant les longueurs de liaisons entre des atomes de carbone. Les prédictions

concernant les longueurs de liaisons entre des atomes d'hydrogène et d'oxygène sont également améliorées d'un facteur moindre. Toutefois, les prédictions des longueurs de liaisons entre des atomes de carbone et d'hydrogène sont impactées négativement par cette nouvelle approche, même si l'erreur maximale est diminuée d'un facteur trois.

Métrique	<i>DIST_REL_CC_02</i>	<i>DIST_REL_CH_02</i>	<i>DIST_REL_OH_02</i>
Moyenne	0,3416	0,2390	0,1519
Médiane	0,2667	0,2010	0,1044
Écart-type	0,3373	0,1884	0,1648
Minimum	0,0000	0,0000	0,0000
Maximum	26,2167	6,2905	4,7264
Erreur rel.	0,2295%	0,2184%	0,1552%

FIGURE 5.32 – Analyse statistique des erreurs des modèles (en pm)

### 5.3.2.2 Analyse graphique des résultats

La représentation graphique des erreurs et des prédictions du modèle prédisant les longueurs de liaisons entre des atomes de carbone font nettement apparaître la diminution des erreurs importantes et la continuité des prédictions entre les différents types de liaisons.

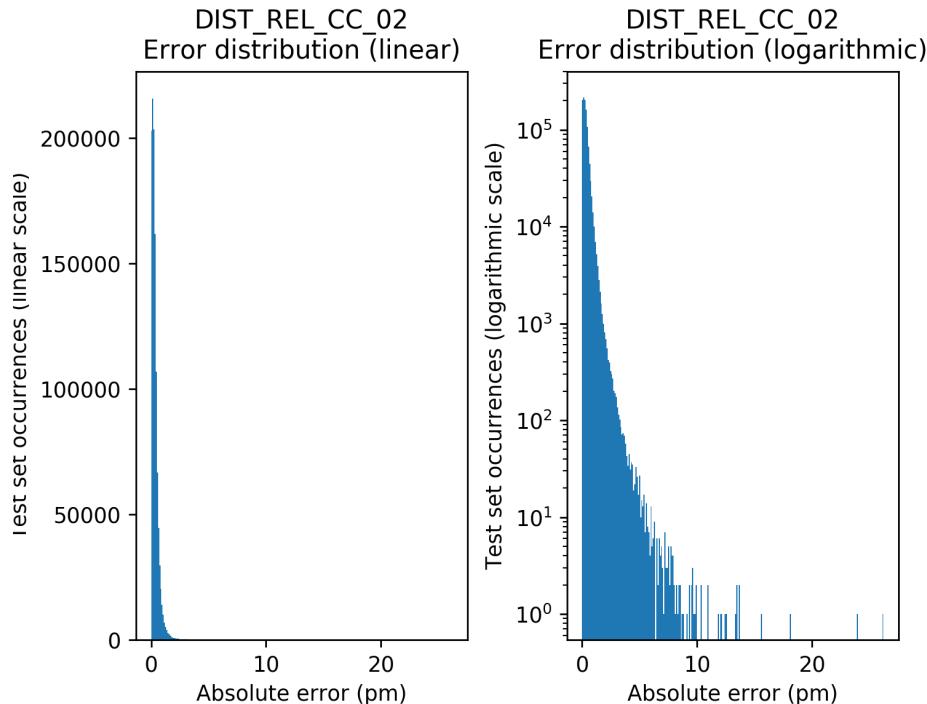


FIGURE 5.33 – Distribution des erreurs du modèle *DIST\_REL\_CC\_02*

La représentation graphique des erreurs et prédictions des deux autres modèles fait apparaître la légère amélioration des prédictions du modèle prédisant les longueurs de liaisons entre les atomes d'oxygène et d'hydrogène, et la baisse de performances du modèle prédisant les longueurs de liaisons entre les atomes de carbone et d'hydrogène. Elles sont disponibles en annexes (REF ANNEXES DIST REL XY).

### 5.3.3 Application de fonctions aux distances

L'application de fonctions aux distances en entrée des modèles (REF DIST REL C FUN DIST) donne des résultats mitigés lorsqu'on tente de généraliser la méthode à plusieurs liaisons différentes sur des jeux d'entraî-

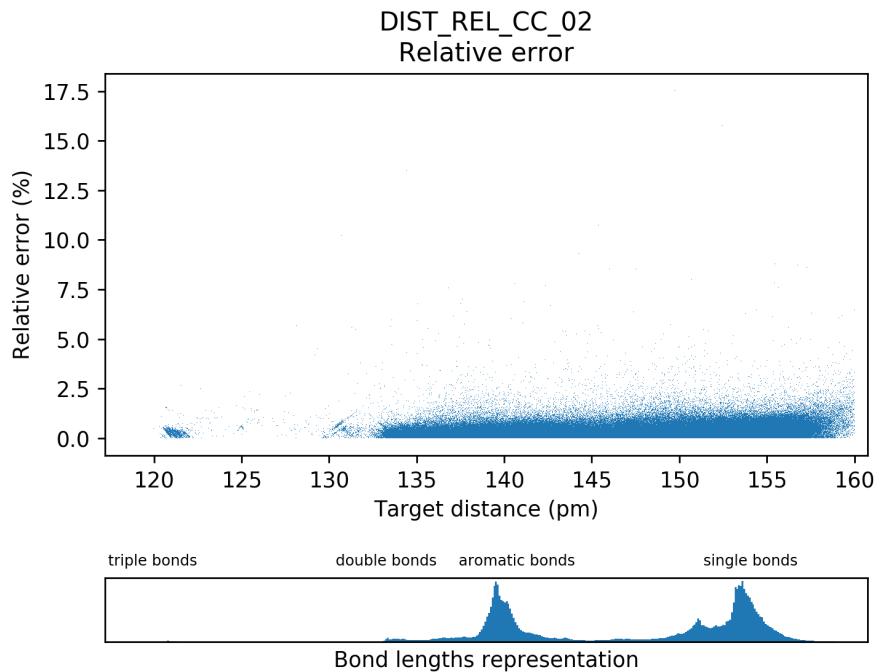


FIGURE 5.34 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_CC\_02*

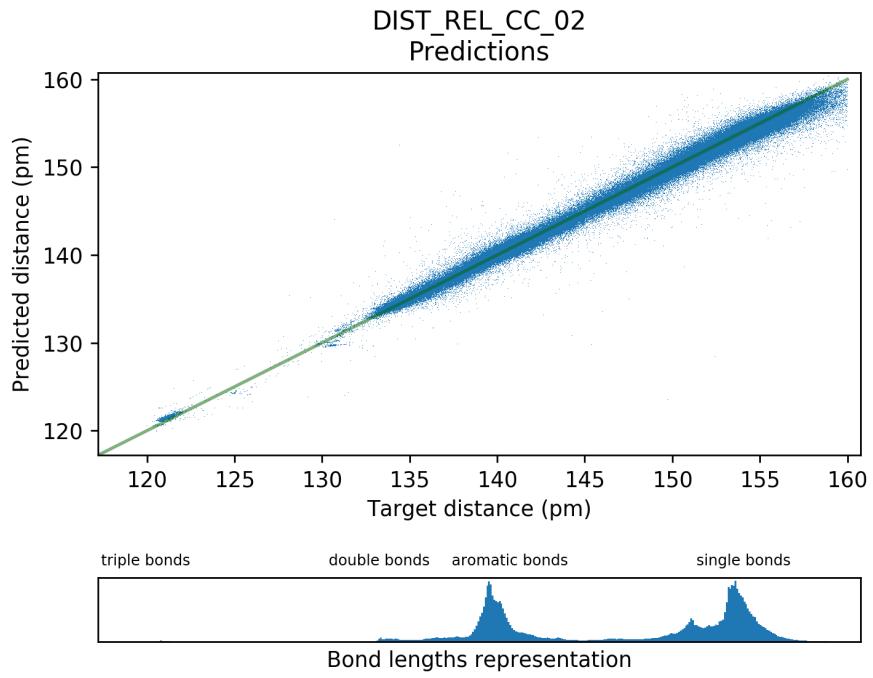


FIGURE 5.35 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_CC\_02*

nement plus grands. Les figures ci-dessous présentent les valeurs des métriques évaluant les erreurs des modèles lorsqu'on applique la fonction inverse et carré de l'inverse aux distances en entrée. Les représentations graphiques des résultats sont disponibles en annexes (REF ANNEXES DIST REL XY) Notons que contrairement aux mo-

dèles de la première classe, le modèle *DIST\_REL\_X\_03* (resp. *DIST\_REL\_X\_04*) correspond à l'application de la fonction carré de l'inverse (resp. fonction inverse). Cela est dû au fait que seule la fonction carré de l'inverse devait être initialement appliquée au vu de ses meilleurs résultats sur le modèle de la première classe, mais ses résultats mitigés lors de la généralisation à d'autres liaisons nous ont poussé à tester également la fonction inverse. Notons enfin que les modèles décrits ici appliquent la restriction aux atomes au voisinage le plus proche (REF DIST REL XY 02).

Métrique	<i>DIST_REL_CC_03</i>	<i>DIST_REL_CH_03</i>	<i>DIST_REL_OH_03</i>
Moyenne	0,4508	0,1634	0,2107
Médiane	0,3290	0,1206	0,1832
Écart-type	0,4582	0,1591	0,1742
Minimum	0,0000	0,0000	0,0000
Maximum	16,6302	6,1820	7,0743
Erreurs rel.	0,3017%	0,1491%	0,2157%

FIGURE 5.36 – Analyse statistique des erreurs des modèles *DIST\_REL\_XY\_03* (en pm)

Métrique	<i>DIST_REL_CC_04</i>	<i>DIST_REL_CH_04</i>	<i>DIST_REL_OH_04</i>
Moyenne	0,4727	0,1659	0,2478
Médiane	0,3773	0,1242	0,2080
Écart-type	0,4288	0,1564	0,2111
Minimum	0,0000	0,0000	0,0000
Maximum	18,2097	7,2360	6,4610
Erreurs rel.	0,3182%	0,1514%	0,2535%

FIGURE 5.37 – Analyse statistique des erreurs des modèles *DIST\_REL\_XY\_04* (en pm)

On peut déjà remarquer que la relation d'ordre de la qualité des performances pour les deux fonctions est identique pour les deux classes de modèles, la fonction inverse du carré étant la plus efficace dans les deux cas. Lorsqu'on compare les prédictions des modèles prédisant les longueurs de liaisons carbone-carbone et hydrogène-oxygène, on s'aperçoit que l'application des deux fonctions détériore la qualité générale des résultats, même si elle permet de diminuer les erreurs maximales. On peut raisonnablement en déduire que ces fonctions ne sont pas optimales pour décrire l'intensité de l'influence des atomes au voisinage d'une liaison en fonction de leur distance aux atomes de la liaison, et qu'il est plus simple pour les réseaux de neurones d'approximer la fonction optimale à partir des distances brutes que des distances transformées.

Les résultats des prédictions pour les distances entre les atomes de carbone et d'hydrogène sont toutefois très surprenants. En effet, l'application du carré de l'inverse sur les distance les améliore d'un facteur deux. La disparité de ces résultats les rend toutefois difficiles à interpréter. Il faudrait dans un premier temps entraîner tous les modèles plusieurs fois sur des données différentes afin d'évaluer la dispersion de leurs résultats. Nous ne pouvons en effet pas affirmer que les différences entre les résultats que nous observons ici ne sont pas dues au hasard des différentes exécutions, même si le fait que la relation d'ordre entre les performances des deux fonctions soit constante sur les différents modèles laisse penser que l'application des fonctions améliore la prédiction des liaisons CC et OH et détériore la prédiction des liaisons CH.

Il semble que nous soyons face à un problème complexe, et qu'il n'existe pas un type de modèle et une représentation des données uniques permettant de prédire de façon optimale tous les types de liaisons au sein des molécules.

## 5.4 Ouverture à d'autres modèles d'apprentissage automatique

Dans le but de comparer les modèles basés sur des réseaux de neurones artificiels que l'on entraîne à d'autres modèles prédictifs d'apprentissage automatique, nous entraînons deux modèles de régression linéaire à effectuer les mêmes tâches. Ces modèles sont une régression d'arête (*ridge regression*) avec l'astuce du noyau (*kernel trick*),

et une machine à vecteur de support (SVM). Nous utilisons pour cela les implémentations fournies par la bibliothèque Scikit-Learn[2]. L'implémentation de ces deux modèles est identique, à la différence que le modèle *Kernel Ridge Regression* (KRR) utilise le carré des erreurs comme fonction de coût, alors que le modèle SVM utilise une fonction  $\epsilon$ -insensible, c'est à dire que les erreurs coûtent leur valeur brute, ou valent zéro si elles sont inférieures à un seuil  $\epsilon$  donné.

### 5.4.1 Données d'entrée et complexité algorithmique

Afin d'avoir une idée des performances relatives de tous ces modèles, nous les entraînons à prédire les distances relatives entre des atomes de carbone formant une liaison. Ces modèles sont équivalents au modèle *DIST\_REL\_C\_05* (REF DIST REL C 05), car on applique une restriction aux atomes les plus proches de la liaison, et on limite la largeur des entrées à 15 atomes. Ce choix de données d'entrée est lié à la nécessité de fournir des entrées de petite taille (comparativement aux données que l'on donne aux réseaux de neurones) à ces modèles, qui ont une complexité d'entraînement augmentant très vite avec le nombre et la taille des exemples. Avec  $n$  le nombre d'exemples et  $m$  leur largeur, la complexité de l'entraînement d'un modèle SVM varie entre  $O(n^2 \times m)$  et  $O(n^3 \times m)$ . La complexité de l'entraînement d'un modèle KRR n'est pas disponible dans la documentation de Scikit-Learn, mais elle plus grande encore que celle des SVM.

Pour la même raison, le nombre d'exemples dans les jeux d'entraînement de ces modèles est beaucoup plus faible que celui des jeux utilisés pour entraîner les réseaux de neurones artificiels. Les modèles que l'on décrit dans cette partie s'entraînent sur des jeux contenant 60000 exemples, ce qui représente tout de même environ cinq jours de temps CPU pour l'entraînement d'un modèle KRR.

Enfin, la fonction inverse est appliquée aux distances dans les données d'entrée de ces modèles. En effet, si les réseaux de neurones artificiels sont capables d'approximer ces fonctions, ce n'est pas le cas des modèles que l'on entraîne ici. L'application de la fonction inverse permet de donner des coefficients aux modèles exprimant l'influence des atomes au voisinage des liaisons en fonction de leur distance.

### 5.4.2 Entraînement de modèles KRR

#### 5.4.2.1 Recherche par quadrillage des paramètres

Afin d'entraîner un modèle fournissant de bons résultats, nous commençons par effectuer une recherche par quadrillage avec trois validations croisées des paramètres pour le modèle KRR. Cette recherche est effectuée sur la grille de paramètres suivante. Selon la documentation, une valeur faible du paramètre alpha va diminuer la variance des erreurs. Le degré correspond au degré du polynôme, et le paramètre coef0 correspond au coefficient constant du polynôme. Les résultats de la recherche par quadrillage sont donnés en annexe (REF ANNEXES QUADRI KER RIDGE).

Paramètres	Valeurs
Noyau ( <i>kernel</i> )	linéaire
Alpha	0.1, 0.01, 0.001

Paramètres	Valeurs
Noyau ( <i>kernel</i> )	polynomial
Degré	2, 6
Alpha	0.1, 0.01, 0.001
Coef0	1, 0.5, 2

FIGURE 5.38 – Grille de recherche par quadrillage des paramètres pour les modèles KRR

#### 5.4.2.2 Entraînement d'un modèle et analyse des prédictions

À l'issue de la recherche par quadrillage, on utilise les meilleurs paramètres pour entraîner un modèle KRR (*DIST\_REL\_C\_KER\_RIDGE\_01*) sur les 60000 exemples de notre jeu d'entraînement. Les paramètres sont donnés dans le tableau suivant.

Paramètre	Valeur
Noyau	polynomial
Degré	2
Alpha	0.01
Coef0	1

FIGURE 5.39 – Paramètres d’entraînement du modèle *DIST\_REL\_C\_KER\_RIDGE\_01*

Les valeurs des métriques évaluant les erreurs sont données dans le tableau suivant. On y voit que le modèle est très performant pour un premier ensemble de paramètres issu d’une petite recherche par quadrillage. L’erreur médiane est en effet de l’ordre du demi picomètre et l’erreur moyenne de l’ordre du picomètre.

Métrique	Valeur
Moyenne	1,0378
Médiane	0,5891
Écart-type	1,2668
Minimum	0,0000
Maximum	21,6264
Erreur relative moyenne	0.7057%

FIGURE 5.40 – Analyse statistique des erreurs du modèle *DIST\_REL\_C\_KER\_RIDGE\_01* (en pm)

La représentation graphique de la distribution des erreurs et des prédictions disponible dans les figures ci-dessous montre que les erreurs au delà de 10 pm sont très minoritaires, mais montre également que le modèle prédit difficilement les distances peu représentées, à la limite entre les différents types de liaisons.

Pour résumer, les modèles prédictifs de type KRR semblent permettent d’obtenir des prédictions proches de la précision requise et pourraient donc être utilisés pour prédire la géométrie convergée complète d’une molécule (REF MODULES), notamment dans le cas des liaisons qui sont peu représentées dans les données, et pour lesquelles les modèles prédictifs basés sur des réseaux de neurones artificiels seront probablement moins efficaces.

