




Universitat Autònoma de Barcelona

**ADVERTIMENT.** L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons:  [http://cat.creativecommons.org/?page\\_id=184](http://cat.creativecommons.org/?page_id=184)

**ADVERTENCIA.** El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons:  <http://es.creativecommons.org/blog/licencias/>

**WARNING.** The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license:  <https://creativecommons.org/licenses/?lang=en>



Universitat Autònoma de Barcelona

# Development and Application of a Computational Platform for Complex Molecular Design

A DISSERTATION SUBMITTED BY  
JAIME RODRÍGUEZ-GUERRA PEDREGAL

✂ DIRECTED BY  
PROF. DR. JEAN-DIDIER MARÉCHAL

IN FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
DOCTOR OF BIOTECHNOLOGY

TUTOR: PROF. DR. JORDI JOAN CAIRÓ BADILLO  
DEPARTMENT OF CHEMICAL, BIOLOGICAL AND ENVIRONMENTAL ENGINEERING  
UNIVERSITAT AUTÒNOMA DE BARCELONA  
JULY 2018





Universitat Autònoma de Barcelona

# Development and Application of a Computational Platform for Complex Molecular Design

A DISSERTATION SUBMITTED BY



RECOMMENDED FOR ACCEPTANCE

BY ADVISOR

JAIME RODRÍGUEZ-GUERRA PEDREGAL

PROF. DR. JEAN-DIDIER MARÉCHAL

TUTOR: PROF. DR. JORDI JOAN CAIRÓ BADILLO  
DEPARTMENT OF CHEMICAL, BIOLOGICAL AND ENVIRONMENTAL ENGINEERING  
UNIVERSITAT AUTÒNOMA DE BARCELONA  
JULY 2018



©2018 – JAIME RODRÍGUEZ-GUERRA PEDREGAL  
LICENSED AS CREATIVE COMMONS BY-NC-ND  
ATTRIBUTION-NONCOMMERCIAL-NO DERIVS



*In the beginning, there was nothing.  
And God said «Let there be light».  
And there was light.  
There was still nothing,  
but you could see it a lot better.*

*—Woody Allen.*





# Development and Application of a Computational Platform for Complex Molecular Design

by Jaime Rodríguez-Guerra Pedregal

## ABSTRACT

In this dissertation, a series of novel computational modeling tools is reported. All of them have been written in Python and include: (1) GaudiMM, (2) Tangram, and (3) a collection of command-line applications. This Ph.D. demonstrates the power of this unique high-level language, particularly in software development for molecular modeling.

1. GaudiMM allows to build and refine chemobiological structures through a multi-objective genetic algorithm. It features a modular, extensible architecture that can be applied to diverse molecular modeling exercises, depending on the modules chosen.
2. Tangram is a collection of graphical interfaces for UCSF Chimera. Some of these extensions provide interactive methods for setting up calculations in external programs, like Quantum Mechanics in Gaussian or Molecular Dynamics in OpenMM. Others rely on the interactive 3D viewer to depict properties of molecular structures as calculated previously in other software, turning UCSF Chimera into an even more versatile analysis tool.
3. A variety of command-line tools has been also developed along GaudiMM and Tangram. They are mainly concerned with optimizing common workflows in molecular modeling, like running GPU-accelerated Molecular Dynamics simulations (OMMProtocol), extending the force fields used in QM/MM approaches (Garleek), or automating the elaboration of Supporting Information documents for computational chemistry calculations (ESIgen).

To prove their usage and applicability in molecular modeling, a series of illustrative cases will be described in detail. These include toy examples that showcase the potentiality of GaudiMM —some of them unreachable with standard methodologies—, like siderophore chelation, standard and exotic docking protocols, ligand design and metal binding site prediction.



# Official Acknowledgments

THE AUTHOR IS THANKFUL for the funding support received from several public institutions. His 3-year Ph.D. scholarship was possible thanks to the Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR, Generalitat de Catalunya) and the European Social Fund (2015-FI-B-00768, 2016-FI-B1-00069 and 2017-FI-B2-00168). He also acknowledges the support received by the projects CTQ-2014-54071-P and CTQ-2017-87889-P (MINECO, Government of Spain), 2014-SGR-989 and 2017-SGR-1323 (Generalitat de Catalunya), and CMST COST action CM1306.

The final period of his Ph.D. studies was possible thanks to the 6-month stay with Prof. Dr. Feliu Maseras' group, at the Institut Català d'Investigació Química (ICIQ), Tarragona.



# Contents

PREFACE	19
1 INTRODUCTION	23
1.1 Molecular modeling: accuracy vs sampling	24
1.2 Multiscale and integrative modeling	27
1.2.1 Dealing with software fragmentation	28
1.2.2 The ecosystem of integrative platforms	30
1.2.3 The role of scripting in the integration of software projects	33
1.3 Modeling with scarce data: abusing modular approaches	34
2 MATERIALS & METHODS	37
2.1 Origins of molecular modeling	37
2.2 Energy description	39
2.2.1 Quantum Mechanics (QM)	39
2.2.2 Force fields: Molecular Mechanics, Molecular Dynamics and Metadynamics	41
2.2.3 QM/MM	42
2.2.4 Coarse-grained modeling	43
2.3 Conformational sampling	44
2.3.1 Normal Modes Analysis	44
2.3.2 Recognition processes	44
2.4 Beyond cartesian coordinates: navigating the chemical space	45
2.5 Building models from scratch	49
2.6 Optimization methods	50
2.6.1 Steepest descent and conjugate gradient	52
2.6.2 The Newton and quasi-Newton algorithms	52
2.6.3 Heuristic and meta-heuristic methods	54
2.6.4 Machine learning	55
2.7 Multi-objective optimization	56
3 OBJECTIVES	59
4 GAUDIMM	61
4.1 Algorithmic details	62
4.2 Implementation	63
4.2.1 Of individuals and genes: the exploration stage	63
4.2.2 Of environments and objectives: the evaluation stage	67
4.2.3 Of tournaments and trade-offs: the selection stage	69
4.2.4 The code behind: Python as glue	70
4.3 Usage: from recipes to molecular modeling tasks	71
4.3.1 Tutorial: Obtaining a cyclic alkane	72

4.4	Analyzing the results of multi-objective optimization . . . . .	74
4.5	Conclusions & Further work . . . . .	75
5	PYTHON-BASED MOLECULAR MODELING WORKFLOWS . . . . .	77
5.1	Implementation of a common interactive canvas: Tangram . . . . .	77
5.1.1	Multiscale modeling with Tangram . . . . .	80
5.2	Optimizing workflows from the command-line . . . . .	84
5.2.1	PyChimera . . . . .	84
5.2.2	GPU-accelerated Molecular Dynamics, the easy way: OMMProtocol . . . . .	86
5.2.3	Extended QM/MM for Gaussian: Garleek . . . . .	89
5.2.4	Automated Electronic Supporting Information Generator: ESIgen . . . . .	91
5.2.5	Easy MECP calculations . . . . .	94
5.3	Conclusions & Further work . . . . .	94
6	BENCHMARK & APPLICATION . . . . .	97
6.1	GaudiMM as a versatile molecular modeling tool . . . . .	97
6.1.1	From standard to more exotic dockings . . . . .	97
6.1.2	Metal ions: Organization & binding site prediction . . . . .	103
6.2	Multiscale modeling of multivalent enzyme inhibitors . . . . .	110
6.2.1	Introduction to multivalent enzyme inhibition . . . . .	110
6.2.2	Experimental results . . . . .	111
6.2.3	Computational approaches towards an explanation . . . . .	114
6.2.4	Discussion & Further work . . . . .	120
6.3	Final conclusions . . . . .	121
7	GENERAL CONCLUSIONS . . . . .	123
	EPILOG . . . . .	127
	APPENDIX A PERSPECTIVES FOR MOLECULAR MODELING . . . . .	129
A.1	The impact of molecular modeling . . . . .	129
A.2	What the next generation will bring to the table . . . . .	131
	APPENDIX B GAUDI MM AS AN EDUCATIONAL TOOL: UNDERGOING DEVELOPMENTS . . . . .	135
B.1	Navigating the chemical space . . . . .	135
B.2	Finding ligand binding pathways . . . . .	135
	APPENDIX C LIVING WITH METAL IONS IN MOLECULAR MODELING . . . . .	137
C.1	Quantum Mechanics . . . . .	137
C.2	Molecular Mechanics . . . . .	138
C.3	Lower levels of theory . . . . .	139
	APPENDIX D TANGRAM EXTENSIONS FOR ANALYSIS . . . . .	141
D.1	Interaction analysis . . . . .	141
D.1.1	GaudiView . . . . .	141
D.1.2	NCIPlotGUI . . . . .	142
D.1.3	PLIPGUI . . . . .	142
D.2	Structure analysis . . . . .	144
D.2.1	3D-SNFG . . . . .	144
D.2.2	BondOrder . . . . .	145
D.2.3	OrbiTraj . . . . .	145