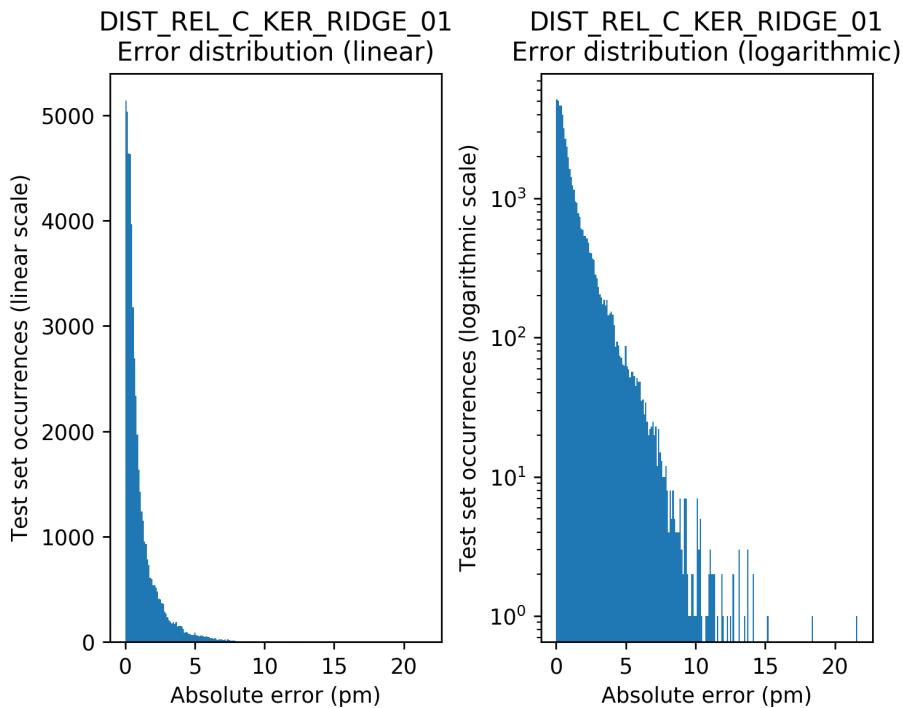


FIGURE 5.41 – Distribution des erreurs du modèle *DIST\_REL\_C\_KER\_RIDGE\_01*

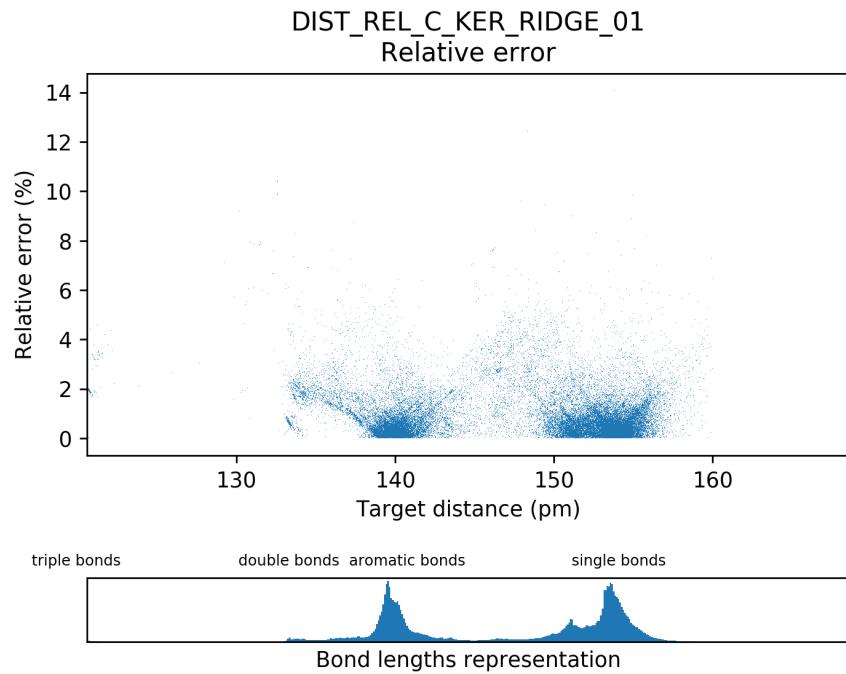


FIGURE 5.42 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_C\_KER\_RIDGE\_01*

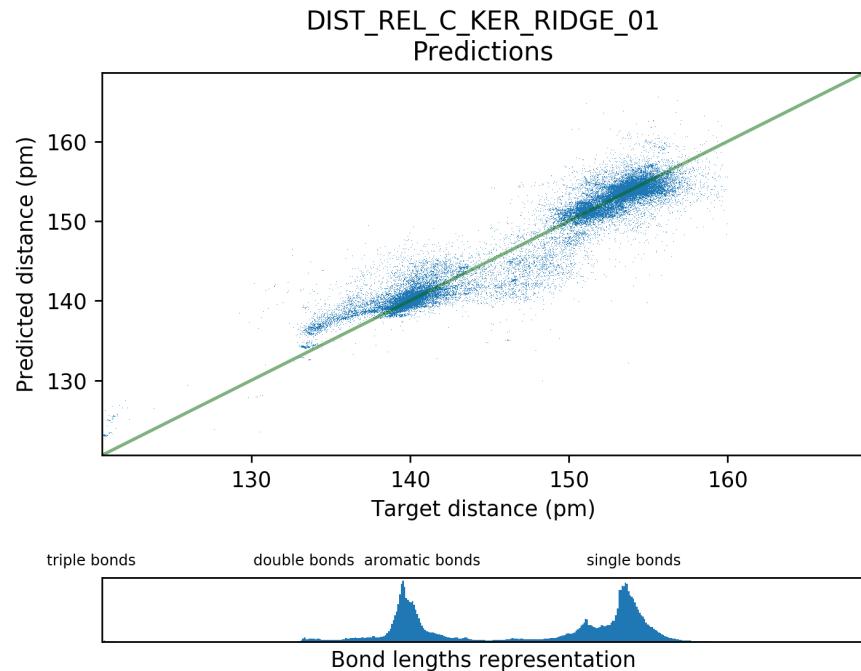


FIGURE 5.43 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_C\_KER\_RIDGE\_01*

### 5.4.3 Entraînement de modèles SVM

#### 5.4.3.1 Recherche par quadrillage des paramètres (non aboutie)

Dans l'idée d'appliquer la même méthodologie que pour le modèle de type KRR, nous définissons une grille de recherche par quadrillage des paramètres des modèles SVM. Cette grille est en revanche plus large, puisqu'elle définit l'entraînement de 256 modèles différents avec trois validations croisées, soit l'entraînement de 768 modèles. La grille de paramètres utilisée est disponible dans la figure ci-dessous. Malheureusement, la recherche n'a pas abouti car certains ensembles de paramètres menaient à l'entraînement de modèles en un temps non raisonnable. Parmi les modèles de la grille, 747 ont été entraînés en environ deux minutes, tandis que l'entraînement des 21 restants n'était pas terminé au bout d'une dizaine d'heures, ce qui a mené à l'interruption de la recherche.

Paramètres	Valeurs
Noyau ( <i>kernel</i> )	linéaire
$\epsilon$ (fonction de coût)	0,1, 0,001
Coef0	0, 0,1, 0,5, 1
Heuristique de rétrécissement ( <i>Shrinking</i> )	True, False
Tolérance de l'arrêt de l'optimisation	0,001, 0,01
Pénalité sur le terme d'erreur (C)	1, 0,01

Paramètres	Valeurs
Noyau ( <i>kernel</i> )	polynomial
$\epsilon$ (fonction de coût)	0,1, 0,001
Gamma (coefficcient utilisé par le noyau)	auto, 0,001, 0,01
Coef0	0, 0,1, 0,5, 1
Heuristique de rétrécissement ( <i>Shrinking</i> )	True, False
Tolérance de l'arrêt de l'optimisation	0,001, 0,01
Pénalité sur le terme d'erreur (C)	1, 0,01

FIGURE 5.44 – Grille de recherche par quadrillage des paramètres pour les modèles SVM

#### 5.4.3.2 Entraînement d'un modèle et analyse des prédictions

La recherche par quadrillage des paramètres n'ayant pas abouti, nous utilisons les paramètres issus de la recherche par quadrillage pour le modèle de type KRR, dans l'idée qu'ils devraient permettre également d'obtenir de bons résultats, et nous laissons les autres paramètres à leur valeur par défaut. Les statistiques des erreurs du modèle alors entraîné sont donné dans le tableau suivant. Notons que le fait que le numéro chronologique du modèle soit « 03 » est la conséquence de l'entraînement de deux modèles préalables dont le but était d'estimer le temps d'entraînement en fonction de la taille des données d'entrée.

Métrique	Valeur
Moyenne	1,2854
Médiane	0,5005
Écart-type	2,2301
Minimum	0,0000
Maximum	24,2225
Erreur relative moyenne	0,8907%

FIGURE 5.45 – Analyse statistique des erreurs du modèle *DIST\_REL\_C\_SVM\_03* (en pm)

Les statistiques comme les figures suivantes représentant graphiquement les erreurs et les prédictions du modèle disponibles montrent que ses résultats sont nettement inférieurs aux modèles précédents. Le modèle se contente en effet de prédire des valeurs de l'ordre des deux types de liaisons les plus représentées, et ne prédit pas correctement les longueurs de liaisons de tailles intermédiaires.

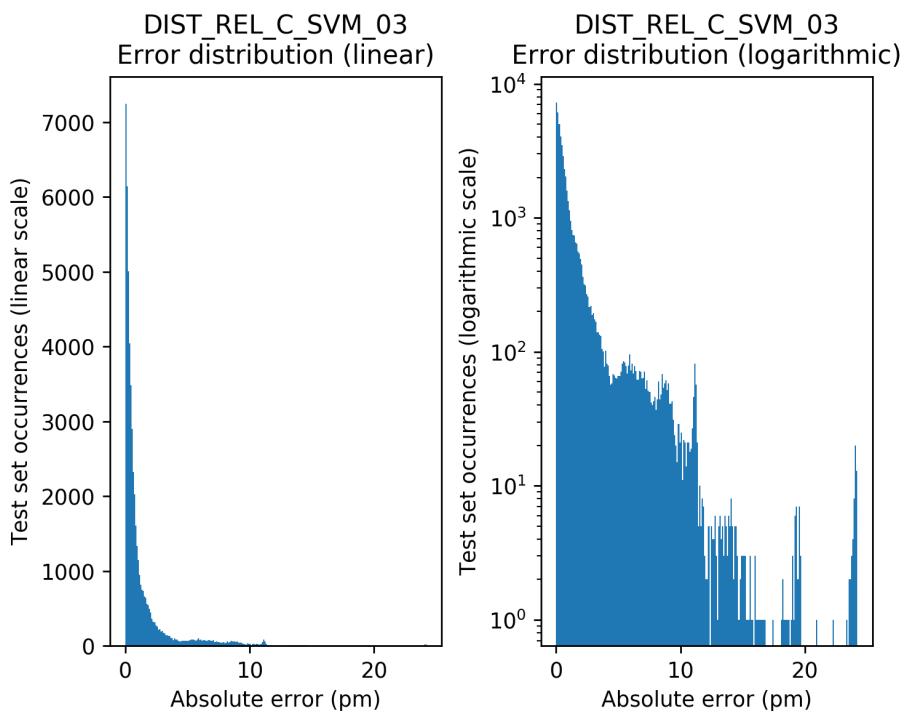


FIGURE 5.46 – Distribution des erreurs du modèle *DIST\_REL\_C\_SVM\_03*

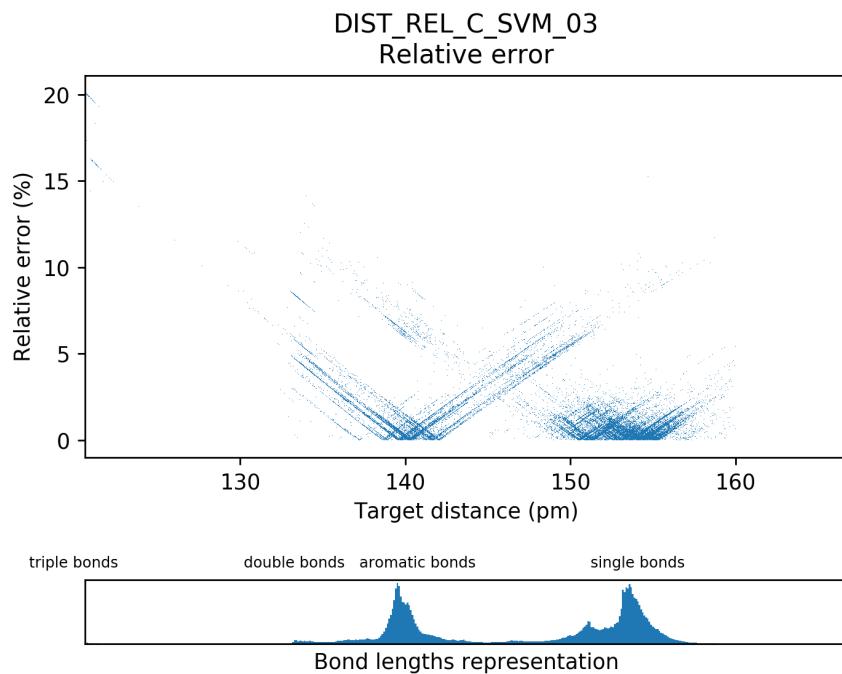


FIGURE 5.47 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_C\_SVM\_03*

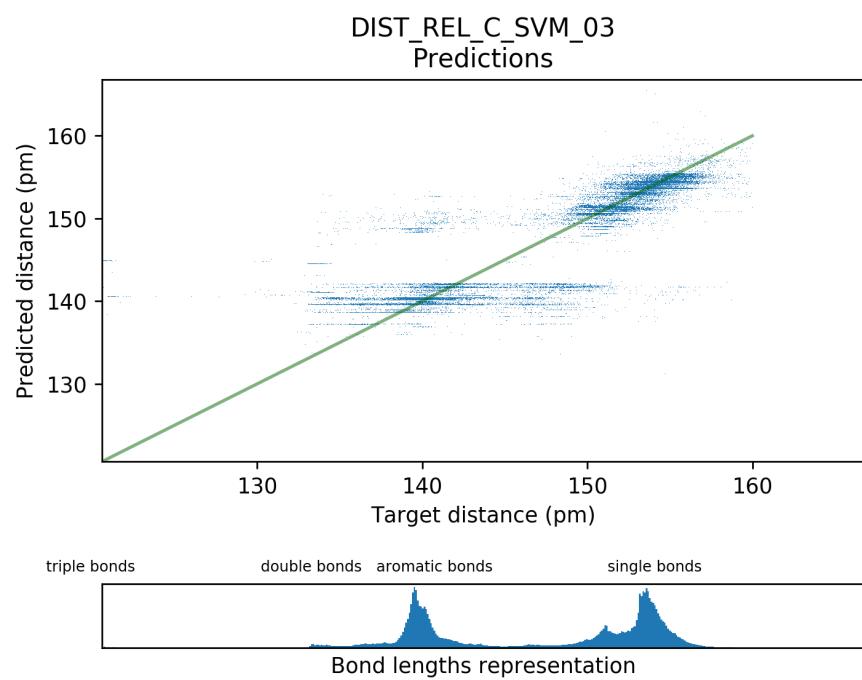


FIGURE 5.48 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_C\_SVM\_03*

# Chapitre 6

## Prédiction de géométries moléculaires convergées

### 6.1 Introduction

#### 6.1.1 Motivation

L'objectif des modèles prédictifs que l'on décrit dans ce chapitre est de prédire la géométrie convergée (REF GEOM CONVERG) d'une molécule complète, à partir d'une géométrie non convergée. Ils sont issus d'une tentative de reproduction de résultats antérieurs, afin de confirmer la méthode élaborée lors des stages précédents sur le projet QuChemPedIA.

Chronologiquement, ces modèles ont constitué la première partie de mon travail, avant de passer aux modèles tentant de prédire les longueurs de liaisons (REF DIST\_REL), à cause de l'impossibilité de produire des prédictions de qualité suffisante (REF RESULTATS).

L'objectif à terme de ces modèles est de pouvoir constituer une alternative au DFT (REF DFT) pour calculer rapidement la géométrie convergée d'une molécule. Cela nécessite de produire des prédictions d'une très grande précision. Cependant, le but ici est avant tout de valider une méthode et notre capacité à produire des prédictions d'ordre géométrique. Nous ne cherchons donc pas à créer un modèle effectuant de très bonnes prédictions, mais plutôt à définir une représentation des données et un ensemble de paramètres permettant d'obtenir de bons résultats.

#### 6.1.2 Méthodologie

**Introduction de bruit** Afin de prédire des géométries moléculaires convergées à partir de géométries moléculaires non convergées, la situation idéale serait que les modèles apprennent à partir d'un ensemble de géométries non convergées issues de mesures ou de résultats théoriques, et l'ensemble associé de géométries convergées issu de l'optimisation géométrique quantique (REF DFT). Cela constituerait en effet un ensemble de données homogène qui aurait l'avantage d'être comparable aux données que l'on utiliserait dans un cas d'utilisation réel.

La solution retenue lors des stages précédents est d'introduire du bruit (REF PREP DONNEES BRUIT) dans les coordonnées des géométries optimisées, et d'entraîner les modèles à prédire ce bruit. La différence entre la géométrie bruitée et le bruit prédit permet alors d'obtenir la géométrie optimisée par le modèle. L'introduction de bruit ne garantit donc pas que les modèles se généraliseront aux données réelles, mais semble tout de même raisonnable pour tenter de valider la méthode, puisque nous entraînons des modèles dont l'objectif est de déplacer les atomes d'une molécule de sorte à obtenir une géométrie convergée.

**Modèles** Cinq modèles différents ont été entraînés. Ils diffèrent par les représentations utilisées en entrée et en sortie, les caractéristiques des molécules dont on tente de prédire la géométrie convergée, et les paramètres propres aux réseaux de neurones comme les fonctions de coût ou la topologie. Dans la section suivante, nous allons répertorier les différentes caractéristiques utilisées de façon non chronologique, puisque aucun ensemble de caractéristiques n'a produit de résultats significativement meilleurs que les autres (REF RESULTATS). Cependant,

une table des caractéristiques utilisées modèle par modèle est disponible en annexe (REF CARAC ANNEXES).

### 6.1.3 Nomenclature

Afin de simplifier leur dénomination, on nomme les différents modèles prédictifs. Tous les modèles décrits dans ce chapitre ont un nom de préfixe « `DELTA_DIST_+H` », issu de leur vocation à prédire des différences ( $\Delta$ ) de distances pour obtenir des géométries convergées. Le suffixe « `+H` » indique que les données d'entrée contiennent les informations concernant les atomes d'hydrogène. Initialement, des modèles ne contenant pas les atomes d'hydrogène en entrée devaient être créés par la suite, mais ce projet a été abandonné faute de pouvoir obtenir des résultats satisfaisants avec le modèle actuel (REF RESULTATS).

Le nom des modèles possède enfin comme suffixe leur numéro chronologique.

## 6.2 Données et paramètres des modèles

### 6.2.1 Données

#### 6.2.1.1 Représentations géométriques

Les modèles que l'on entraîne devant prédire la géométrie des molécules, nous devons leur fournir des données utilisant des représentations synthétisant de la façon la plus simple possible la position des atomes. Nous ne donnons pas les coordonnées brutes aux modèles car ils devraient leur appliquer trop de traitements (REF MAT POS).

Les modèles élaborés lors des stages antérieurs utilisaient la représentation géométrique par matrice réduite des distances inter-atomiques (REF MAT. RED. DIST). Elle est basée sur les distances relatives des atomes et possède donc l'avantage d'être indépendante de tout repère absolu. Cependant, il n'est pas possible de reconstruire systématiquement une matrice des positions atomiques à l'issue des prédictions des modèles utilisant cette représentation en sortie. C'est pourquoi nous avons abandonné cette représentation cette année au profit de la matrice des distances à des points fixes (REF MAT PTS FIXES), qui dépend d'un repère absolu mais dont on peut toujours déduire une matrice des positions atomiques.

Nous avons toutefois entraîné deux modèles de noms `DELTA_DIST_+H_03` (resp. `DELTA_DIST_+H_04`) utilisant comme entrée les deux représentations et comme sortie la représentation par matrice des distances à des points fixes (resp. matrice réduite des distances inter-atomiques). L'entraînement du premier de ces modèles avait pour but de tester si la représentation en repère relatif permettait d'obtenir de meilleurs résultats, et l'entraînement du second avait pour but de vérifier si les mauvaises performances des modèles s'expliquaient par l'utilisation d'une représentation dans un repère absolu. Notons que ce dernier modèle avait uniquement une vocation de test, puisque l'on n'aurait pas été capables de construire la matrice des positions atomiques à l'issue des prédictions, et que l'on n'aurait donc pas pu l'utiliser dans un cas d'utilisation réel.

Nous avons également utilisé une variante de la représentation par matrice des distances à des points fixes comme entrée de l'un des modèles (`DELTA_DIST_+H_02`). Dans cette variante, les points fixes de référence sont considérés comme des atomes fictifs et leurs distances relatives sont donc données. Elles avaient initialement été ignorées car elles sont constantes et les réseaux de neurones sont donc théoriquement capables de les déduire. Ce modèle permettait de s'assurer que les mauvais résultats ne sont pas dus à cette information manquante.

#### 6.2.1.2 Propriétés atomiques

En plus de la géométrie des molécules, nous donnons aux modèles des informations concernant chaque atome et ayant une influence sur la géométrie convergée. Tous les modèles que l'on a entraînés possèdent en entrée la masse atomique de chaque atome, et un des modèles (`DELTA_DIST_+H_05`) possède également les numéros atomiques. De même que pour les distances entre les points du repère, il s'agit d'une information que les réseaux de neurones sont capables de déduire, nous la donnons pour nous assurer que les mauvais résultats ne sont pas dus à son absence.

### 6.2.1.3 Bruit

L'introduction de bruit dans la géométrie moléculaire convergée et le déplacement des atomes selon les prédictions des modèles pour obtenir la géométrie initiale permet de simuler la prédiction de géométries convergées sur des données réelles (REF INTRODUCT BRUIT). Il nous faut toutefois définir précisément quel type de bruit est introduit, quelles sont les données bruitées et quelle est son intensité.

**Nature du bruit** Le bruit que l'on introduit est un bruit gaussien de moyenne nulle. Cela semble un choix raisonnable car la symétrie de la distribution permet a priori d'éloigner autant les atomes les uns des autres que de les rapprocher, et le paramètre d'écart-type  $\sigma$  permet de contrôler son amplitude avec précision.

**Données bruitées** Lors des stages antérieurs, le bruit était introduit sur les distances entre les paires d'atomes, au sein de la matrice réduite des distances inter-atomiques. Cela présentait l'avantage de contrôler précisément ses effets. L'utilisation d'une représentation par matrice des distances à des points fixes rend toutefois impossible l'utilisation de cette méthode, car les distances aux points fixes du repère décrivant un point deviendraient incohérentes entre elles. Cela provoquerait la résolution de nombreuses intersections nulles lors de la reconstruction de la matrice des positions atomiques (REF RECONSTRUCT MAT PT FIXES). Pour pallier ce problème, nous introduisons le bruit sur la matrice des positions atomiques avant de calculer la matrice des distances à des points fixes, ce qui garantie sa cohérence mais nous fait perdre une partie du contrôle des effets du bruit. Le bruit étant ajouté aux coordonnées, on peut en effet difficilement vérifier si le déplacement moyen relatif des atomes est nul et on ne peut donc pas savoir si les atomes sont autant éloignés les uns des autres que rapprochés par le bruit.

**Intensité du bruit** Le déplacement relatif des atomes doit être suffisamment important pour que la tâche d'optimisation de la géométrie moléculaire soit difficile et comparable à des cas d'utilisation réels, mais suffisamment modérée pour que l'on n'inverse pas la position de couples d'atomes, ce qui constituerait une perte d'information trop importante car cela conduirait à tenter d'optimiser des molécules différentes et dans la plupart des cas impossibles selon les lois de la chimie. Un compromis raisonnable semble de déplacer les atomes de 5 pm ( $5 \cdot 10^{-12}$  m) en « moyenne », ou plus précisément d'appliquer un déplacement tel que 68% des atomes sont déplacés de 5 pm ou moins. Cela revient à utiliser le paramètre d'écart-type  $\sigma$  de la loi normale solution de l'équation suivante, exprimant le déplacement d'un atome en pm en fonction de  $\sigma$ . On note  $(x, y, z)$  la position d'un atome dans une géométrie convergée et  $(x', y', z')$  sa position après déplacement.

$$\begin{aligned} 5 &= \sqrt{(x' - x)^2 + (y' - y)^2 + (z' - z)^2} \\ 5 &= \sqrt{\Delta_x^2 + \Delta_y^2 + \Delta_z^2} \\ 5 &= \sqrt{\sigma^2 + \sigma^2 + \sigma^2} \\ 5 &= \sqrt{3\sigma^2} \\ \sigma &= 2,88675 \end{aligned}$$

Certains modèles sont entraînés avec un bruit plus important, tel que 68% des atomes sont déplacés de 30 pm ou moins. On trouve alors avec la même méthode une valeur pour  $\sigma$  de 17,32051. Dans la table des paramètres des modèles en annexe, le bruit faible est noté « + » et le bruit élevé est noté « ++ ».

### 6.2.1.4 Homogénéisation des tailles de données

Les modèles prédictifs possédant une entrée de taille fixe et les molécules une taille (nombre d'atomes) variable, nous devons adapter la représentation des molécules dont on tente de prédire la géométrie convergée pour fournir une représentation homogène de taille fixe.

Le nombre de caractéristiques (*features*) pour chaque atome d'une molécule et un modèle donné est fixe et dépend de la représentation utilisée. Nous ne décrivons ici en détail que la procédure de *padding*<sup>1</sup> pour le modèle *DELTA\_DIST\_+H\_01* car elle est semblable pour tous les modèles.

La représentation géométrique par matrice des distances à des points fixes (REF MAT DIST PT FIXES) est composée de quatre valeurs par atome. Nous ajoutons en outre les masses atomiques, ce qui fait un total de cinq caractéristiques par atome. Pour obtenir une entrée de taille fixe, nous devons déterminer une taille maximale des atomes que l'on accepte dans le modèle. Cette borne est ici fixée à 200. La taille de l'entrée du modèle est alors le produit du nombre de caractéristiques et de la taille maximale des molécules soit ici 1000. Lorsqu'une molécule est de taille inférieure à la taille maximale, les caractéristiques des atomes non définis sont fixées à zéro. Une schématisation de cette entrée est donnée dans la figure suivante.

$d_{a_1,p_0}$
$d_{a_1,p_1}$
$d_{a_1,p_2}$
$d_{a_1,p_3}$
$m_{a_1}$
:
$d_{a_n,p_0}$
$d_{a_n,p_1}$
$d_{a_n,p_2}$
$d_{a_n,p_3}$
$m_{a_n}$
0
:
0

FIGURE 6.1 – Entrée du modèle *DELTA\_DIST\_+H\_01* pour une molécule de taille n

De même, la sortie du modèle est de taille fixe, mais n'est composée que des valeurs propres à la matrice de distances à des points fixes. À la différence de l'entrée, nous n'attendons pas que les valeurs concernant les atomes non définis soit nulles (REF EVALUATION). Une schématisation de la sortie est donnée dans la figure suivante.

$d_{a'_1,p_0}$
$d_{a'_1,p_1}$
$d_{a'_1,p_2}$
$d_{a'_1,p_3}$
:
$d_{a'_n,p_0}$
$d_{a'_n,p_1}$
$d_{a'_n,p_2}$
$d_{a'_n,p_3}$
?
:
?

FIGURE 6.2 – Sortie du modèle *DELTA\_DIST\_+H\_01* pour une molécule de taille n

### 6.2.1.5 Unités

De même que pour les modèles prédisant les longueurs de liaisons (REF UNITES DIST\_REL), les modèles décrits ici effectuent leurs prédictions en mÅ, afin d'évaluer des prédictions dans un ordre de grandeur de 10<sup>2</sup> et

---

1. rembourrage

d'éviter que la fonction d'évaluation (REF EVALUATION) ne soit influencée par la présence de valeurs proches de zéro, qui les minimiseraient du fait de la présence d'un carré.

#### 6.2.1.6 Synthèse du flux de données

Le flux de données général des modèles est donc le suivant. On extrait les caractéristiques d'une molécule (géométrie et différentes propriétés atomiques des atomes), on ajoute du bruit à la géométrie puis on tente de prédire le vecteur bruit. La différence entre la géométrie bruitée et le vecteur bruit donne enfin la géométrie convergée prédictive par le modèle. Ce flux est synthétisé dans le schéma suivant.

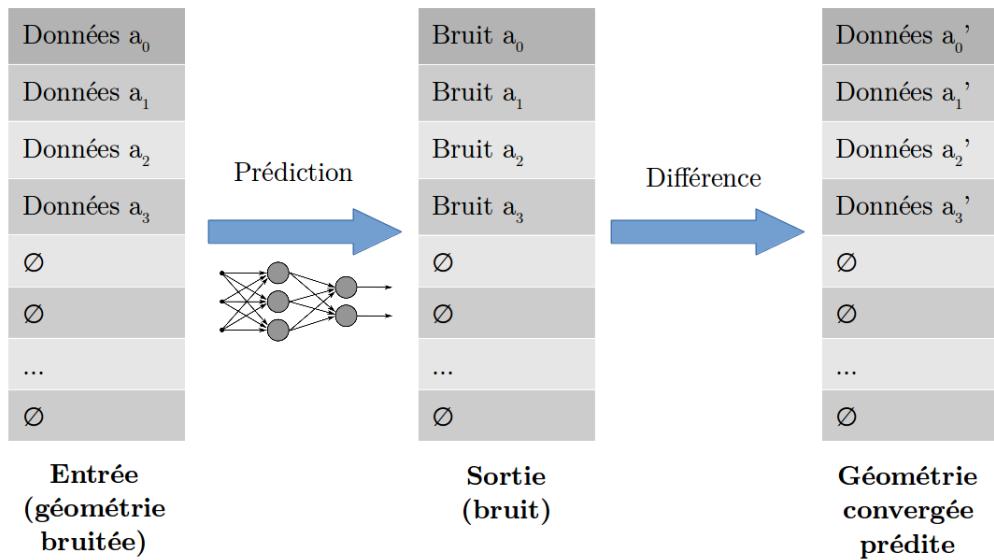


FIGURE 6.3 – Flux de données des modèles *DELTA\_DIST+H* pour une molécule de taille 4

#### 6.2.2 Fonctions d'évaluation

##### 6.2.2.1 Fonctions de coût

Afin d'évaluer la qualité des prédictions et pour guider les modèles lors de la procédure d'optimisation des poids (REF DEEP LEARNING), nous devons définir une fonction de coût. Pour chaque prédition évaluée, celle-ci doit renvoyer une valeur évaluant sa qualité. Par définition, plus la prédition est bonne et plus le coût associé doit être faible. Pour évaluer la sortie des modèles qui est constituée de multiples valeurs, nous utilisons la métrique *Root Mean Square Error* (RMSE). Celle-ci consiste à calculer la moyenne du carrés des erreurs (différence entre le vecteur prédit et le vecteur attendu), puis à appliquer une racine carrée pour remettre le résultat dans l'ordre de grandeur des données d'entrée.

Ce RMSE (que l'on qualifie de total) est toutefois trop simpliste pour nos modèles car il considère toutes les valeurs du vecteur bruit prédit, alors que certaines valeurs correspondant à des atomes non définis en entrée doivent être ignorées (REF HOMOGEN TAILLE DONNEES). C'est pourquoi nous définissons une métrique que l'on nomme RMSE partiel et qui utilise un masque pour ne calculer le RMSE que sur les valeurs prédictes correspondant à des valeurs non nulles en entrée.

Sans l'utilisation du RMSE partiel, les résultats d'évaluation des modèles seraient trompeurs à cause du fait que la plupart des vecteurs cibles (bruit à prédire) contiennent de nombreux zéros du fait de la nécessité d'avoir des entrées et sorties de taille fixe (REF HOMOGEN TAILLE DONNEES) et de la distribution des tailles de molécules (REF DONNEES TAILLES MOL). En effet, le RMSE total évaluerait en grande partie la capacité des modèles à prédire des valeurs nulles, ce qui constitue une tâche très simple et éloignée de nos objectifs.

Si tous les modèles ont été entraînés avec le RMSE partiel comme fonction de coût, un des modèles (voir table des paramètres en annexe) a été entraîné une seconde fois avec le RMSE total comme fonction de coût. Cela

avait pour but de tester si le changement de fonction de coût le guidait vers de meilleures solutions. Toutefois, afin d'avoir une mesure objective des performances, l'opposé du RMSE partiel était alors utilisé comme fonction de validation.

#### 6.2.2.2 Fonctions de validation

En plus des fonctions de coût qui permettent de guider les modèles vers de bonnes solutions lors de l'entraînement, nous utilisons deux fonctions de validation qui ont pour objectif d'évaluer les performances des modèles sur les jeux de test (REF DONNEES JEUX TEST). Les premiers modèles utilisaient le score R2<sup>2</sup>, défini comme le quotient de la somme du carré des erreurs par la somme du carré de l'écart des valeurs cibles à la moyenne. Le score R2 a peu à peu été abandonné au profit de l'opposé du RMSE partiel, notamment dans le but d'uniformiser l'évaluation des modèles entre leur entraînement et leur test sur des données inconnues.

#### 6.2.2.3 Erreur introduite par le bruit

Afin d'évaluer les bénéfices des prédictions des modèles par rapport aux données géométriques bruitées, nous calculons le RMSE (REF FONCT COUT) des données bruitées. Formellement, nous calculons la moyenne des RMSE partiels des vecteurs bruit sur tout le jeu de données. Cela nous donne une idée précise de l'erreur introduite par le bruit. Tout modèle possédant un RMSE partiel inférieur à cette valeur sur le jeu de test aura donc prédit une partie du bruit et mené à une amélioration de la géométrie. Le RMSE du bruit introduit « faible » est d'environ 2,8 pm et celui du bruit « fort » est de 17,2 pm (REF BRUITS).

### 6.2.3 Architectures

Les modèles décrits dans ce chapitre sont tous des réseaux de neurones possédant des architectures simples. Ils sont composés d'une entrée et d'une sortie dont la taille dépend des données qu'ils doivent traiter (REF DONNEES), et d'un certain nombre de couches internes de taille fixe et entièrement connectées, c'est à dire que chaque neurone d'une couche est connecté à tous les neurones de la couche suivante.

Le nombre de couches et le nombre de neurones par couche varie en fonction des modèles. Les premiers modèles possédaient des couches internes plus larges que les entrées et sorties, ce qui pouvait potentiellement apporter un gain de performances mais qui augmentait de manière significative le temps d'entraînement. C'est pourquoi le dernier modèle est composé de couches internes de même taille que la couche d'entrée. Le détail est disponible dans la table des paramètres en annexe (REF TABLE ANNEXE).

#### 6.2.4 Optimisation des paramètres

En plus du choix des données d'entrée, la performance des réseaux de neurones dépend de nombreux paramètres (REF PARAMS RN). Les résultats des modèles décrits dans ce chapitre étant peu probants (REF RESULTATS), j'ai effectué une recherche par quadrillage (REF RECHERCHE QUADRI) large des différents paramètres pour le modèle *DELTA\_DIST\_+H\_05*, avec l'objectif de trouver un ensemble de paramètres menant à de meilleures performances. De même que pour les modèles décrits dans le chapitre précédent (REF QUADRI DIST REL), le temps d'exécution de l'entraînement d'un modèle limite grandement la possibilité d'entraîner des modèles avec des jeux de paramètres variés et un nombre élevé de validations croisées (REF VALID CROISEE) en un temps raisonnable. Il faut donc effectuer un compromis entre la quantité de modèles différents entraînés et le nombre d'entraînements de chacun de ces modèles. L'objectif ici étant de trouver un jeu de paramètres menant à de bonnes performances, dans l'idée de le perfectionner et de le valider par la suite s'il existe, la priorité est donnée au nombre de jeux de paramètres différents plutôt qu'au nombre de validations de chacun de ces jeux.