D.2.4	PoPMuSiCGUI . . . . .	145
D.2.5	PropKaGUI . . . . .	146
D.2.6	SubAlign . . . . .	147

BIBLIOGRAPHY		163
--------------	--	-----





# List of Figures

1	Dorothy M. Crowfoot's 1945 Penicillin model . . . . .	20
1.1	Accuracy vs accessible space in multiscale modeling . . . . .	25
1.2	Example of a multiscale protocol . . . . .	28
1.3	Proliferation of standards . . . . .	29
2.1	Chemobiological spaces . . . . .	46
2.2	SMILES notation . . . . .	48
2.3	Gradient descent vs Newton's optimization . . . . .	53
2.4	A Pareto front for two dimensions . . . . .	57
4.1	NSGA-II algorithm . . . . .	64
4.2	Mutation and crossover . . . . .	65
4.3	GaudiMM input example . . . . .	73
4.4	GaudiView . . . . .	75
5.1	Multiscale funnel . . . . .	79
5.2	Tangram QMSetup . . . . .	81
5.3	Tangram MMSetup . . . . .	82
5.4	Tangram DummyMetal . . . . .	83
5.5	Uncoupled software architecture . . . . .	85
5.6	Popularity of InsiliChem packages . . . . .	86
5.7	OMMProtocol input file structure . . . . .	88
5.8	Example results with OMMAnalyze . . . . .	89
5.9	ONIOM workflow with Garleek . . . . .	91
5.10	ESIgen 3D report . . . . .	93
6.1	Proposed Cu-containing phenanthroline cofactors . . . . .	101
6.2	Dimer of dimers in streptavidin . . . . .	102
6.3	Linker length optimization . . . . .	104
6.4	Peptide folding . . . . .	107
6.5	Coordination objective . . . . .	108
6.6	Siderophore folding . . . . .	110
6.7	Tested inhibitors against ScGH13 and TmGH1 . . . . .	112
6.8	Single di-ONJ inhibitor test . . . . .	115
6.9	Di-ONJ linker length optimization . . . . .	116
6.10	Dual di-ONJ interface meeting test . . . . .	118
A.1	Publication trends in molecular modeling . . . . .	130
A.2	Machine learning publication trends . . . . .	133
D.1	Tangram NCIPLOTGUI . . . . .	143
D.2	Tangram 3D-SNFG . . . . .	144

D.3 Tangram PoPMuSiC GUI . . . . . 146

## List of Tables

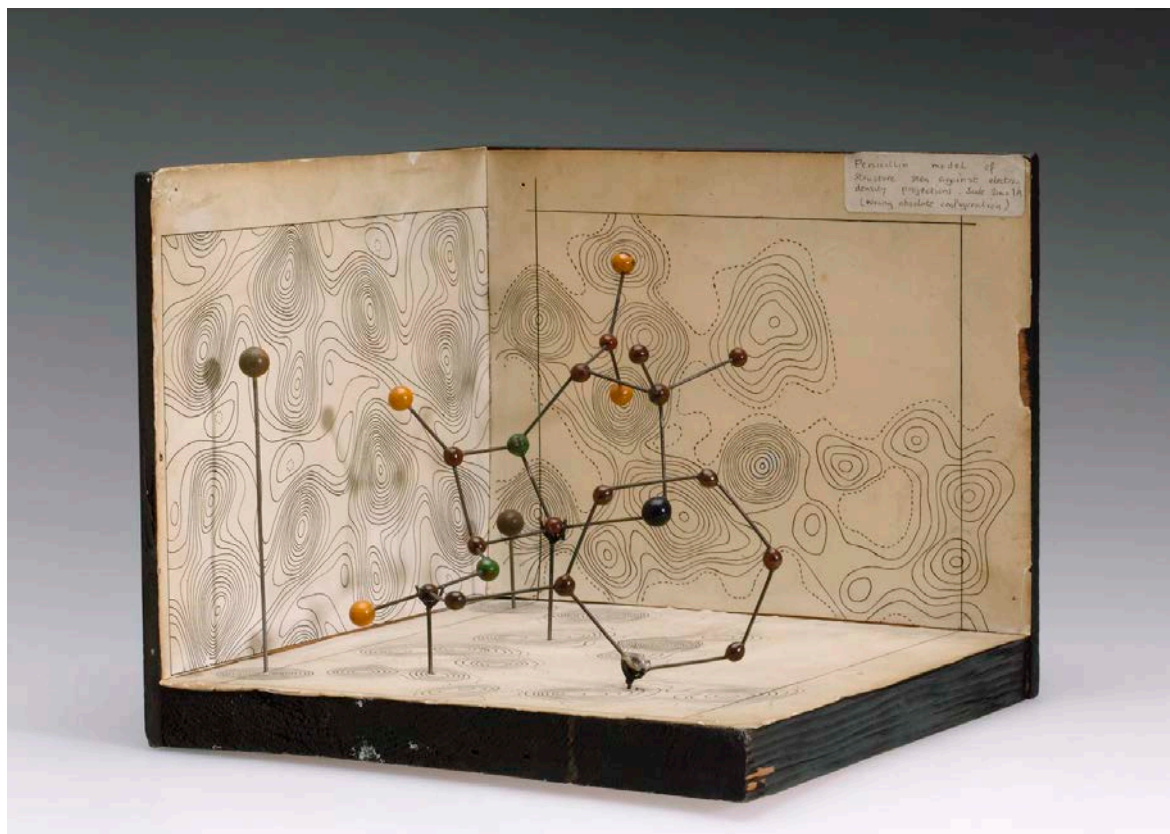
4.1	GaudiMM: technical datasheet . . . . .	62
4.2	List of genes implemented in GaudiMM . . . . .	67
4.3	List of objectives implemented in GaudiMM . . . . .	70
4.4	Final recipe for the cyclodecane example . . . . .	73
5.1	The full InsiliChem Molecular Modeling software suite . . . . .	78
5.2	Tangram Suite: Technical datasheet . . . . .	78
5.3	Full list of Tangram extensions . . . . .	80
5.4	PyChimera: Technical datasheet . . . . .	86
5.5	OMMProtocol: Technical datasheet . . . . .	87
5.6	Garleek: Technical datasheet . . . . .	90
5.7	ESigen: Technical datasheet . . . . .	92
5.8	EasyMECP: Technical datasheet . . . . .	94
6.1	Recipe applied in the docking benchmark . . . . .	98
6.2	Success rate of GaudiMM in a docking benchmark . . . . .	99
6.3	Recipe applied for the LmrR competitive docking calculations . . . . .	100
6.4	Recipe applied for the Streptavidin-dibiotin system . . . . .	103
6.5	Recipe applied for the Al(III)-amyloid complexes . . . . .	106
6.6	Recipe applied for the enterobactin exercise . . . . .	109
6.7	Glycosidases inhibition tests . . . . .	113
6.8	Single di-ONJ recipe . . . . .	115
6.9	Stretchable di-ONJ recipe . . . . .	117
6.10	Double di-ONJ recipe . . . . .	117
6.11	Pillar-5-ene recipe . . . . .	120



## Preface

**S**CIENTIFIC progress is tightly linked to human curiosity, a driving force that has brought us further and further every decade. The same impulses that thrust imagination into wondering what those lights in the sky were, kept us on Earth trying to figure out how far we can get splitting matter into pieces. A mountain can be reduced to rocks, stones, pebbles, sand, dust and... where do we stop? That question gave birth to the *atomism* philosophy in Ancient Greece, when the term for the modern word *atom* was coined: *ἄτομον* (*atomon, indivisible*). In atomism, all matter is composed of atoms and void. Those philosophical atoms, in all shapes and sizes, could collide with each other or hook together to form clusters resulting in their observable, macroscopic counterparts or *substances*.

*Atomism* was just a philosophical current trying to explain the world without any empirical observations to prove those hypotheses. It can be considered the first atomic model; albeit a useless one. This does not necessarily mean it is a bad model. Models are *simplifications of reality that can provide explanations and predictions of reproducible observations*. In this case, atomism failed to explain actual phenomena, but did satisfy the philosophical curiosity behind its inception. As a result, one can only assess the quality of a model in terms of its purpose: it will be valid as long as this is fulfilled (see fig. 1).