Cette recherche par quadrillage est toutefois relativement large car elle est composée d'une grille décrivant les paramètres de 576 modèles différents avec deux validations croisées, soit un total de 1152 entraînements pendant environ cinq jours. La grille se veut également large car elle fait varier la plupart des paramètres avec des amplitudes élevées.

À l'issue de la recherche, aucun ensemble de paramètres n'a mené à de meilleures performances que les modèles précédemment entraînés.

2. [https://en.wikipedia.org/wiki/Coefficient\\_of\\_determination](https://en.wikipedia.org/wiki/Coefficient_of_determination)

Paramètres	Valeurs
Taux d'apprentissage ( <i>learning rate</i> )	0.1, 0.0001, 0.00001
Epsilon (Adam)	1000, 0.0001
Initialisation poids	0.2, 0.002
Fonction d'activation couches cachées	elu, crelu
Fonction d'activation couche de sortie	linéaire
Dégénération des coefficients ( <i>weight decay</i> )	0.1, 0.01, 0.001
Largeur	500
Profondeur	7, 3
Taille de lot ( <i>batch size</i> )	500, 2000
Époques d'entraînement	3

FIGURE 6.4 – Grille de recherche par quadrillage pour le modèle *DELTA\_DIST\_+H\_05*

## 6.3 Résultats

### 6.3.1 Estimation des performances lors de l'entraînement

Lors de l'entraînement d'un modèle, la sortie texte de tensorflow et la sortie graphique de tensorboard (REF SORTIE TENSOR) permettent d'avoir une estimation de la valeur de la fonction de coût (REF RMSE) sur des données qui lui sont inconnues. Cela permet d'avoir une idée de la performance relative des modèles, sans faire d'analyse détaillée comme dans la section suivante. Tous les modèles décrits dans ce chapitre (en dehors des modèles les moins performants de la recherche par quadrillage REF QUADRI) ont des performances très similaires pendant l'entraînement. Les modèles travaillant sur des données ayant un bruit de RMSE (REF ERREUR BRUIT) 2,8 pm (resp. 17.2 pm) effectuent des prédictions de RMSE 1,8 pm (resp. 10,7 pm). Dans les deux cas, cela revient à réduire l'erreur à environ 63% de sa valeur initiale, et donc à prédire 37% du bruit. Il s'agit d'un gain non négligeable, mais nous allons montrer dans la sous-partie suivante qu'il n'est pas réellement utilisable pour optimiser la géométrie des molécules.

### 6.3.2 Analyse détaillée d'un modèle

Nous allons ici analyser les prédictions du modèle *DELTA\_DIST\_+H\_05*. Tous les modèles ayant des performances similaires (REF ESTIM PERF ENTRAIN), nous supposons que l'analyse de leurs résultats est similaire à celle que l'on développe ici.

Le modèle ayant une sortie composée de multiples valeurs, nous allons décomposer ses prédictions afin de pouvoir les analyser. Ce que l'on va nommer par la suite prédictions est l'ensemble des composantes de tous les vecteurs de sortie du modèle sur le jeu de test après l'application d'un masque ne sélectionnant que les composantes de sortie correspondant à des atomes définis en entrée (REF HOMOG TAILLES). De même, ce que l'on nomme cibles est l'ensemble des valeurs attendues en sortie sur tous les exemples du jeu de test après sélection des valeurs correspondant à des atome définis, et le vecteur erreurs est alors la valeur absolue de la différence entre ces deux vecteurs.

#### 6.3.2.1 Analyse statistique

Dans un premier temps, nous allons effectuer une analyse statistique des valeurs présentes dans les vecteurs cibles, prédictions et erreurs.

Moyenne	-0,8216
Médiane	-0,8198
Écart-type	17,3062
Minimum	-94,7950
Maximum	97,2401

FIGURE 6.5 – Analyse statistique des valeurs cibles (en pm)

Les cibles correspondent au déplacement des atomes de la molécule par le bruit relativement à quatre points fixes du repère (REF DONNEES REPR GEOM). Le bruit étant gaussien, la moyenne et la médiane sont comme attendu très proches. L'écart-type des déplacements est très proche de la valeur donnée comme paramètre lors de l'introduction du bruit (REF INTENSITE BRUIT), ce qui est également prévisible. Le fait d'ajouter le bruit sur les coordonnées et non les distances (REF DONNEES BRUITEES) a cependant déplacé le déplacement moyen de zéro vers une valeur légèrement négative, c'est à dire que les atomes ont en moyenne été plus rapprochés de l'origine du repère qu'éloignés.

Moyenne	-0,2328
Médiane	-0,1346
Écart-type	10,4515
Minimum	-9,5675
Maximum	1,2347

FIGURE 6.6 – Analyse statistique des prédictions (en pm)

La moyenne et la médiane des prédictions étant décalées, elles ne suivent pas une distribution gaussienne comme attendu. L'intervalle des valeurs prédites n'est pas centré sur zéro mais est nettement déplacé vers les valeurs négatives. Cela s'explique par le centrage des cibles sur une valeur légèrement négative. L'écart-type n'étant pas comparable avec l'écart-type des valeurs cibles car la distribution n'est pas gaussienne, il est difficile d'estimer la dispersion des prédictions. Elles semblent toutefois très proches de zéro, comparativement aux valeurs attendues. En effet, les prédictions s'étendent entre -9,6 et 1,2, alors qu'on souhaiterait qu'elles s'étendent entre -94,8 et 97,2 dans les cas extrêmes, et qu'elles soient comprises entre -30,0 et 30,0 dans le cas général (REF INTENSITE BRUIT). Le modèle n'arrive donc pas à suffisamment déplacer les atomes pour obtenir les géométries convergées.

Moyenne	13,8335
Médiane	11,6937
Écart-type	10,4515
Minimum	0,0000
Maximum	97,7970

FIGURE 6.7 – Analyse statistique des erreurs absolues (en pm)

On souhaiterait que les erreurs soient de l'ordre de 1 pm, alors qu'elles sont en moyenne de 13,9 pm. Cette erreur moyenne importante est prévisible puisque les prédictions sont dans un intervalle d'amplitude environ vingt fois plus faible que l'amplitude de l'intervalle des valeurs cibles.

### 6.3.2.2 Distribution de l'erreur absolue

Afin de comprendre la façon dont les erreurs sont distribuées, nous représentons graphiquement leur représentation en fonction de leur valeur.

L'erreur semble suivre une distribution gaussienne. L'erreur présentée ici étant l'erreur absolue, nous ne voyons qu'une demie courbe de Gauss. Cela montre que le bruit a été en partie « absorbé » par la prédiction (37%, REF ESTIM PERF) mais que sa distribution est restée semblable.

### 6.3.2.3 Distribution de l'erreur absolue en fonction des cibles

La représentation de la distribution de l'erreur absolue en fonction des cibles montre que la majorité des prédictions sont très proches de zéro, et que les autres sont proches de -9. Il s'agit probablement d'une méthode pour le modèle de minimiser « en moyenne » la fonction de coût. Les prédictions autour de -9 font probablement partie des raisons pour laquelle la fonction de coût diminue de 37% par rapport à l'erreur introduite par le bruit. Le modèle semble en effet ici apprendre une règle d'ordre géométrique qui lui permet de minimiser l'erreur dans certains cas.

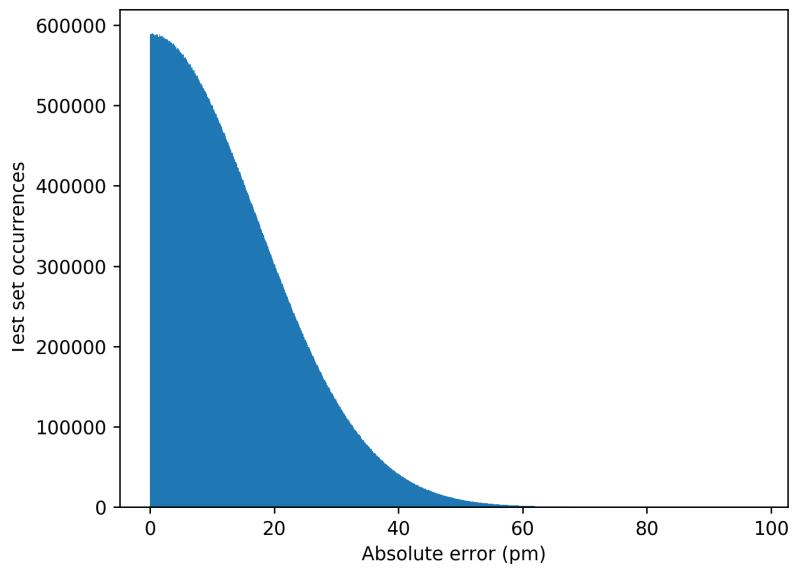


FIGURE 6.8 – Distribution des erreurs du modèle *DELTA\_DIST\_+H\_05*

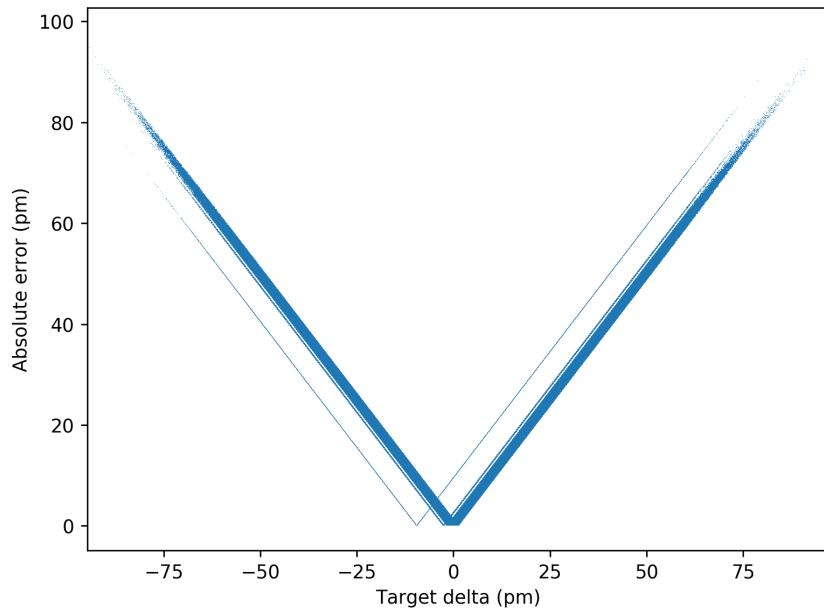


FIGURE 6.9 – Erreur en fonction des cibles pour le modèle *DELTA\_DIST\_+H\_05*

#### 6.3.2.4 Distribution des prédictions en fonctions des cibles

La droite tracée correspond aux valeurs attendues. Les prédictions du modèle ont une intersection avec la droite limitée aux prédictions proches de zéro et de -9.

Lorsque l'on regarde de plus près les prédictions, on s'aperçoit qu'elles prennent des valeurs discrètes. Cette subtilité peut en partie expliquer la capacité du modèle à prédire partiellement le bruit.

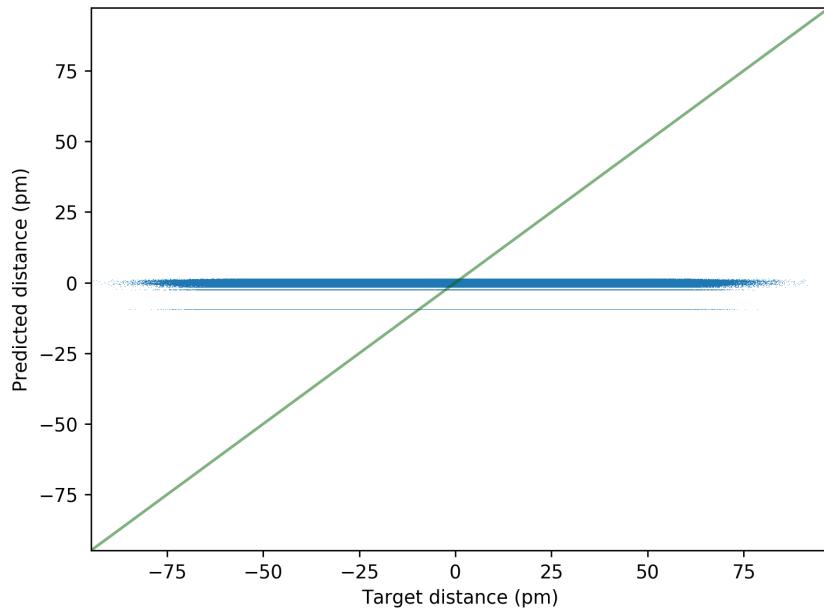


FIGURE 6.10 – Prédiction en fonction des cibles pour le modèle *DELTA\_DIST\_+H\_05*

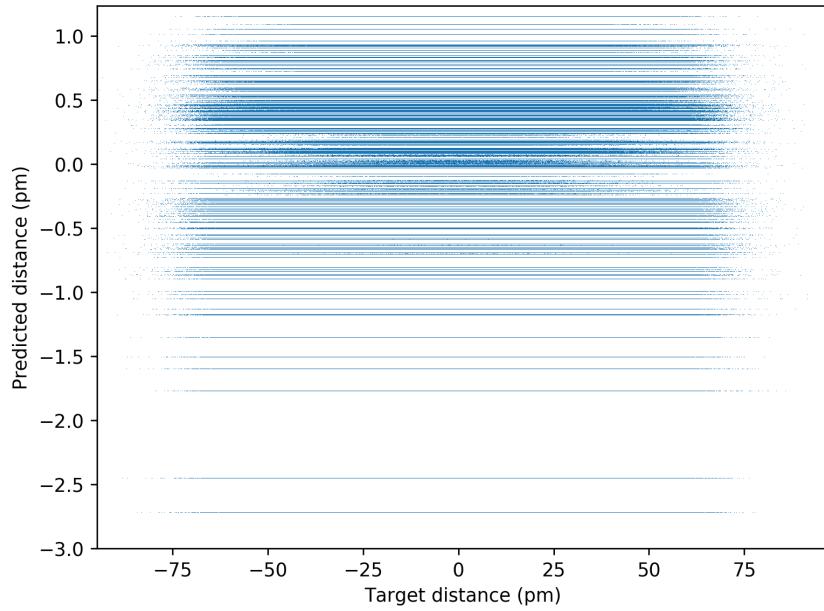


FIGURE 6.11 – Prédiction en fonction des cibles pour le modèle *DELTA\_DIST\_+H\_05* (zoom)

### 6.3.3 Abandon de la méthode

Le fait que le modèle effectue des prédictions constantes et l'impossibilité de produire de meilleurs résultats à l'issue de la recherche par quadrillage (REF QUADRI) ont mené à l'abandon de la méthode pour prédire des géométries moléculaires convergées, au profit d'une méthode moins ambitieuse (REF DIST REL).

Il est démontré qu'il existe une fonction permettant d'optimiser la géométrie moléculaire, et que les réseaux de neurones sont des approximateurs universels de fonctions[3]. La tâche que nous avons tenté d'accomplir avec ces modèles est donc théoriquement possible. Nous pouvons toutefois trouver quelques explications possibles à

notre incapacité à entraîner un modèle suffisamment efficace.

Premièrement, les modèles que nous avons entraînés sont des modèles aux architectures relativement simples, avec un nombre de neurones et de connexions limité par les capacités matérielles. Des architectures plus complexes auraient pu mener à de meilleures performances pour les mêmes données.

Un autre écueil pourrait être le manque de données. Même si nous travaillons sur un jeu de données contenant 3,7 millions de molécules (REF DONNES), il s'agit peut-être d'une quantité insuffisante pour approximer correctement une fonction aussi complexe. De même, il est possible qu'il manque certains descripteurs des molécules en entrée des modèles.

Enfin, il est possible que le problème soit lié à notre méthodologie, et notamment au fait que l'on génère un jeu d'entraînement en ajoutant du bruit sur les données à prédire. Peut-être la tâche de prédiction est-elle impossible à réaliser à cause du caractère aléatoire et par définition imprévisible du bruit gaussien, même si on peut raisonnablement imaginer que si un modèle est capable de prédire une géométrie convergée, il est capable de soustraire la géométrie convergée d'une géométrie bruitée.

## Chapitre 7

# Conclusion

# Bibliographie

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow : Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] Scikit-learn : Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011
- [3] K. Hornik, M. Stinchcombe, and H. White, Multi-layer feedforward networks are universal approximators, preprint, 1988.