**Figure 1:** In the pre-computer era, models were built physically. Here depicted, Dorothy M. Crowfoot's 1945 penicillin model. Reproduced from UK's Science Museum.<sup>1</sup>

Since the clusters described by atomism do not provide useful predictions on molecules, new models have been described over the past decades, each replacing the previous one to address new conflicting observations: Dalton, Thomson, Rutherford, Bohr... During the past century, we have seen the atom acquiring unthought complexity: nuclei, electrons, protons, neutrons, quarks... Harnessing this complexity in a new model does not mean that we always use the most sophisticated theories. Sometimes, it is simply overkill and unnecessary. The same way relativistic effects are not considered during the preparation of a cheesecake, quantum effects can be ignored in some types of studies. Other times, they must be considered, though.

The complexity of the underlying theory of a model usually correlates with the mathematical principles behind, so the more complex a model becomes, the harder it is to apply it. Even if the model itself it is not mathematically complex, the accumulated steps to obtain a satisfactory answer can make it costly. Fortunately, the uprising of computation in last decades has greatly eased the resolution of the equations proposed by the advances in theoretical chemistry. In fact, the marriage of modeling and computation is so widespread that when one says *modeling*, it is commonly understood as *computational modeling*.

When I first started this Ph.D., I did not know the grounds I was standing on. I was very interested in com-

puters and technology, but I had reduced experience with programming. The fascinating field of molecular modeling —and in particular its structural aspects— gave me something visual to work with and taught me how little changes in an algorithm can have radical effects on the modeled structure. The following chapters will tell the story of how I found passion in research.





# 1

## Introduction

**S**INCE their inception, computers have been intimately connected to all areas of engineering and science. In fields like molecular biology and chemistry, they can assist in problems such as optimization of chemical reactions, drug discovery, material design, or structural characterization. The term *molecular modeling*, defined as the use of computational methods to describe the behavior of matter at the atomistic or molecular level,<sup>2</sup> encompasses all these applications as part of a vast family of techniques and tools.

The increasing presence in molecular sciences (see appendix A) must be also attributed to the efforts that the modeling community have been putting on two main fronts: user-friendliness and multiscale applicability. The former is devoted to creating intuitive interfaces with smooth learning curves, so that users can easily set up and analyze their calculations. This allows modelers to be more efficient and brings non-computational scientists closer to the field. The latter is concerned with joining distinct methods in a single coherent protocol to reach higher accuracy across different molecular scales. With this kind of protocols, phenomena ranging from recognition processes to catalysis can be accurately simulated. The success of these approaches finally crystalized in the 2013 Nobel Prize, granted to Warshel, Levitt and Karplus.<sup>3</sup>

Building software that satisfies both requirements —user-friendliness and multiscale applicability—, is not easy and requires an architecture that guarantees long-term development. A popular strategy consists of writing a robust core platform with a programmatic interface that supports the rest of features in the form of plugins or extensions. The core logic is usually written in a compiled language like C++, which provides high performance but slower development times. Once the heavy lifting is off-loaded to the core, the extensions can be written in more agile languages, like Python, Tcl or Ruby.

Among all the possible choices, Python has been the most successful over the last years, growing faster than

any other language.<sup>4</sup> It has been chosen as one of the main development language in very popular companies and software products,<sup>5-8</sup> and, for the interest of this project, molecular sciences. Specifically, it can be found as part of PyMol,<sup>9</sup> UCSF Chimera,<sup>10</sup> OpenMM,<sup>11</sup> Amber,<sup>12</sup> or Vina,<sup>13</sup> to name a few.

Choosing Python is not a temporary trend, but a fully educated decision. It allows to abstract away technical details like memory management, so the developer can focus on the features of the project. Its high-level description and object-oriented architecture provides an intuitive mindset to create packages with a strong modular component. This makes Python the perfect companion for developing new molecular modeling strategies. For example, each method can be abstracted in a separate module, allowing to build multiscale protocols by simply chaining the interfaced functions.

The premises of this Ph.D. stand precisely on these aspects, mainly focusing on: (1) developing and applying a novel multiscale platform based on Python flexibility, and (2) generating a new paradigm in 3D sketching for complex molecular design.

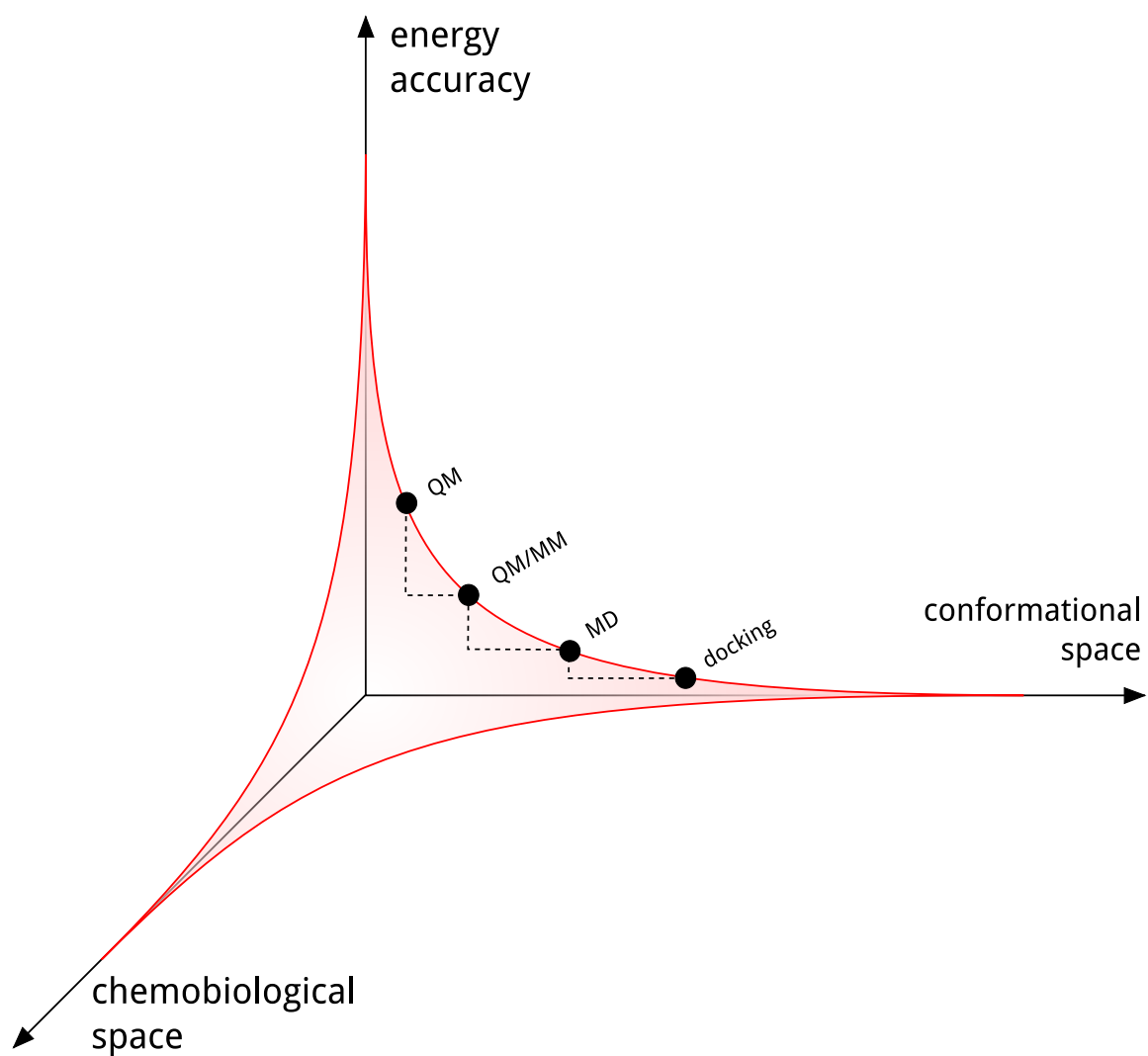
This introductory chapter will provide a brief overview on: (1) the major software categories in molecular modeling, which will be further detailed in chapter 2; (2) how they can be used together in integrative approaches; and (3) the difficulties, caveats and pending challenges present in these approaches.

## 1.1 MOLECULAR MODELING: ACCURACY VS SAMPLING

To understand the benefits of multiscale modeling, and why we need it, we must first review the available molecular modeling techniques: how they are used separately, and how we can join them together in a single protocol.

The most popular task performed in molecular modeling consists of describing the energetics of a system, which can then be used as a proxy for structure optimization, reaction path guessing or studies on dynamic behavior, among others. Depending on the method employed to obtain those energies, the results will resemble the experimental observations with higher or lower accuracy. In general, the higher the accuracy, the narrower the accessible search space (be it conformational or chemobiological, see fig. 1.1).

Atop the accuracy curve we can find Quantum Mechanics (QM) methods, which are based on quantum chemistry and the equations proposed by Schrödinger in 1925. This family of methods consider electrons explicitly, which allows them to study chemical reactivity with the highest precision. However, given the complexity of the calculations involved, even modern hardware cannot deal with more than a few hundred



**Figure 1.1:** When considering a multiscale protocol, one must face the compromise between the phenomena to observe and the reported accuracy on the results. Due to the limited availability of computational resources, one must face the compromise between large sampling capacities and accurate energies. Here depicted, a multiscale protocol on the conformational space.

atoms, resulting in limited sampling capacity.

The next major family, Molecular Mechanics (MM), discards explicit electronics for the sake of speed and scale. Instead of applying Schrödinger's theory, it considers molecules as a set of spheres connected by carefully calibrated springs, whose energy is given by Newton's laws of motion. This results in much simpler calculations which can deal with thousands of atoms almost instantaneously. In fact, their most popular usage is its time-dependent implementation, Molecular Dynamics (MD), which analyzes the evolution of a system over millions of timesteps to obtain an accurate representation on the molecular behavior along a few nanoseconds.

QM and MM can be used simultaneously in the same system using an approach called QM/MM. This hybrid method allows to consider larger structures by splitting the calculations in two regions modeled differently. QM is applied to a reduced part of the system that requires an accurate electronic representation, while the rest of the structure is processed with the simpler MM techniques. Both calculations are then integrated by using hybrid schemes such as IMOMM,<sup>14</sup> IMOMO<sup>15</sup> or, most popularly, ONIOM.<sup>16</sup>

Even though it can be argued that energy calculations are behind every molecular modeling study, not all methods focus on obtaining an accurate value. Sometimes, a distantly close one is enough. For example, protein-ligand docking studies are designed to obtain probable orientations in which a small molecule (ligand) can bind to a bigger one (protein). For this assessment, obtaining an accurate binding energy is not as important as navigating the search space reliably: it is preferred to focus on the fast generation of reasonable candidate poses instead of a locating a true global minimum. To do that in an affordable time, accurate energies are normally replaced by scoring functions that return pseudo-energies or even unitless scalars. These functions can be based on simplified molecular mechanics, knowledge-based potentials, shape complementarity or even simple geometric measurements.

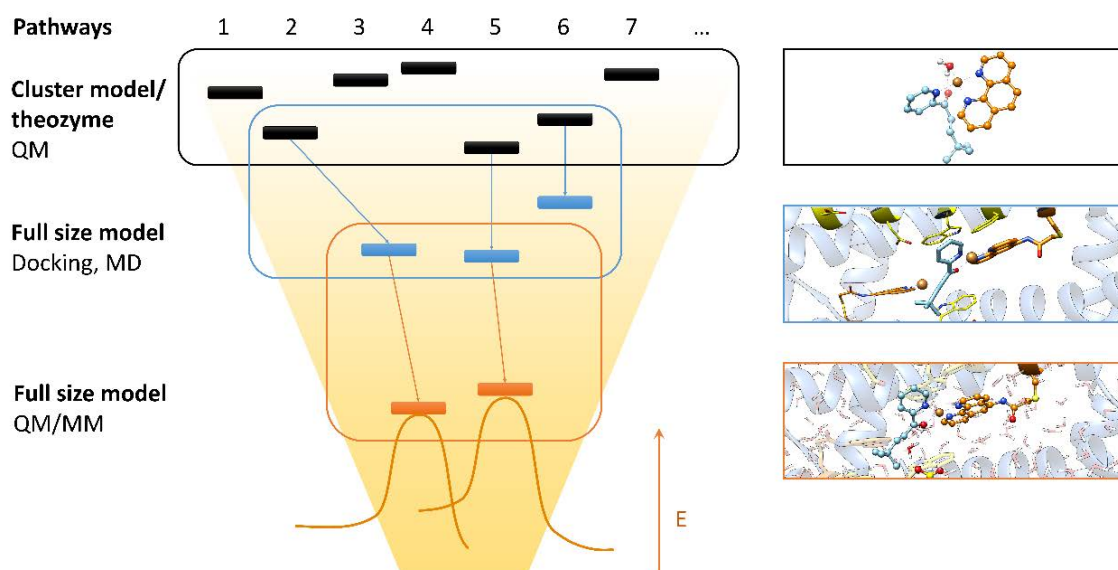
Drastic simplifications of energy are not uncommon in molecular modeling, especially if large search spaces must be analyzed efficiently. Normal Mode Analysis (NMA) apply an even simpler ball-and-springs model to obtain the principal vibration frequencies of a molecular structure. Under this approach, collective motions can be assessed in minutes without resorting to long molecular dynamics, which can take days or weeks to finish.

## 1.2 MULTISCALE AND INTEGRATIVE MODELING

Few computational studies are composed of only one step that relies on a single program to obtain a one-shot result after the first calculation. They usually comprise multiple stages that combine several theory levels and software packages to achieve the intended results. This is especially true in multiscale approaches, as pointed out by Grimme and Warshel (see Appendix A).

Multiscale modeling is often necessary because many molecular phenomena comprise a wide range of magnitudes at different scales. For example, studying the entire mechanism of an enzymatic reaction would require the description of binding processes and catalytic mechanisms, leading to the need of both a wide conformational sampling (docking, large scale or steered MD) and fine electronic treatment (QM or QM/MM), respectively.

The consequences of having different methods for varying sizes and timescales is that those methods must be combined in well-designed hybrid protocols. There is no clear strategy that dictates the proper sequence of methods and levels of theory. Most common strategies start with less accurate methods like docking, select some poses for further assessment using MD simulations, and pick some representative states to be optimized in QM or QM/MM schemes (see fig. 1.2).<sup>17</sup> However, depending on the information available, a study can begin with a DFT optimization of a reduced model and then progressively consider more atoms by decreasing the method accuracy: freezing some atoms in a cluster model, using semi-empirical approaches or hybrid methods and finally checking stability with a MD simulation.<sup>18</sup>



**Figure 1.2:** Example of a multiscale protocol applied for the identification of the catalytic mechanism of a metallic cofactor inside a metalloenzyme. (Reproduced from *Computational Methods in Enzyme Catalysis*).<sup>19</sup>

Every study is different and presents unique scientific and technical challenges, which are almost always overcome on an individual basis. Workarounds, patches and scripts are so commonly needed that it becomes an art on its own. In the future, this will be less of an issue as hardware gets faster and software smarter, but in the meantime standardizing some common workarounds can produce successful results. For this to happen, one must first understand those challenges.

### 1.2.1 DEALING WITH SOFTWARE FRAGMENTATION

The vast landscape of molecular modeling comprises hundreds of programs that have been developed to address different problems and, most of the time, with only that problem in mind. They were not designed to be part of a bigger, integrated workflow. Subsequently, good integration is needed between the involved software, which, unfortunately, it is not always the case.

Each of the steps will require different information depending on the supporting theory (in addition to atoms and coordinates, some will need connectivity, residue grouping, charges, parameters) and each of the involved programs might exhibit slight differences in how the exchanged files are parsed or exported. For example, one particular tool handles element names as case-independent, while the next one would only accept uppercase names. One tool might use different unit systems and a conversion is required (nanometers and angstroms is a common one). As a result, even after managing to correctly thread the needed file

formats as required, a robust behavior of the workflow is not guaranteed and ends up in fragile, unexpected performance.

This is both cause and consequence to the absence of a standard file format to deal with biomolecules and chemical compounds. On the contrary, first projects created custom file formats to deal with their own necessities, which has led to several file formats coexisting with an overlapping feature set. XYZ is the simplest, with only providing a list of elements and their coordinates, line by line; crystallographers use PDB files to handle big macromolecular structures such as proteins; MDL (now part of Dassault Systems) created MOL and SDF files to deal with small compounds; Tripos' Sybyl (now part of Certara) introduced MOL2 for their docking studies... only to name a few.