## Annexes

## Annexe A

# Représentations graphiques des prédictions des modèles *DIST\_REL\_C*

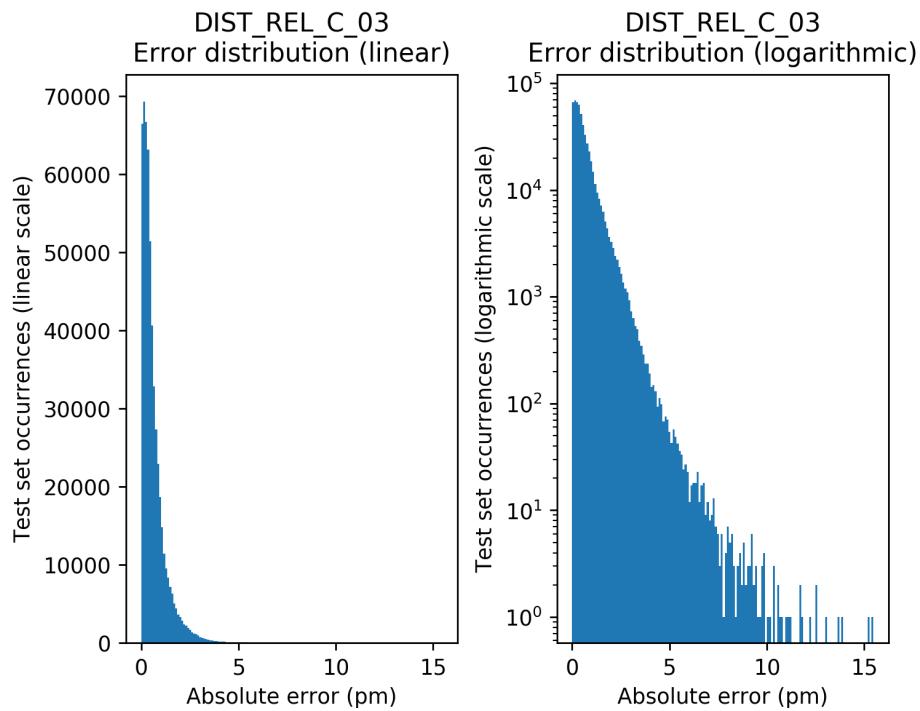


FIGURE A.1 – Distribution des erreurs du modèle *DIST\_REL\_C\_03*

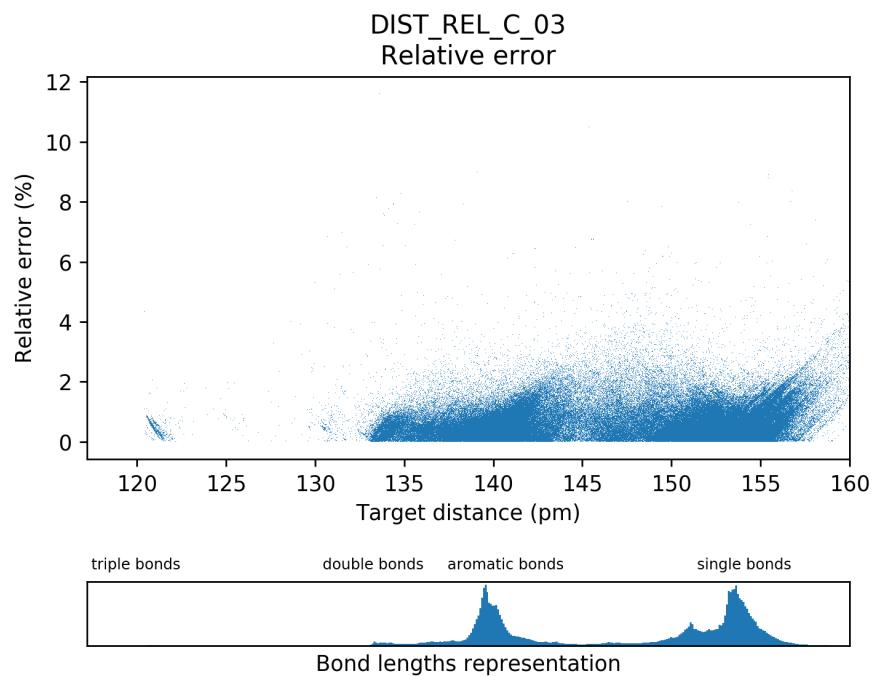


FIGURE A.2 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_C\_03*

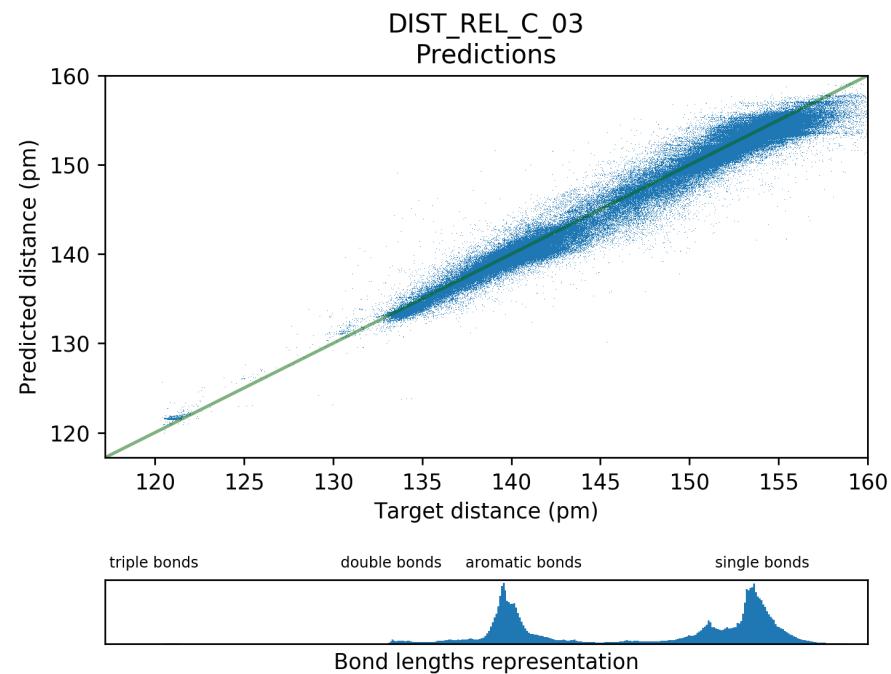


FIGURE A.3 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_C\_03*

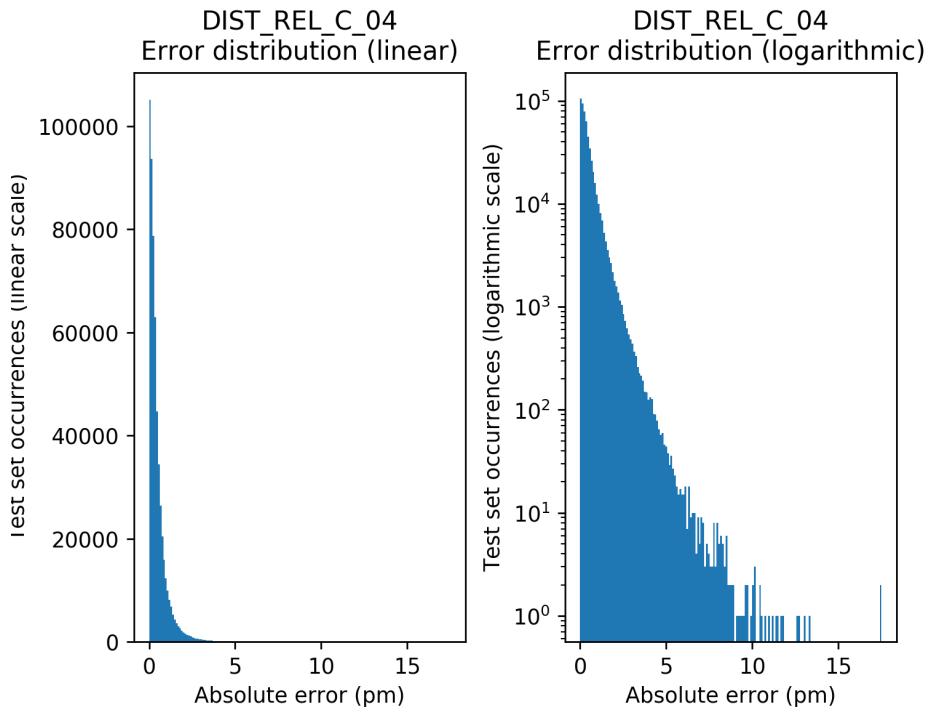


FIGURE A.4 – Distribution des erreurs du modèle *DIST\_REL\_C\_04*

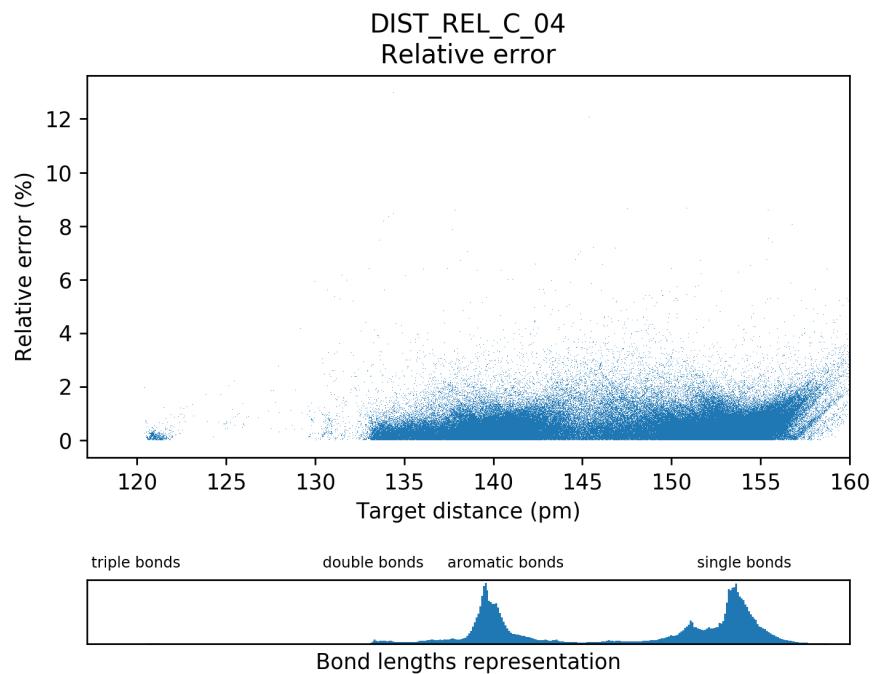


FIGURE A.5 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_C\_04*

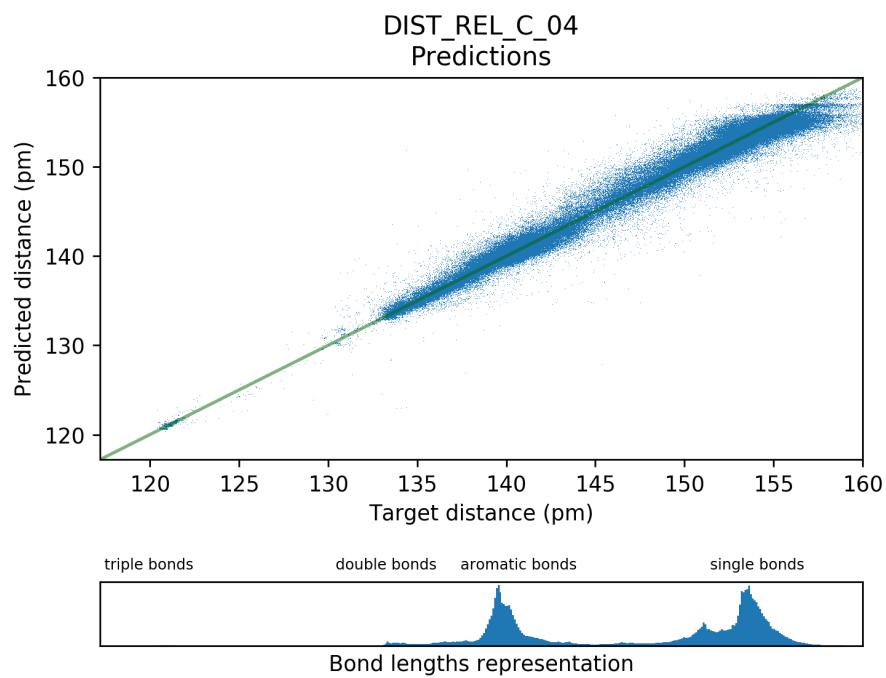


FIGURE A.6 – Prédictions en fonction des cibles pour le modèle *DIST\_REL\_C\_04*