After many years of battling the file format war, a true standard is yet to be defined. While there are ongoing efforts trying to solve part of the issue (the crystallographic community is slowly replacing the troublesome PDB file format with mmCIF<sup>20,21</sup>), new developments still have to deal with multiple input and output files to be competitive (see fig. 1.3). Some projects exist to cover the compatibility issues, such as the popular OpenBabel suite,<sup>22</sup> but that constitutes only a band-aid and not a true solution.

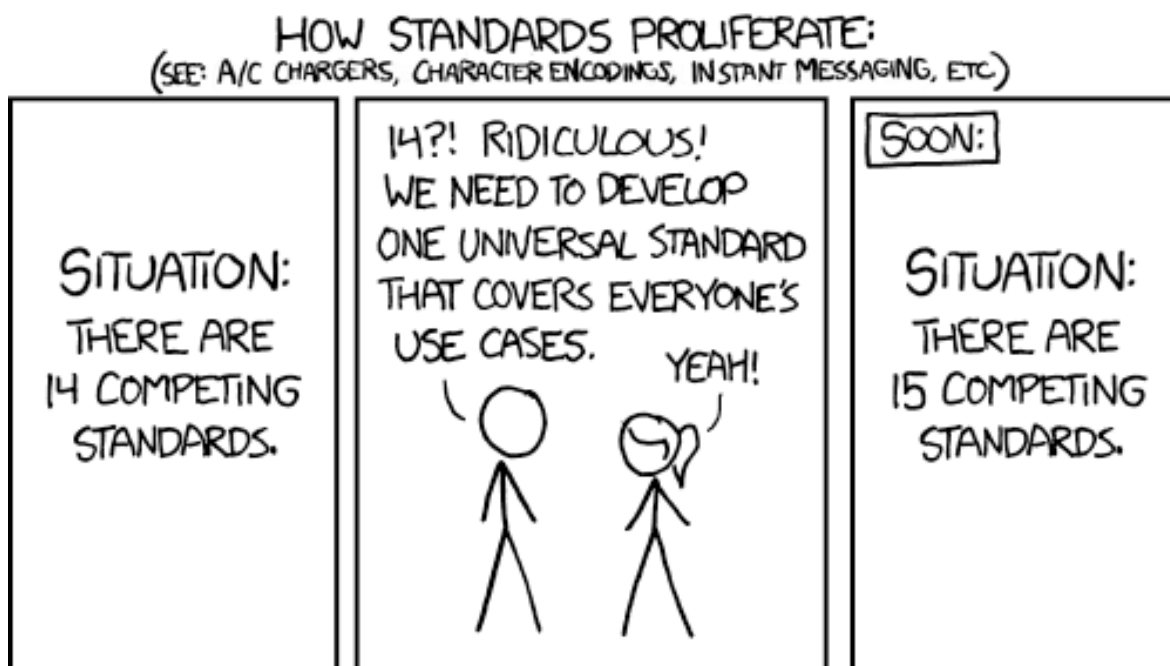


Figure 1.3: How standards proliferate. Reproduced from XKCD 927 (<http://www.xkcd.com>).

The solution to this fragmentation is not an easy feat, and several strategies could be implemented to alleviate the issue. The first milestone is, simply put, write good documentation: time-tested protocols, which detail the software and versions used in each step are of utmost importance. If files need to be converted and/or



edited along the process, these modifications must be noted too. Of course, this is only a patch and does not solve the background problem: format fragmentation. In this matter, several software-only attempts have been made and will be further detailed in the next section.

### 1.2.2 THE ECOSYSTEM OF INTEGRATIVE PLATFORMS

Graphical user interfaces (GUIs) are designed to provide a common workspace in which all operations can be carried out in a cohesive user experience. This avoids having to change contexts and learning new gestures for different tasks.

For molecular modeling, the perfect GUI would consist of a robust software platform that could act as the central hub for all molecular modeling programs, interfacing all of them seamlessly. In this sense, commercial graphical suites are likely the best option, since they have the means to build optimized interfaces for a broad range of computational workflows. Their main problem is, obviously, the licensing costs. Fortunately, alternatives exist for the academic users, be it special discounts for universities or free, open-source software developed by computational research groups. After all, a big part of the features present in commercial suites comes from academic developments (see *charm* vs *CHARMM* vs *CHARMm* (Biovia), or *AmberTools* vs *Amber*).

#### 1.2.2.1 COMMERCIAL SUITES

Molecular modeling companies build products to appeal all audiences, from novices to experts. While advanced and expert users have no problems in dealing with command line interfaces and text edition to manage programs, beginners usually tend to favor graphical interfaces. Dialogues, buttons and dropdown menus are way easier to handle than input files with arbitrary syntax. As a result, most commercial suites are graphical ones, creating the *molecular design software* concept: a graphical interface that provides a real-time visualization of 3D structures with interactive building and conformational modification. On top of this interface several functionalities can be added, such as assembling polymeric, periodic or solvated systems, guessing partial charges, geometry optimization, force field parametrization and much more. Depending on the tasks the user is going to perform more often, some differences can be drawn, but overall the most popular suites will offer the same feature set.

As of 2018, a handful of suites can fulfil the aforementioned requirements: Schrodinger's Maestro,<sup>23</sup> Dassault Systems' Discovery Studio (formerly from Accelrys, now part of Dassault as Biovia),<sup>24</sup> Chemical Com-

puting Group's Molecular Operating Environment (MOE)<sup>25</sup> or OpenEye Lead suite.<sup>26</sup> Obtaining access to these suites is subject to a one-time or periodic license, which even with academic discounts, can sit up well in the thousands. Even with those prices, it seems to be worthy: the sector is growing year after year and the forecast for the next decade are very optimistic.\*

Privative but free, SAMSON Connect (for Software for Adaptive modeling and Simulation Of Nanosystems) seems like a modern alternative backed by Inria. It requires an account and accepting their terms of use just to download the software, which includes a clause stating *You must be at least 18 years old to use the Service*, restricting their use in the school. That said, it offers a good software development kit (SDK), backed by good documentation on how to write custom *Elements* or extensions which are distributed through their online community. However, after further inspection, one quickly realizes that they are primarily focusing on materials design and nanosystems, not macromolecules and small compounds.

*All that glisters is not gold*, said Shakespeare. All-in-one solutions, as provided by commercial suites, are very appealing to novice users, but when the intended purposes of the tool must be pushed, the commodities of the graphical become an impediment instead. This is crucial when modeling new areas of structural biology or organometallics, such as artificial metalloenzymes or metal-organic frameworks. In these frontier-fields, researchers cannot simply rely on existing protocols and tools, they create their own by trial-and-error or even code new algorithms to overcome the difficulties. In other words, the state of the art is not always available within the licensed suite, whose update might come a year too late.

It is true that through extended configuration files, some hardcoded parameters can be modified and, if the software features a decent Application Programming Interface (API), scripting languages can be applied to implement new algorithms and techniques. Modifying the source code is normally not possible because, being commercial, the source is not disclosed. If they do release it as open source, it is for an old version that, while helpful, it is not ideal.<sup>9</sup> In that matter, the academic and/or open source software have a clear advantage.

---

\*A study by Industry Arc Research projected that the Computational Medicine and Drug Discovery Software would reach 6.78 billion USD by 2020,<sup>27</sup> which agrees with GrandView Research studies: the structural biology and molecular modeling market will be worth 13.1 billion by 2025.<sup>28</sup> According to a study published by Accuray Research in May, 2017,<sup>29</sup> which cites companies such as Accelrys, Certara, CCG or Schrodinger, the global computational biology market will reach a value of 11.25 billion USD by 2025. More studies on biosimulation, offer markets worth up to 2.88 billion by 2022.<sup>30</sup> This has been known for investors, of course, as evidenced by the 10 million USD round received by Schrodinger LLC from Bill Gates in 2010.

### 1.2.2.2 ACADEMIC SOFTWARE

While there are software companies that do some research and develop new methods, only a handful publish their results, so it is fair to assume that most of the public knowledge comes from publications submitted by academic research groups. After all, most of the commercial scientific software was, at some point, of academic nature.<sup>31,32</sup>

The academic landscape is not only broad, but also disperse. Lots of small projects are released weekly and it is very difficult to keep track of them all. A couple of web directories have emerged recently,<sup>33,34</sup> giving a small insight into the field. In OMICTools, only the proteomics category displays almost 9000 entries. GitHub,<sup>35</sup> the de-facto online repository for open-source software, shows more than 2000 repositories for *chemistry* searches.

When it comes to integrative suites, the analysis is much simpler. Few groups can dedicate all their efforts to building a wide-spectrum tool, especially when the commercial suites are well-established. If any, the one weakness that can be easily exploited is price: releasing an open-source tool with a comparable feature set would be very competitive.

One the best attempts to fill the gap is the UCSF Chimera project. First released in 2000<sup>36</sup> and published in 2004,<sup>10</sup> after 18 years of development UCSF Chimera shows its age: the graphical interface looks dated and, with today's standards, clunky. But that age is also a blessing: the software is stable, robust and mature, and accumulates a lot of modules to perform all kinds of analysis: clashes detection, H-bond depiction, density map fitting, peptide building... It comprises a huge number of small tools that, together, make for a good modeling suite. However, the diverse origin of the tools (some are built by the Chimera developers, but a good part comes from third-party collaborators), end up creating a feeling of unstructured workflow. It also lacks key elements like Quantum Mechanics integration or a modern Molecular Dynamics program (it does include MMTK, an abandoned project that cannot provide the performance expected with modern architectures).

This is caused by three main issues: (1) There is no developer documentation. The few resources are scattered between the mailing list and the Python code itself. (2) 15 years of back-compatibility surely comes with a price, which means shipping old projects with deprecated dependencies. (3) A deliberate isolation of the platform to ensure consistent behavior in all platforms prevents the developer from writing software with modern tools and libraries. Solving point (1) would be a huge effort that only the developers could satisfy adequately, but points (2) and (3) can be addressed with patches and clever workarounds, which are the

reasoning behind one of the developments presented in this thesis (see chapter 5). Fortunately, the same team behind UCSF Chimera is now working on ChimeraX, focused on the migration to modern standards and providing a central repository for 3<sup>rd</sup> party extensions (the *Toolsbed*). While the core code is now available (and with proper documentation), the extra modules would take more time. This means that the feature set is yet to be comparable.

Classic visualizing software like VMD or PyMol could also fill the gap: they have been developed for years and now accumulate a good number of extra features thanks to the contributed extensions. The problem is that they lack an attractive, intuitive interface to begin with and both feel like a modest 3D viewer with extra modules bolted on: functional, but not ideal. The open-source project Avogadro does offer a tighter interface, good documentation and interfaces to most QM and MM software, but its focus seems more centered on small compounds rather than macromolecules. For example, by default it does not depict the secondary structure of proteins like ribbons, and when selected the result is not as aesthetically pleasant as Chimera, VMD or PyMol.

### 1.2.3 THE ROLE OF SCRIPTING IN THE INTEGRATION OF SOFTWARE PROJECTS

Putting different tools to work together, even when they were not designed with that purpose in mind, is one of the key skills that an advanced molecular modeler must master. Without programming knowledge, the task becomes almost impossible: copy-pasting parts of a file only gets you so far and quickly become tedious.

Writing little *glue* scripts to adapt the output and input files of several programs is relatively easy and only involves knowledge in text manipulation. For this task, several languages are adequate, such as Bash, Tcl, Lua, Perl or Python. Each has enjoyed a period of popularity, but nowadays Python is king both in scripting and more advanced tasks.<sup>37</sup> On top of being free, this is attributed to its easy-to-learn syntax, high readability, dynamic typing, and its general-purpose, rich library of built-in packages (the *batteries included* motto) which has allowed the development of a huge ecosystem of high-quality scientific packages (NumPy, SciPy, Scikit, Pandas, SimPy, Matplotlib, Jupyter...).

Being interpreted, Python is not a particularly fast programming language and can fall behind the performance of compiled languages (C, C++, Go, Rust) or even Java. However, putting different programs or libraries to work together is not very computationally demanding and the easy syntax really pays off in developing times. Even if performance is an issue, it is often smarter to accelerate the critical parts (with NumPy,<sup>38</sup> Numba,<sup>39</sup> Cython,<sup>40</sup> or C/Fortran extensions) and code the rest of the program logic in pure Python.

The trend is obvious and most of the new advances in scientific programming are either built with Python or provide a Python layer around the compiled core, as evidenced in all the machine learning/deep learning/neural networks/blockchain projects recently launched (i.e. TensorFlow,<sup>41</sup> Theano<sup>42</sup> or PyTorch<sup>43</sup>).

Instead of programming, the researcher can also devote to using a single modeling suite. The previous sections have tried to shed light on all the graphical suites, both with commercial and non-commercial products. Here, the graphical canvas (or, more precisely, the programmatic objects thereby represented) acts like the communicating thread across the involved steps. For example, the user builds a molecule in the 3D viewer, and a plugin writes the input file to an external program to do additional operations. The results are then imported back into the canvas and update the needed fields. Originally, the canvas and the external program did not understand each other, but with a specifically crafted intermediate module, they can. It is up to each of the extra modules to act as interpreters between the core platform and the external software.

To support the development of additional plugins, almost all modeling suites feature some kind of programmatic interface (API) to extend their core features. That API exposes the functionality of the platform to other developers, usually with a scripting language. In all the suites mentioned, Python is consistently chosen as that language. One way or another, learning even some programming skills is highly beneficial to any molecular modeler.

### 1.3 MODELING WITH SCARCE DATA: ABUSING MODULAR APPROACHES

All modeling approaches seem to follow a notion: accurate structural details are subordinated to fine energy descriptions. While true, this mindset can be limiting when it comes to modeling systems with scarce experimental information available. This is important because, regardless what methods are finally applied in a multiscale protocol, any software will always need input information to work with. Depending on the tasks to perform, the required data can range from simple geometry specifications, to connectivity, atom types, charges, spin state, temperature, optimization steps or algorithmic treatment (see section 2.5).

For some modeling tasks, the hardest part is simply telling whether that exercise is feasible or not. In those cases, an accurate energy description is not always initially needed, and the input requirements can be relaxed. In those cases, one can focus on simply obtaining a good enough starting structure. Instead of committing to strict parameterization, sometimes a reduced number of descriptors are enough for dealing with hypotheses-driven questions. If all that matters is obtaining an answer, those descriptors can be supported by any existing technique: simple geometric measurements, knowledge-based scoring functions, molecular mechanics force

fields, or, if necessary, even energies provided by quantum mechanics methods.

If this type of exercise is performed regularly, molecular modelers would start accumulating experience and recognizing patterns in the protocols applied. Ultimately, they would devise a platform where each descriptor is encapsulated in a separate module and can be recruited on demand to compose solutions for arbitrary molecular modeling tasks. If necessary, separate modules can deal with specific areas of the problem to bypass any potentially unmet requirements in other regions.

With a scripting language like Python, building such a platform is possible. Additionally, thanks to the great number of existing libraries for scientific computing, the development efforts are greatly simplified and can fit within the typical timescales of a Ph.D. scholarship.



# 2

## Materials & Methods

COMPUTATIONAL chemistry was first coined by Frank Westheimer at a conference in 1966 to refer to Allinger's work on molecular mechanics,<sup>44</sup> separating it from other studies that would fall under the *theoretical chemistry* category at that time. Nowadays, the term is broader and considers far more techniques. In fact, under the modern definition earlier works would be considered *computational chemistry*; albeit analog ones. In 1930, Kettering and more researchers in General Motors, built ball-and-spring models for several molecules and correlated their vibration modes with their Raman spectra. This work, titled *A representation of the dynamic properties of molecules by mechanical models*,<sup>45</sup> could be considered one of the first molecular modeling studies.