## Annexe B

# Représentations graphiques des prédictions des modèles *DIST\_REL\_XY*

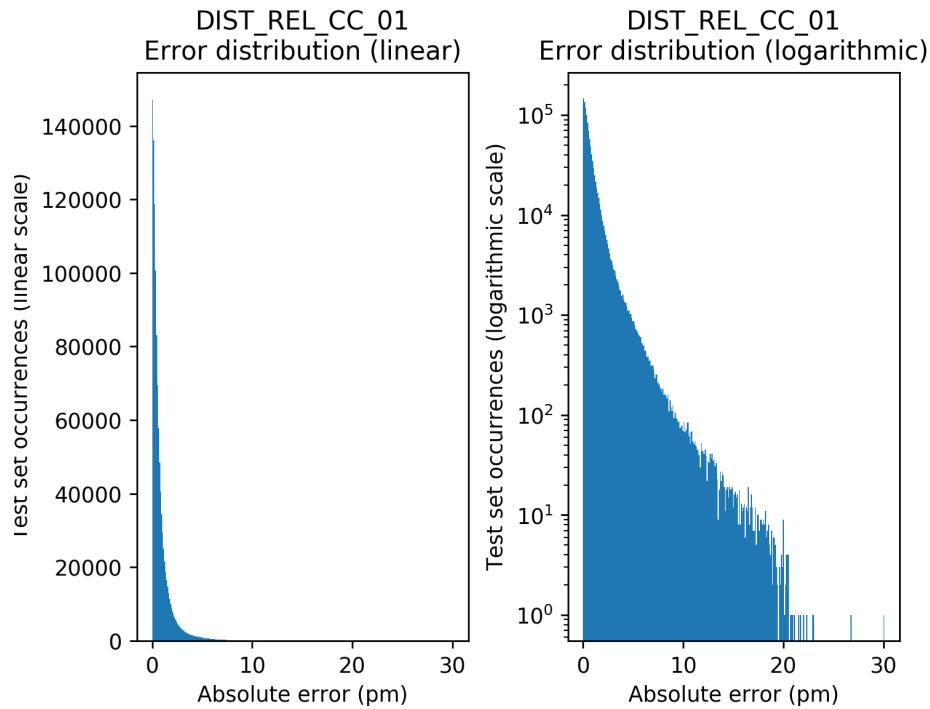


FIGURE B.1 – Distribution des erreurs du modèle *DIST\_REL\_CC\_01*

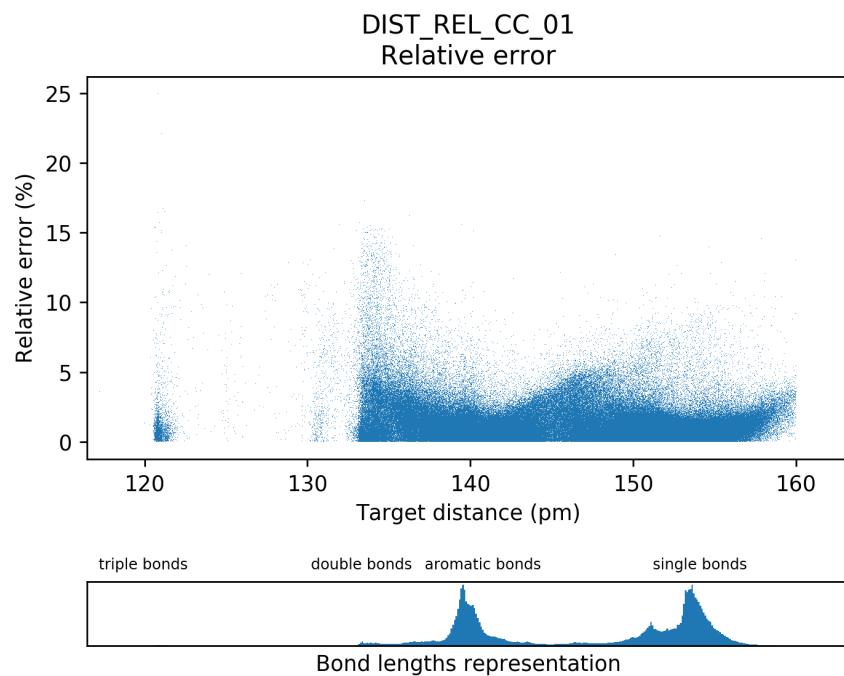


FIGURE B.2 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_CC\_01*

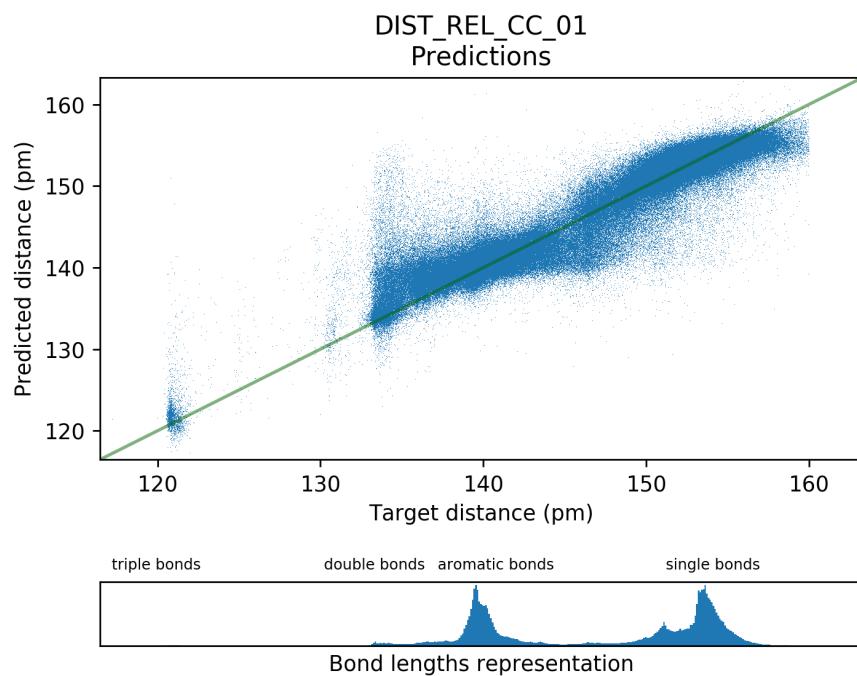


FIGURE B.3 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_CC\_01*

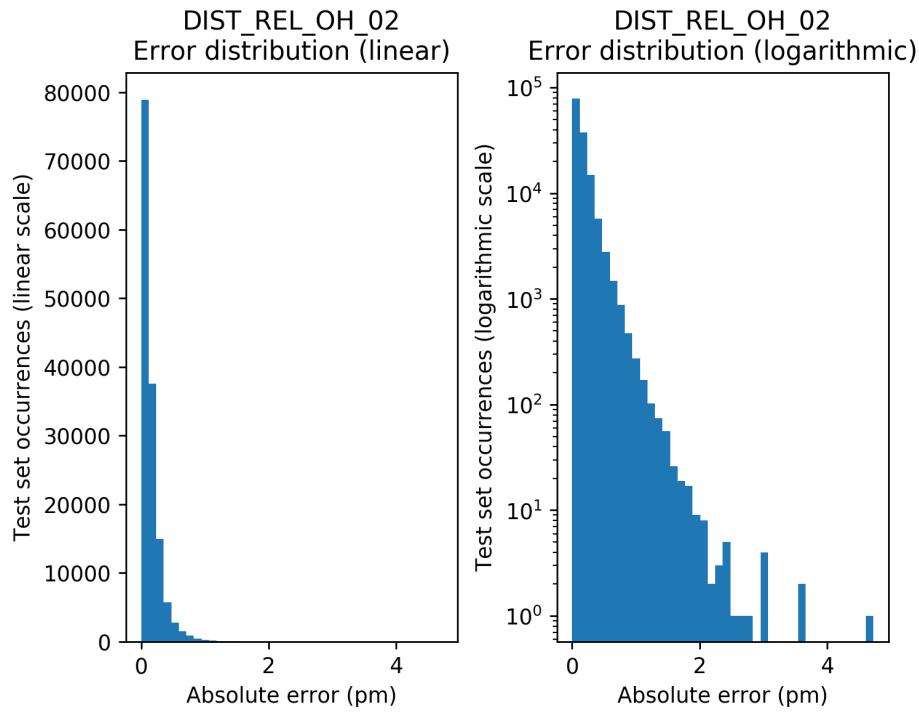


FIGURE B.4 – Distribution des erreurs du modèle *DIST\_REL\_OH\_02*

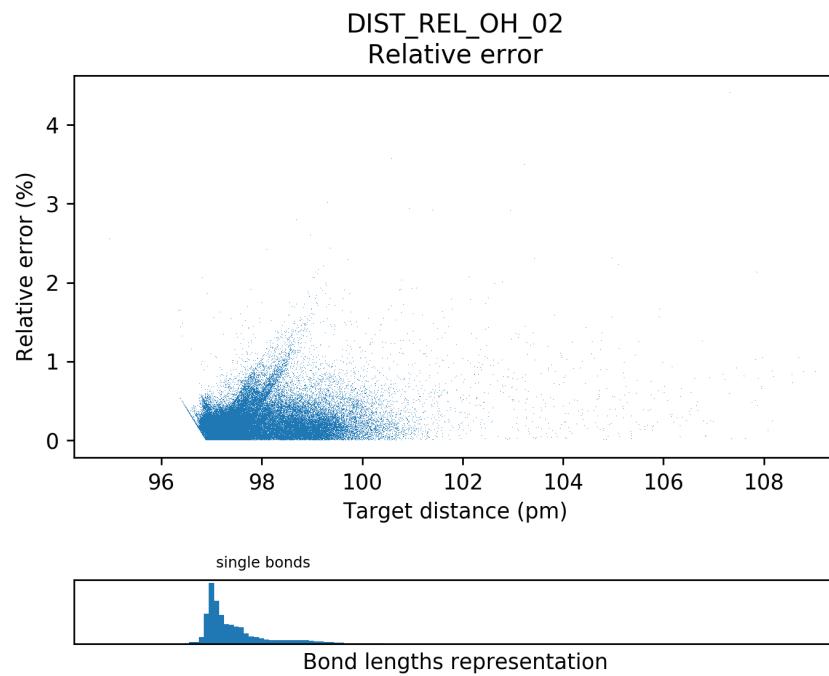


FIGURE B.5 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_OH\_02*

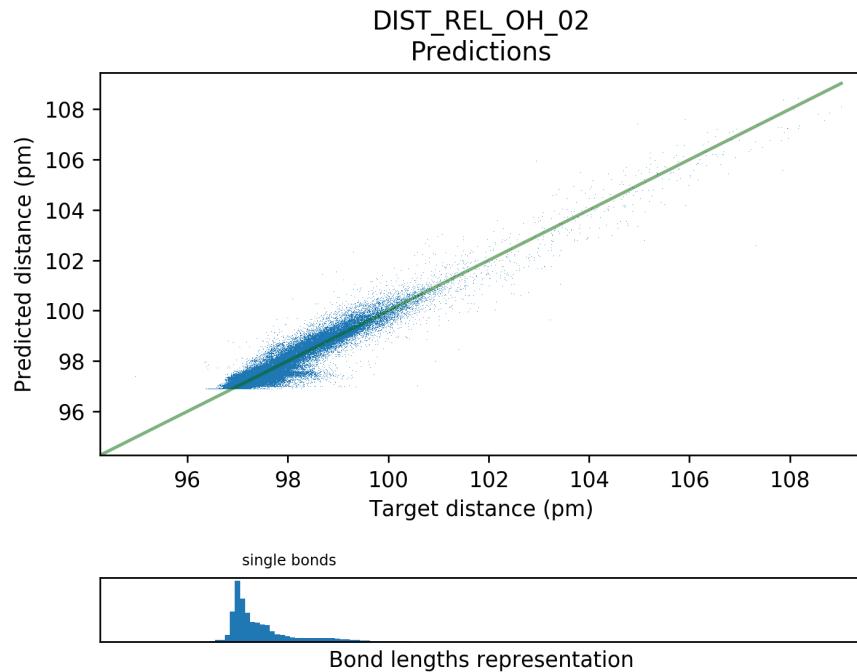


FIGURE B.6 – Prédictions en fonction des cibles pour le modèle *DIST\_REL\_OH\_02*