### 2.1 ORIGINS OF MOLECULAR MODELING

The birth of theoretical chemistry, from a quantum chemistry perspective, can be pinpointed with the description of the Schrödinger equations in 1925–1926.<sup>46</sup> The first application of quantum mechanics came a year later, in 1927, with the publication of Burrau's studies<sup>47</sup> on  $H_2^+$  and Heitler and London calculations<sup>48</sup> on  $H_2$ . The field began to grow rapidly (Teller,<sup>49</sup> Mulliken,<sup>50</sup> Born,<sup>51</sup> Oppenheimer,<sup>52</sup> Pauling,<sup>53</sup> Hückel,<sup>54</sup> Hartree,<sup>55</sup> Fock<sup>56</sup>...) and computational implementations of the new theoretical framework started to be feasible after the advances in computer technology during the late 40s.<sup>57</sup> In the 50s and 60s, several milestone papers were published, making for the first documented computational chemistry calculations.<sup>58,59</sup> Also, non-quantum, classical approaches stemming from theoretical physics started to emerge, although not strictly dealing with chemistry problems.<sup>60–62</sup>

By the 70s, several journals had appeared to target computational chemistry and the first quantum chem-



istry packages began to be distributed (including the first version of the now ubiquitous Gaussian<sup>31</sup>). The available hardware back then only allowed for *ab initio*\* calculations of molecules as big as naphthalene and azulene (18 atoms). In those same years, molecular mechanics methods became more popular, especially with the contributions by Lifson's CFF,<sup>63–65</sup> Allinger's MM series,<sup>66,67</sup> Scheraga's ECEPP potentials,<sup>68,69</sup> Karplus' CHARMM,<sup>70</sup> van Gunsteren's GROMOS<sup>71</sup> and others, proving the power of empirical parameterization. By the 80s, Computer Assisted Molecular Design (CAMD) was the new hype that would revolutionize the pharmaceutical industry,<sup>72</sup> and by the 90s it was clear that computational chemistry was broader than quantum chemistry. In the preface of *Reviews in Computational Chemistry Volume I*,<sup>73</sup> editors acknowledge that:

"[...] we do not view the terms theoretical chemistry and computational chemistry as synonymous. Computational chemistry sometimes involves application of computerized algorithms from quantum theory, but computational chemistry is certainly more than quantum chemistry [...]. Molecular mechanics, molecular dynamics, computer graphics, molecular modeling, and computer-assisted molecular design are other important aspects of computational chemistry [...]"

Nowadays, molecular modeling encompasses techniques and strategies beyond what is traditionally considered computational chemistry (this is, quantum and molecular mechanics, mainly). With the current popularity of more expressive languages, it will be possible to devise new algorithms and protocols in less time. Additionally, it can be argued that recent advances do not only come from developing new methods, but also from reapplying the existing ones under new computational architectures, both in terms of hardware and software. For example, parallelizable code and, in particular, GPU-acceleration, have taken the performance of Molecular Mechanics methods to the next level.<sup>†</sup>

The following pages will introduce the major families of software approaches in computational chemistry and molecular modeling. They will be listed in two major groups depending on the focus (energy description or conformational sampling), and in descending order of level of theory,<sup>‡</sup> which, in practice, means going from high-accuracy to lower-accuracy methods. Finally, an overview on optimization methods will depict the relationships between energy description and space sampling in the context of molecular modeling.

---

\* A model is said to be *ab initio* when it only considers the resolution of the first principle equations (Schrödinger's), without support from experimental observations. Empirically derived models do take these observations into account, very often as a workaround to avoid solving the full equation system. When the two approaches are combined, those are semi-empirical methods.

<sup>†</sup>Quantum Mechanics software is also starting to use GPU-accelerated code, but the support is still limited (see appendix A).

<sup>‡</sup>An estimation of the complexity of the theory supporting the method, that while arbitrary, gives a quick understanding of the intricacies involved. Higher theory levels usually refer to highly accurate, computationally demanding methods that rely on complex mathematical models. Lower theory levels, on the contrary, refer to less accurate and computationally cheap methods relying on simpler models.

## 2.2 ENERGY DESCRIPTION

### 2.2.1 QUANTUM MECHANICS (QM)

The main idea behind quantum chemistry is that molecules and, by extension, all ordinary matter, can be viewed as composed only of positively charged nuclei and negatively charged electrons. Mathematically, this can be expressed with the time-independent Schrödinger equation:

$$H\Psi = E\Psi = T_n + T_e + V_{ne} + V_{ee} + V_{nn} \quad (2.1)$$

(Time-independent Schrödinger's equation)

, where  $H$  is the Hamiltonian operator,  $E$  is the total energy of the system,  $T_n$  and  $T_e$  are the kinetic energies of nuclei and electrons, respectively, and  $V_{ne}$ ,  $V_{ee}$  and  $V_{nn}$  are the potential energy between nuclei and electrons, electrons against each other, and nuclei against each other, respectively. Solving this equation for any system would mean seeking the eigenfunctions and eigenvalues of that Hamiltonian.

The details of such resolution are out of the scope of this thesis, but some comments can be made about its practical effects. Since only one-particle and two-particle systems can be solved analytically, numerical methods are employed to approximate the solution for systems of 3 or more particles: the many-body problem. While not analytical, the same methods can be applied iteratively to any given precision; the only restraints are computational and time resources. As a result, some approximations have been developed over the years to simplify the equation solving process without much accuracy loss.

The first approximation to appear was the Born-Oppenheimer approximation. It relies on the big mass difference between nuclei and electrons. In hydrogen, the monoprotonic nucleus is already 1800 times heavier than the electron; for uranium, the nucleus/electron mass ratio goes up to 430,000. This leads to consider that, given the enormous mass difference, electrons and nuclei move in different time-scales: if the nucleus moves, the electrons would follow *instantaneously*. This means that the nucleus can be considered stationary for electronic timescales, and appear as parameters in the equation, greatly simplifying its solution.

Even with uncoupled motions, the dynamics of electrons are complex and require advanced computational methods. A significant simplification would be to treat electrons as independent from each other by introducing an *independent-particle* model, either by neglecting all interactions altogether, or, even better, by

introducing an average interaction factor. These approximations are collected under the Hartree-Fock (HF) theory. In HF methods, electronic interactions are not explicitly described, but with a large basis set 99% of the energy can be described by the HF wave function. The difference between the energy predicted by (Restricted) HF calculations and the real energy is called *electronic correlation*, which in certain chemical phenomena is key to obtaining accurate predictions. As a result, three main strategies have been developed to calculate it explicitly: Configuration Interaction (CI), Many-Body Perturbation Theory (MBPT) and Coupled Cluster (CC).

Evidently, these methods involve extra computational complexity, so in some cases, more aggressive approximations have been applied. This is the case of semi-empirical methods, which instead of trying to resolve some of the most complex integrals, resort to experimental parametrization of the results. While it is true that this leads to less accurate results, they are way faster and, with sensible parameters, the difference can be neglected depending on the study at hand.

HF theory is not the only applicable approximation to simplify the many-electron problem. In a way, Density Functional Theory (DFT) can be seen as a more efficient strategy to tackle the challenge. DFT is based on the Hohenberg and Kohn proof, which suggests that the ground state electronic energy can be determined completely by the electron density. In practice, it offers a computational cost similar to HF theory, but with the possibility of providing more accurate results<sup>§</sup> and is widely used nowadays.

All these methods are applied to solve the time-independent Schrödinger equation; this is, to obtain the energy of a system with given coordinates. If the dynamics of the system (i.e. evolution along time) must be studied, the time-dependent equation must be solved, which involves highly complex calculations to obtain reasonable accuracy. With current resources, only di- and triatomic species can be simulated using this approach. Additional atoms would need to be frozen or treated classically.

A workaround would involve a semi-classical approach. In Ab Initio Molecular Dynamics (AIMD), electrons are treated quantum-mechanically, and nuclei, classically (Born-Oppenheimer Molecular Dynamics, BOMD). At each time step, a converged wave function is obtained and the corresponding nuclear gradients are used to propagate the time-evolution. However, for accurate results, one must resort to tightly converged wave functions (in BOMD) or very small timesteps (in Car-Parrinello Molecular Dynamics, CPMD).

---

<sup>§</sup>They need a good exchange-correlation functional, though, which can only be obtained exactly for the free electron gas; for other cases, approximations must be employed, like local-density (LDA), local spin-density (LSDA) or generalized gradient (GGA, meta-GGA) approaches.

## 2.2.2 FORCE FIELDS: MOLECULAR MECHANICS, MOLECULAR DYNAMICS AND METADYNAMICS

Depending on the phenomena under study, explicit consideration of electrons is not always necessary. If that's the case, the Schrödinger equation can be bypassed and the electronic energy can be written as a parametric function of the nuclear coordinates. Those parameters are fitted to reproduce either experimental measurements or results obtained with higher levels of theory (i.e. QM). The set of parameters needed to write the set of equations is called *force field*, and the theory behind this strategy is called *Molecular Mechanics* (MM). Given the size of nuclei, these can be treated classically with Newton's second law with sufficient accuracy, resulting in equations much simpler than their QM counterpart, which allow faster computations and, as a result, dealing with larger systems (tens of thousands of atoms). Neglecting the existence of electrons has some consequences, though: bonding information is lost and must be provided explicitly, rather than being an inherent result of the equation.