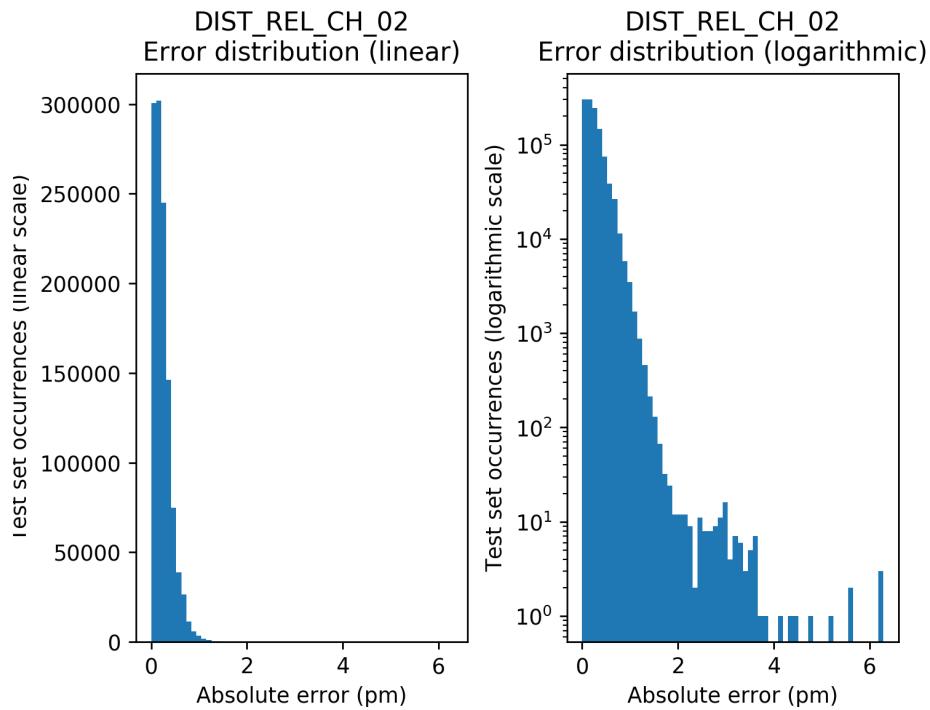


FIGURE B.7 – Distribution des erreurs du modèle *DIST\_REL\_CH\_02*

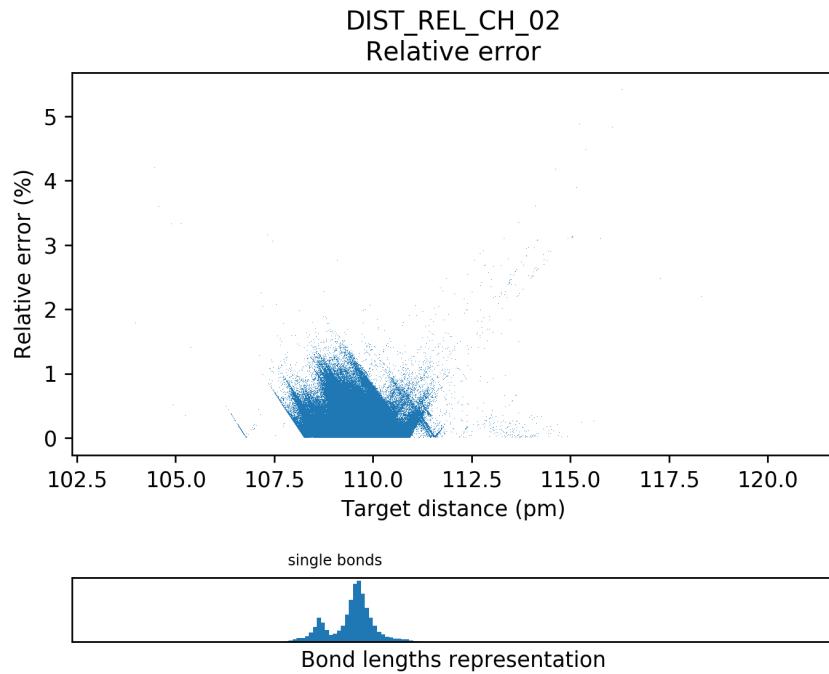


FIGURE B.8 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_CH\_02*

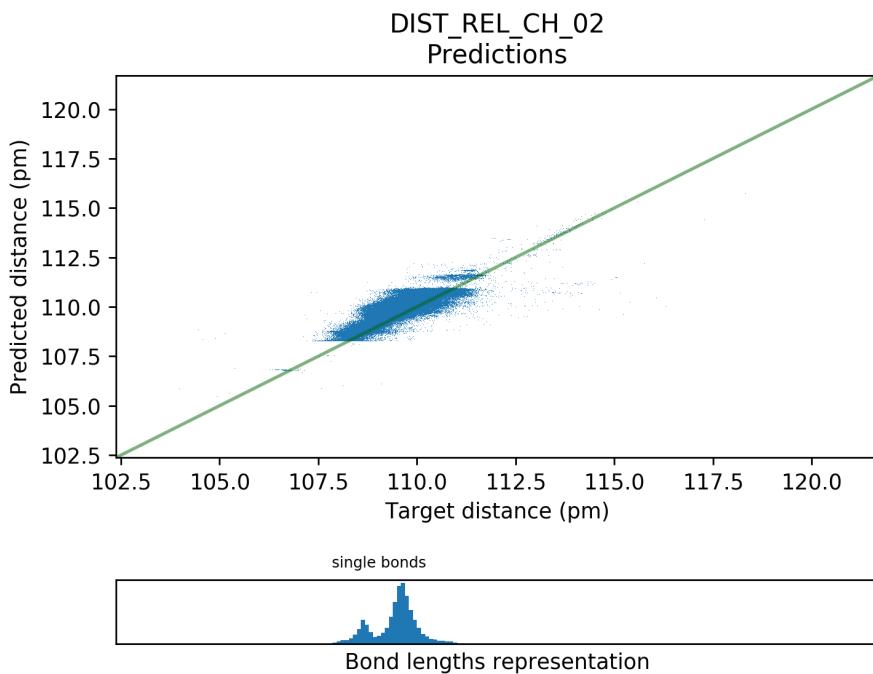


FIGURE B.9 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_OH\_02*

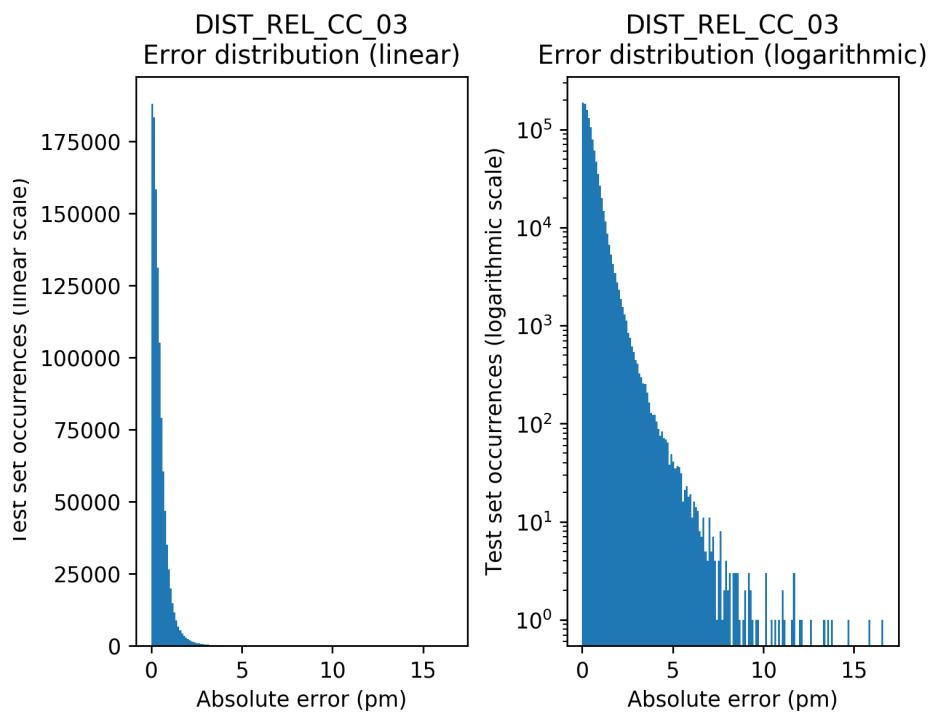


FIGURE B.10 – Distribution des erreurs du modèle *DIST\_REL\_CC\_03*

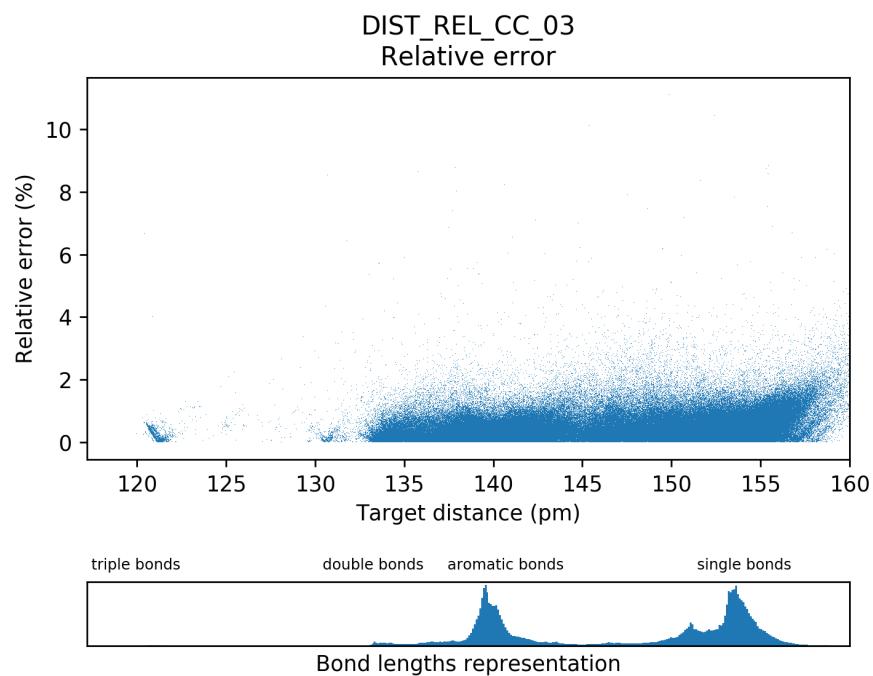


FIGURE B.11 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_CC\_03*

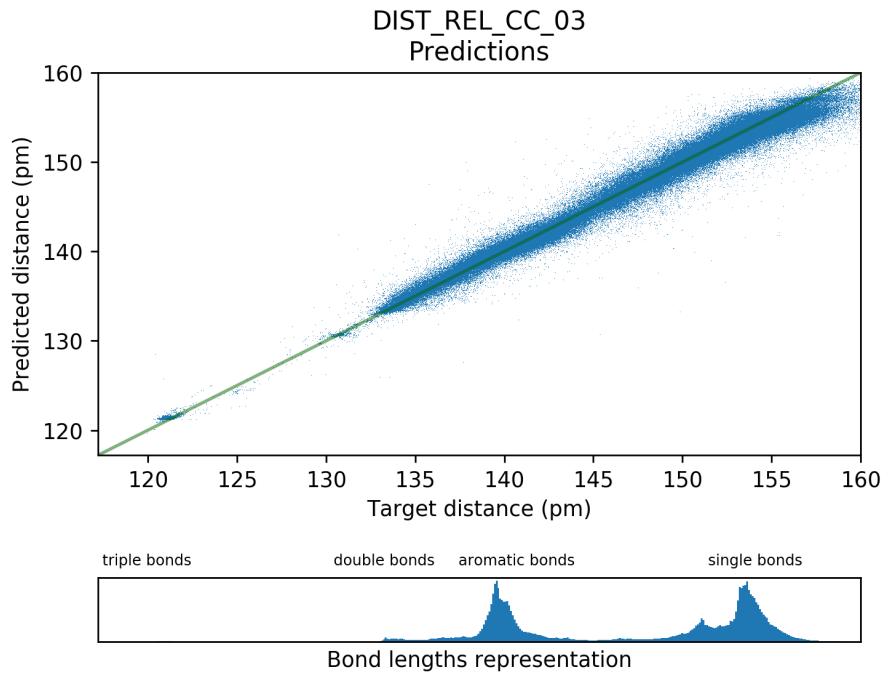


FIGURE B.12 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_CC\_03*

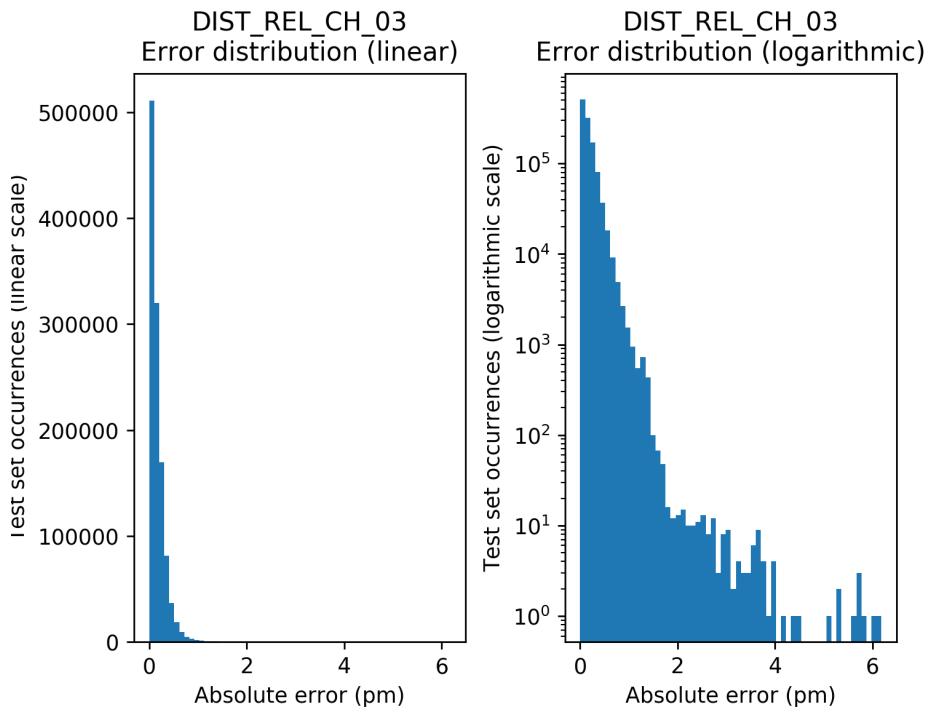


FIGURE B.13 – Distribution des erreurs du modèle *DIST\_REL\_CH\_03*

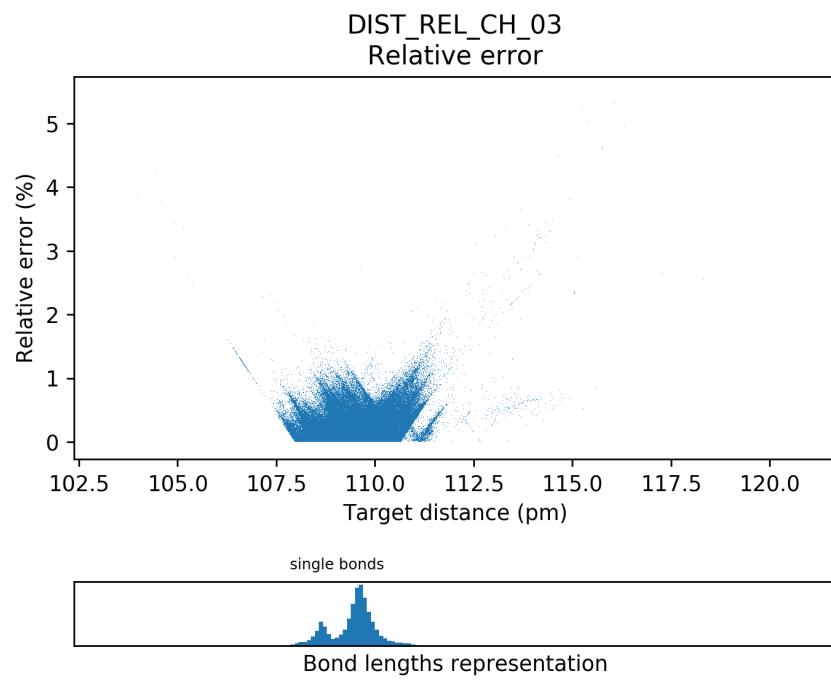


FIGURE B.14 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_CH\_03*

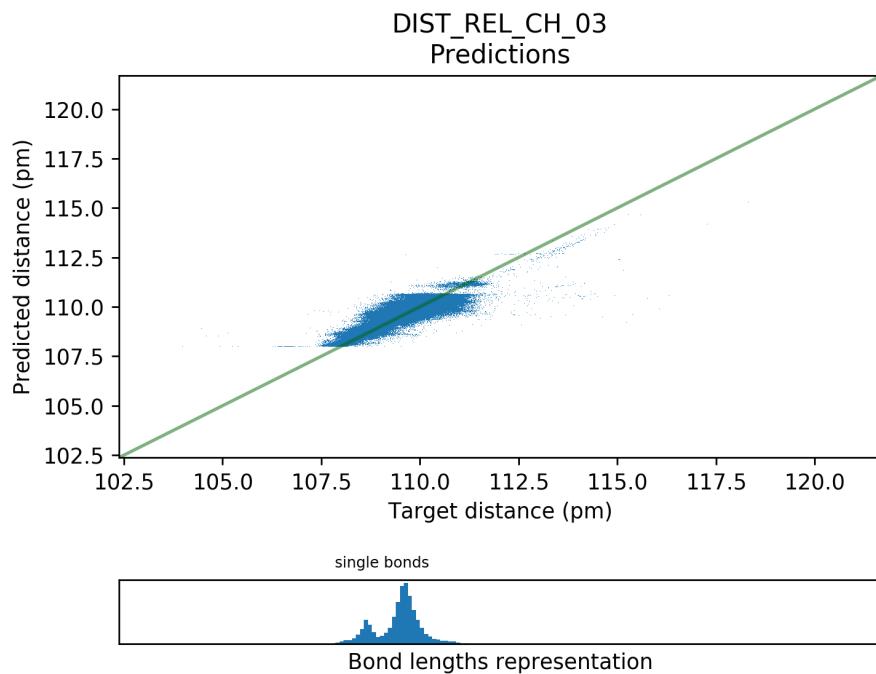


FIGURE B.15 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_CH\_03*

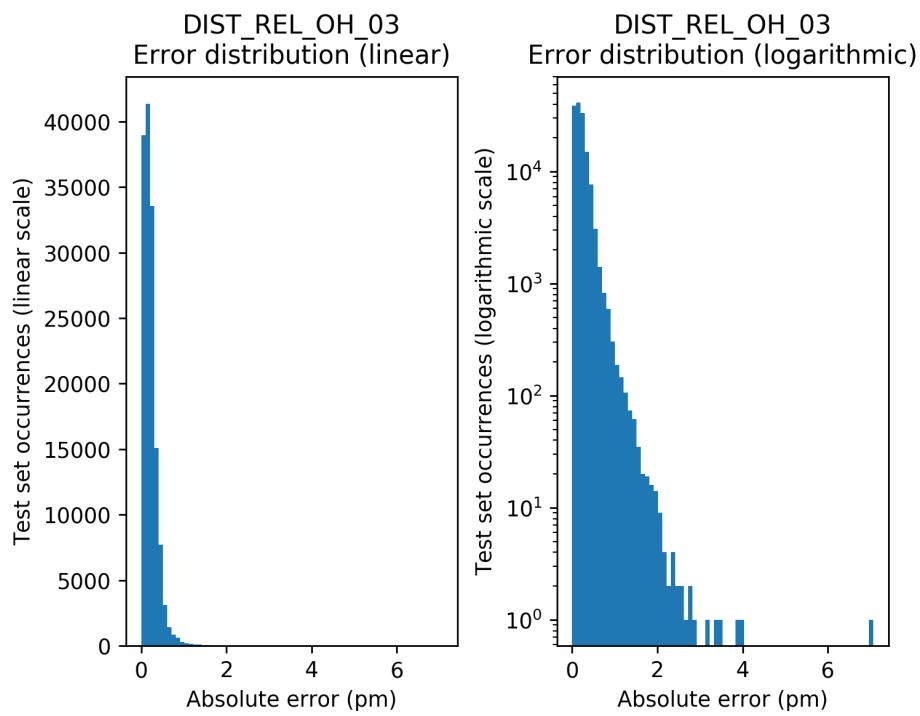


FIGURE B.16 – Distribution des erreurs du modèle *DIST\_REL\_OH\_03*

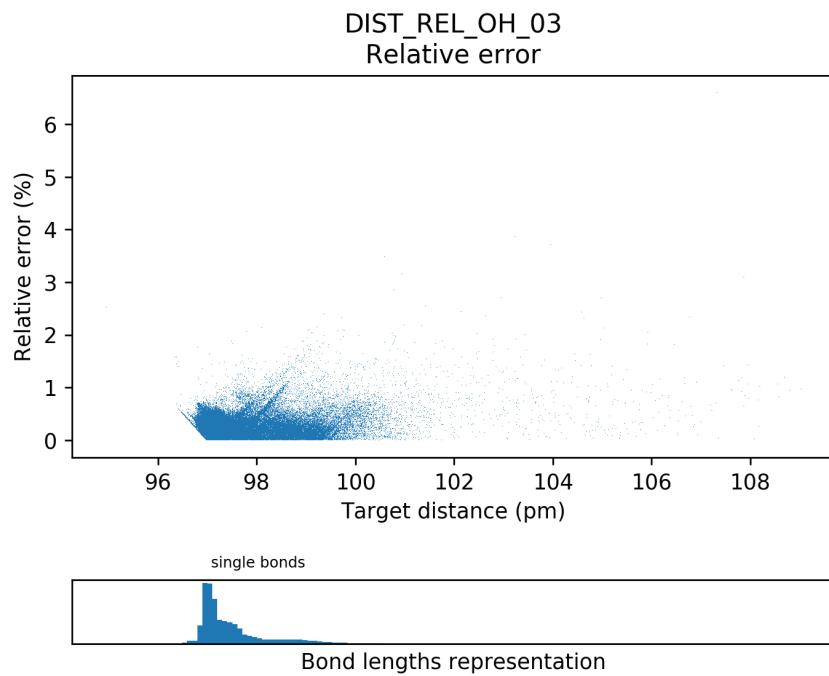


FIGURE B.17 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_OH\_03*

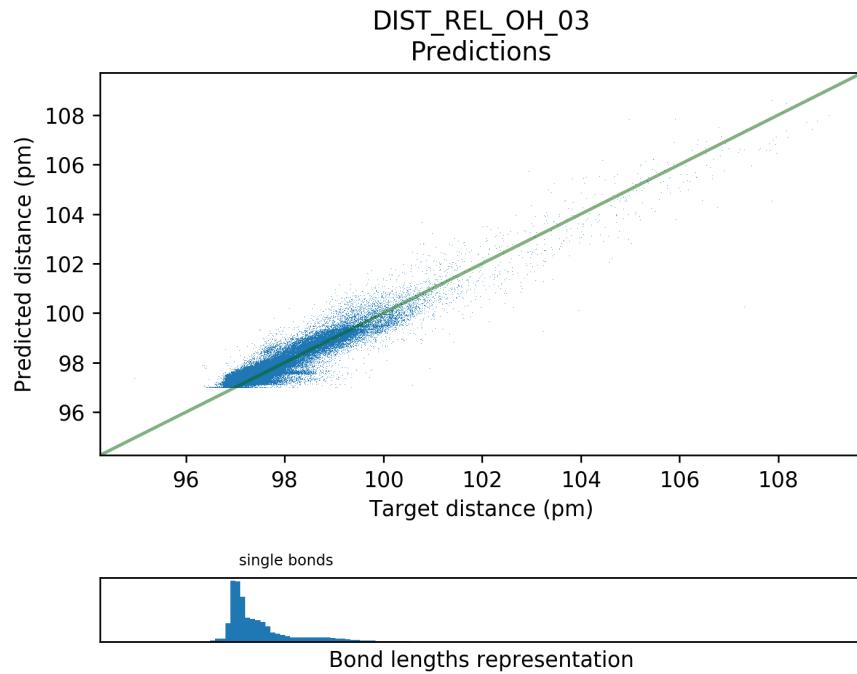


FIGURE B.18 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_OH\_03*

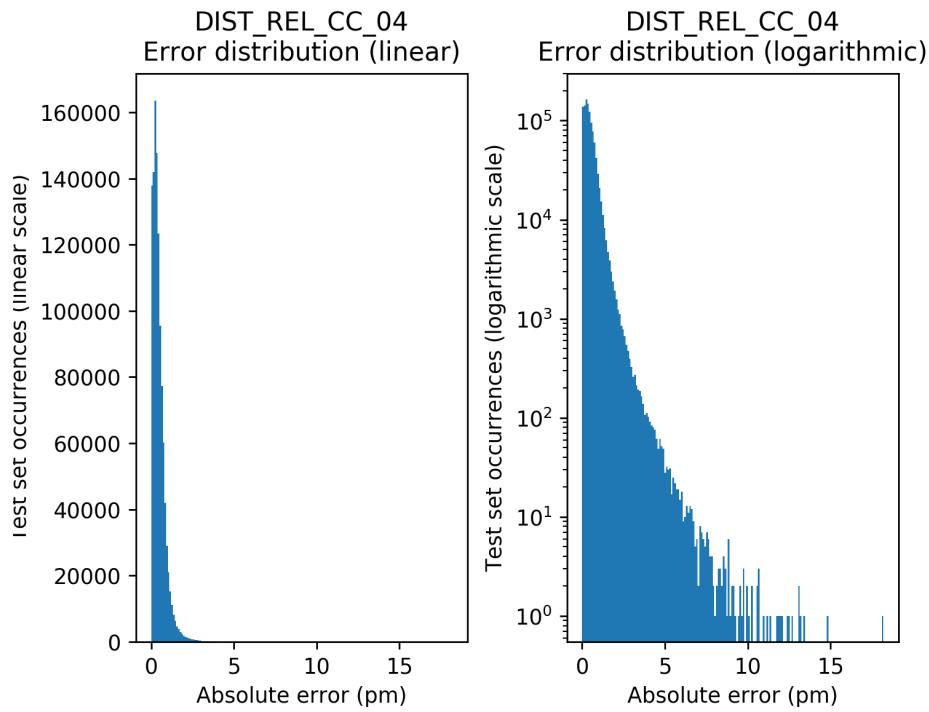


FIGURE B.19 – Distribution des erreurs du modèle *DIST\_REL\_CC\_04*

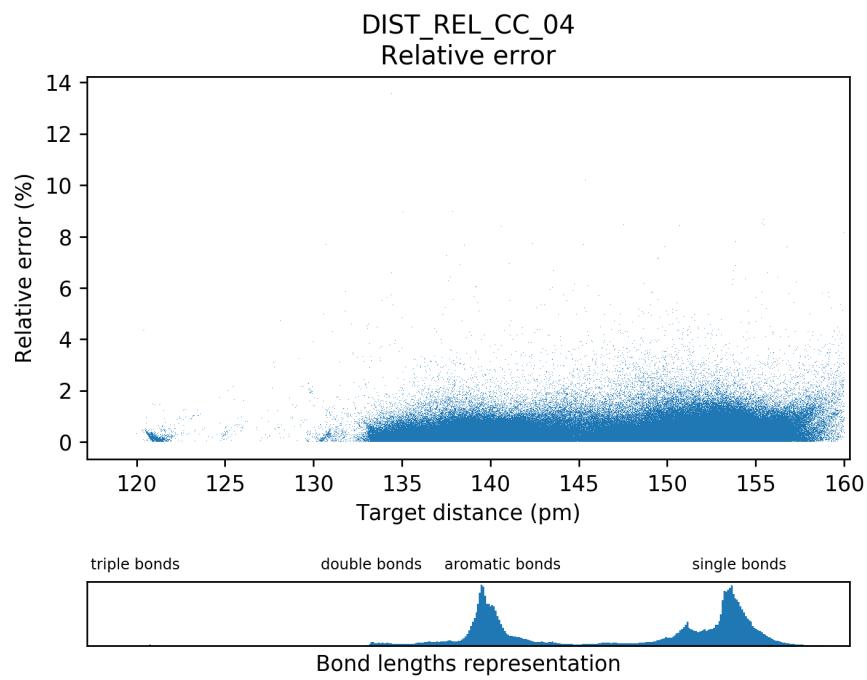


FIGURE B.20 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_CC\_04*

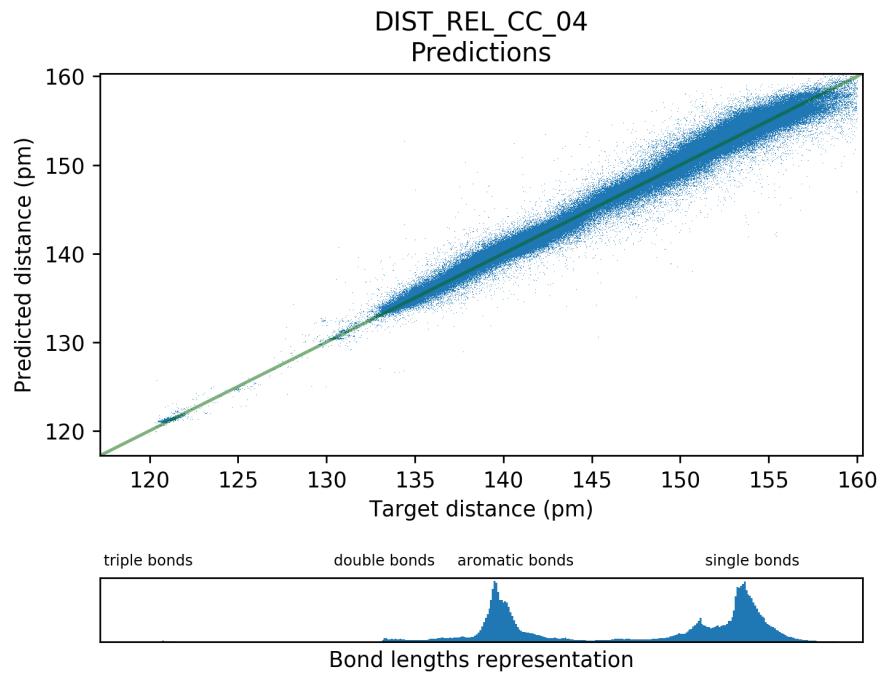


FIGURE B.21 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_CC\_04*

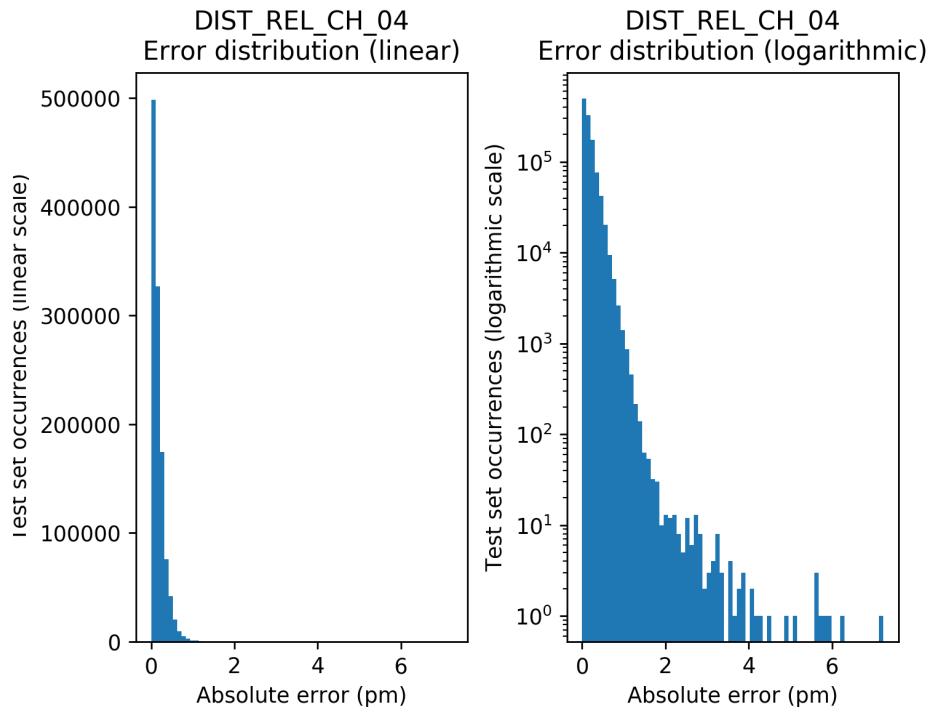


FIGURE B.22 – Distribution des erreurs du modèle *DIST\_REL\_CH\_04*

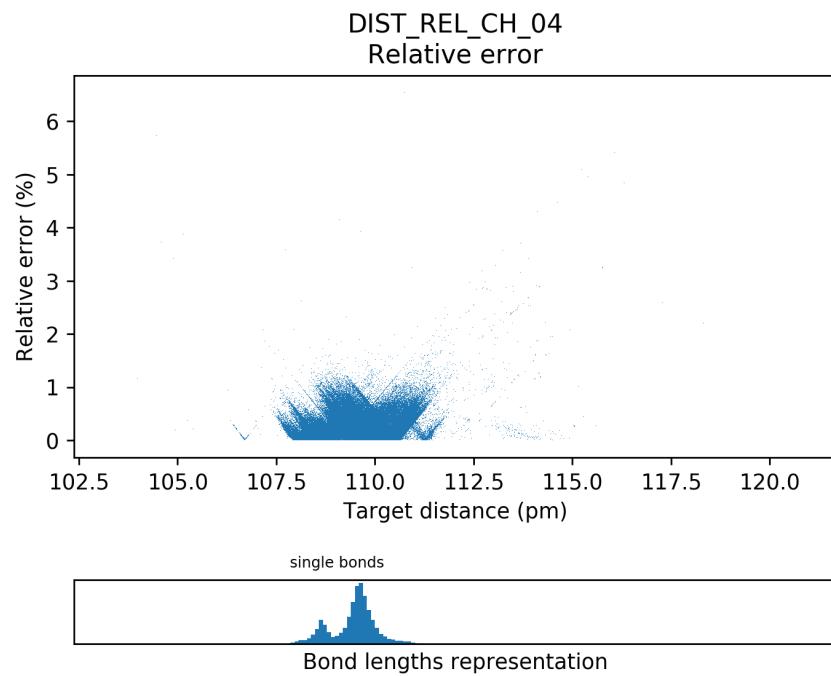


FIGURE B.23 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_CH\_04*

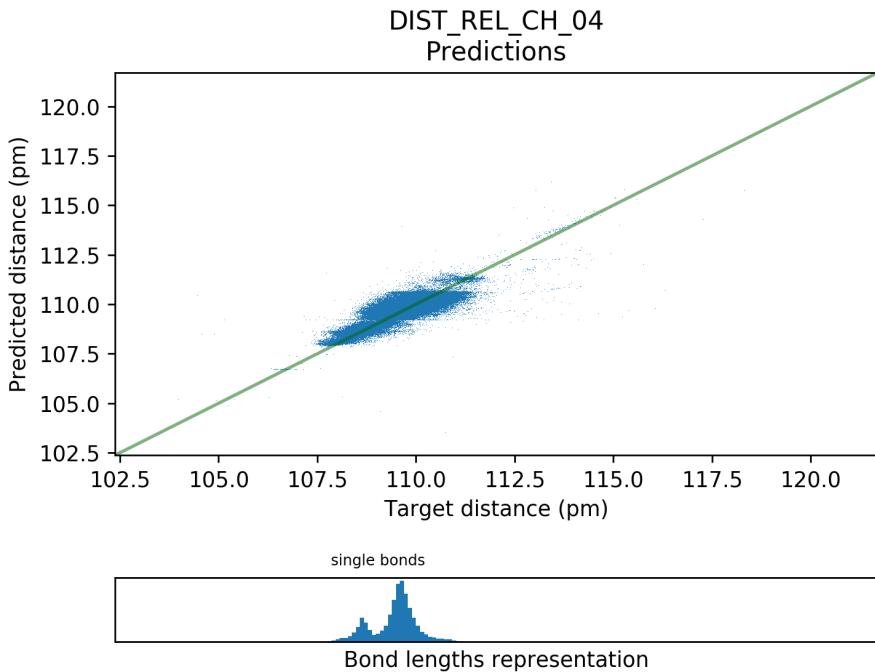


FIGURE B.24 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_CH\_04*

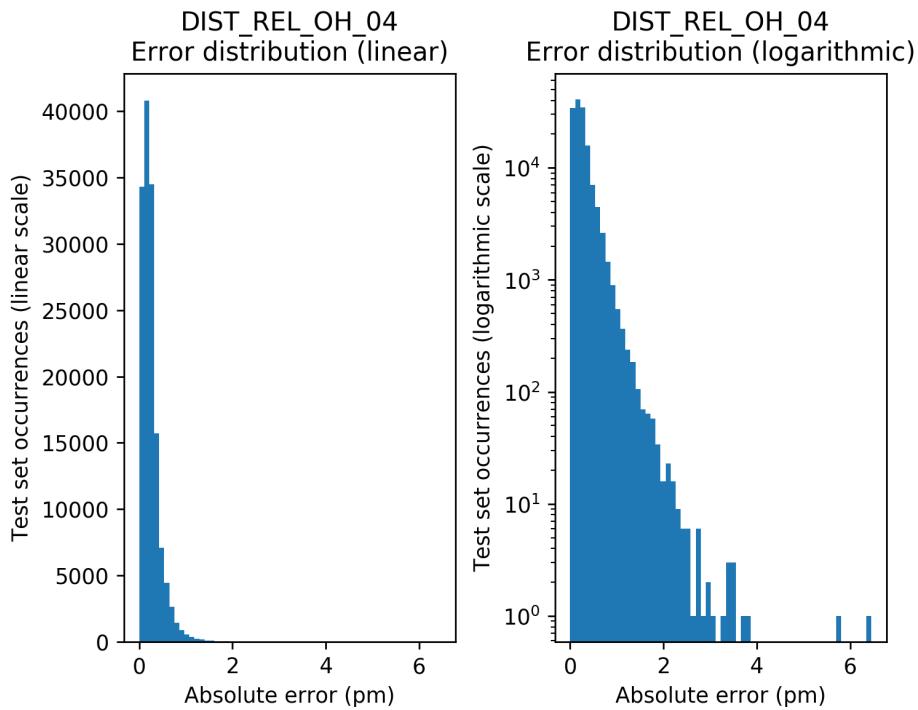


FIGURE B.25 – Distribution des erreurs du modèle *DIST\_REL\_OH\_04*

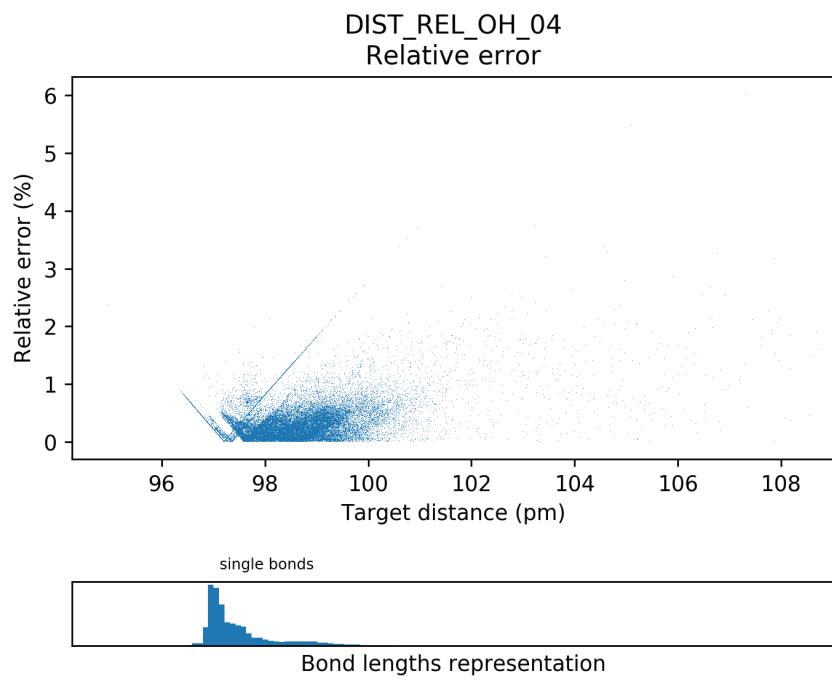


FIGURE B.26 – Erreur en fonction des cibles pour le modèle *DIST\_REL\_OH\_04*

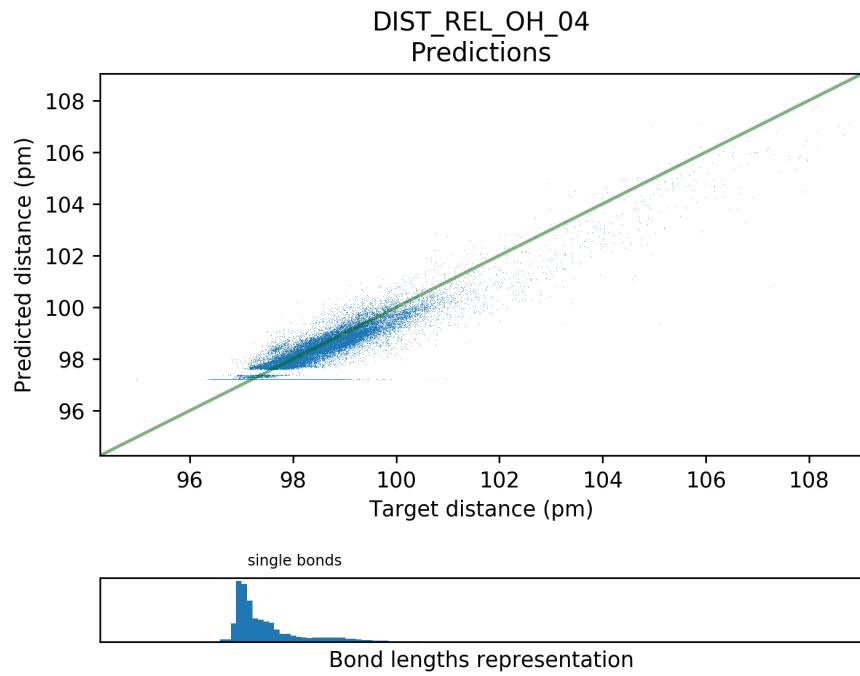


FIGURE B.27 – Prédiction en fonction des cibles pour le modèle *DIST\_REL\_OH\_04*

## Annexe C

# Résultats de la recherche par quadrillage du modèle KRR

```
0.935 (+/-0.007) for {'coef0': 1, 'kernel': 'linear', 'degree': 1, 'gamma': None, 'alpha': 0.1}
0.726 (+/-0.292) for {'coef0': 1, 'kernel': 'linear', 'degree': 1, 'gamma': None, 'alpha': 0.01}
0.615 (+/-0.495) for {'coef0': 1, 'kernel': 'linear', 'degree': 1, 'gamma': None, 'alpha': 0.001}
0.935 (+/-0.011) for {'coef0': 1, 'kernel': 'poly', 'degree': 2, 'gamma': None, 'alpha': 0.1}
0.817 (+/-0.285) for {'coef0': 1, 'kernel': 'poly', 'degree': 6, 'gamma': None, 'alpha': 0.1}
0.934 (+/-0.012) for {'coef0': 0.5, 'kernel': 'poly', 'degree': 2, 'gamma': None, 'alpha': 0.1}
0.586 (+/-0.772) for {'coef0': 0.5, 'kernel': 'poly', 'degree': 6, 'gamma': None, 'alpha': 0.1}
0.937 (+/-0.010) for {'coef0': 2, 'kernel': 'poly', 'degree': 2, 'gamma': None, 'alpha': 0.1}
0.899 (+/-0.124) for {'coef0': 2, 'kernel': 'poly', 'degree': 6, 'gamma': None, 'alpha': 0.1}
0.951 (+/-0.006) for {'coef0': 1, 'kernel': 'poly', 'degree': 2, 'gamma': None, 'alpha': 0.01}
0.760 (+/-0.384) for {'coef0': 1, 'kernel': 'poly', 'degree': 6, 'gamma': None, 'alpha': 0.01}
0.950 (+/-0.007) for {'coef0': 0.5, 'kernel': 'poly', 'degree': 2, 'gamma': None, 'alpha': 0.01}
0.703 (+/-0.312) for {'coef0': 0.5, 'kernel': 'poly', 'degree': 6, 'gamma': None, 'alpha': 0.01}
0.951 (+/-0.006) for {'coef0': 2, 'kernel': 'poly', 'degree': 2, 'gamma': None, 'alpha': 0.01}
0.909 (+/-0.107) for {'coef0': 2, 'kernel': 'poly', 'degree': 6, 'gamma': None, 'alpha': 0.01}
0.917 (+/-0.049) for {'coef0': 1, 'kernel': 'poly', 'degree': 2, 'gamma': None, 'alpha': 0.001}
0.742 (+/-0.394) for {'coef0': 1, 'kernel': 'poly', 'degree': 6, 'gamma': None, 'alpha': 0.001}
0.942 (+/-0.004) for {'coef0': 0.5, 'kernel': 'poly', 'degree': 2, 'gamma': None, 'alpha': 0.001}
0.699 (+/-0.330) for {'coef0': 0.5, 'kernel': 'poly', 'degree': 6, 'gamma': None, 'alpha': 0.001}
0.935 (+/-0.012) for {'coef0': 2, 'kernel': 'poly', 'degree': 2, 'gamma': None, 'alpha': 0.001}
0.895 (+/-0.146) for {'coef0': 2, 'kernel': 'poly', 'degree': 6, 'gamma': None, 'alpha': 0.001}
Best score: 0.951
Best parameters set:
{'kernel_params': None, 'kernel': 'poly', 'degree': 2, 'gamma': None, 'alpha': 0.01, 'coef0': 1}
```

## Annexe D

### Paramètres des modèles *DELTA\_DIST\_+H*

Modèle	Tailles molécules	Repr. géom. entrée	Repr. géom. sortie	Numéros atomiques	Masses atomiques	Distances inter at. fictifs	Fonction de coût	Profondeur	Largeur	Taille entrée	Bruit
DELTA_DIST+H_01	0 - 200	Matr. dist. pts. fixes	Matr. dist. pts. fixes	Non	Oui	Non	RMSE partiel/total	4	8650	1000	+/++
DELTA_DIST+H_02	0 - 200	Matr. dist. pts. fixes	Matr. dist. pts. fixes	Non	Oui	Oui	RMSE partiel	4	8650	1020	+
DELTA_DIST+H_03	0 - 200	Matr. dist. pts. fixes + Matr. red. dist. interat.	Matr. dist. pts. fixes	Non	Oui	Non	RMSE partiel	3	9000	1800	++
DELTA_DIST+H_04	0 - 200	Matr. dist. pts. fixes + Matr. red. dist. interat.	Matr. red. dist. interat.	Non	Oui	Non	RMSE partiel	3	9000	1800	++
DELTA_DIST+H_05	2 - 60	Matr. dist. pts. fixes	Matr. dist. pts. fixes	Oui	Oui	Non	RMSE partiel	3	360	360	++