In MM, molecules are described by a *ball and spring* model: atoms are abstracted as spheres of given radius and *softness*, and bonds have length and *stiffness*. In such an ensemble, the potential energy of the system can be described as the sum of several components: stretching, bending, torsions, non-bonding interactions (usually, Van der Waals and Coulomb), and a cross-term of the first three.

$$E_{FF} = E_{stretching} + E_{bending} + E_{torsional} + E_{non-bonding} + E_{cross} \quad (2.2)$$

(Force field energy)

If each of the terms can be expressed as a function of the coordinates for the involved atoms, the potential energy can be obtained, and geometries and relative energies can be calculated by optimization. Thus, all that remains is to obtain the parameters for each type of interaction involved in the system under study. Fortunately, most molecules can be described as a composition of a small set of functional subunits or groups, the properties of which rarely change. For example, all C=O bonds are around 1.22 Å long. Instead of having to parameterize each type of atom and bond for each type of molecule, these similarities allow to construct set of parameters with reasonable easiness, in principle.

As opposed to QM, these equations can be solved efficiently enough to consider time-dependent analysis. If one takes the force field equations and calculates the resulting forces and velocities, the position of the atoms can be figured out with high accuracy if the timestep is small enough (i.e. in the same order of magnitude as

the smallest perturbation studied: hydrogen vibration, in the order of femtoseconds). This strategy gives rise to a field called Molecular Dynamics, which allows to study the behavior of molecules along time. Modern computer architectures allow to resolve each timestep around  $10^8$  times daily! Processes involving seconds in real-life can be calculated within a month.

Gaining access to such magnitudes allows to calculate properties not available in other higher levels of theory: conformational changes, binding pathways, or thermodynamic magnitudes such as free energy. However, in those calculations the potential energy surface (PES) must be well-sampled. Several methods account for this issue, such as metadynamics (MTD), umbrella sampling or adaptively-biased MD. In the case of MTD, the system is assumed to be describable by a few collective variables or reaction coordinates, which are then explored in such a way that revisiting sampled states is discouraged.

### 2.2.3 QM/MM

Dealing with big systems (more than 300-500 atoms) with QM techniques is not feasible due to computational restraints. In some cases, though, only a subset of the system actually needs explicit consideration of the electrons. One strategy would consist of pruning the nonimportant parts, replacing them by smaller functional units which would keep the system's *chemical identity*.

In other studies, like enzyme reactivity, the nonreactive part of the enzyme is assumed to be important in keeping the active site conformation, and as such, cannot be pruned out. While some authors do model enzyme reactivity with a prune-and-freeze strategy (the *quantum chemical cluster*<sup>74</sup>), a large part of the research community prefers to consider the whole protein in the calculation. This can be approached with QM/MM methods, in which the reactive atoms and its immediate surroundings are studied with QM, while the rest of the system is treated classically. The energy of the system is then calculated as a sum of the QM energy, the MM energy and the interaction between both.

$$E_{total} = E_{QM} + E_{MM} + E_{QM/MM} \quad (2.3)$$

(QM/MM energy)

The independent QM and MM terms can be calculated straightforwardly as explained before, but deciding on how to calculate the interaction is not as intuitive. In general, three strategies can be applied: (1) mechanical embedding, which only considers bonded and steric effects; (2) electrostatic embedding, which

also considers the electric field of the MM part; and (3) polarizable embedding, which adds polarizabilities between the QM and MM parts. Even in the simplest case, mechanical embedding, technical difficulties may arise if a bond is cleaved by the QM/MM partitioning, which would require balancing the unpaired electrons in the QM part by adding a *link* atom invisible to the MM part. Similarly, the dangling bond in the MM part must be dealt somehow, normally adding the needed stretching, bending and torsional terms.

Mixing methods of different level theories for a single system is generalized in the ONIOM approach, which assumes additivity of the different *layers*. If two layers are considered, it classifies the system in *model* (the subset of the system which is dealt with a higher level of theory) and *real* (the full model, including the *model* subset). The *model layer* is calculated both with the higher level of theory (usually QM) and the lower one (usually a force field), while the real layer is only calculated with the lower level of theory. Then, the energy is given as:

$$E_{ONIOM} = E_{high}^{real} = E_{high}^{model} - E_{low}^{model} + E_{low}^{real} \quad (2.4)$$

(ONIOM energy)

Another intrinsic problem to QM/MM methods is the sampling. As the contribution of the MM part to the QM/MM energy is larger than the QM due to the number of atoms involved (around 100 in QM or 1,000-10,000 in MM), reported energy will be very sensitive to even small changes in the MM layer. As a result, when an energy profile must be calculated, the MM part is kept rigid during the whole process, which requires a carefully chosen initial structure to begin with. This usually involves a protocol where a long MD run is performed to obtain a representative snapshot of the simulation.

#### 2.2.4 COARSE-GRAINED MODELING

Coarse-grained models constitute an additional simplification level in the diverse representations of molecular models, including proteins,<sup>75,76</sup> nucleic acids,<sup>77,78</sup> or lipid membranes.<sup>79</sup> Instead of individually describing each atom in the system, they group related atoms together in a single entity. These groups, sometimes called *pseudoatoms*, can have several levels of granularity depending on the system scale. For example, simulating the dynamics of a viral capsid might need monomer-level granularity,<sup>80</sup> while trying to simulate a full cell might involve full organelles being represented individually.<sup>81</sup>

## 2.3 CONFORMATIONAL SAMPLING

Even with the latest advances in computer architecture and software improvements to make the most out of it, some potential energy surfaces are too vast and complex to be explored efficiently with molecular dynamics, let alone quantum dynamics. While this might not apply in the next decade, techniques that take educated shortcuts to traverse broad conformational spaces are still necessary. This is particularly true in large molecules such as proteins, where structural fluctuations occur at very different time scales and amplitudes: local motions take place are fast and short ( $10^{-15}$  to  $10^{-1}$ s, 0.01-5 Å), and collective motions are slow and long ( $10^{-9}$  to 1s, 1-10 Å).

### 2.3.1 NORMAL MODES ANALYSIS

In Normal Modes Analysis (NMA), slow, collective motions of a molecular structure can be approximated through the composition of its independent (normal) harmonic oscillations (modes). The NMA approach is parallel to MM in its theoretical support, meaning that bonds are replaced by *springs* or harmonic oscillators. As a result, the mathematical treatment consists of finding the eigenvalue of each coupled oscillator, which is simple enough to be calculated efficiently for thousands of atoms. In fact, there is no obligation to treat each atom and bond independently, as nearby atoms can be grouped together in a new body to reduce the needed number of oscillators (coarse-grained NMA). All these approximations allow dynamic insights on the movements of a macromolecule without the technical limitations of MD, which require long runs to observe slow motions. Since NMA does not consider time at all and only requires an initial structure on which to compute the normal modes, the vibrations obtained only apply to that particular conformation. In other words, only the local potential energy wells are explored, which can provide a short-sighted analysis if no other conformations are assessed.

### 2.3.2 RECOGNITION PROCESSES

Recognition processes involve large conformational changes and are key in research areas such as drug development or metabolic studies. Docking techniques were developed to describe feasible binding modes of small molecules (ligands) within a bigger macromolecule (the host, which is normally a protein). To do that, potential binding pockets in the host are explored explosively by placing the ligand in random orientations and positions (rigid body transformations), contemplating some internal flexibility if necessary (through torsions in the small molecule or rotamer exchange for the side chains in the protein), and finally assessing

their non-bonding interactions. Since the spatial landscape needs to be explored efficiently, with thousands of attempts before finding a good pose, the energy representation is often cruder than in higher levels of theory in honor of speed and efficiency: the scoring function. While in principle it could use full molecular dynamics simulations,<sup>82,83</sup> most popular approaches resort to simplifications, like knowledge-based parameters obtained out of structural databases<sup>84</sup> or empirical evidences.<sup>85-87</sup> Other approaches resort to shape complementarity between the Van der Waals surfaces of the involved molecules<sup>88-90</sup> or, more recently, to Artificial Intelligence (AI) deep learning techniques.<sup>91</sup>

Docking techniques can also involve other type of molecules, like protein-protein or protein-nucleic acid studies. In this variant, more approximations are needed due to the broader conformational space involved, like only considering rigid-body transformations.

## 2.4 BEYOND CARTESIAN COORDINATES: NAVIGATING THE CHEMICAL SPACE

Of all the techniques detailed until now, only QM studies allow topology changes like bond breaking or creation. This is, when dealing with MM, NMA or docking, most approaches will assume that the starting set of atoms and their connectivity will be the same during the whole simulation. As a result, to assess different variants of the same compound one must run the same protocol separately for each variant.

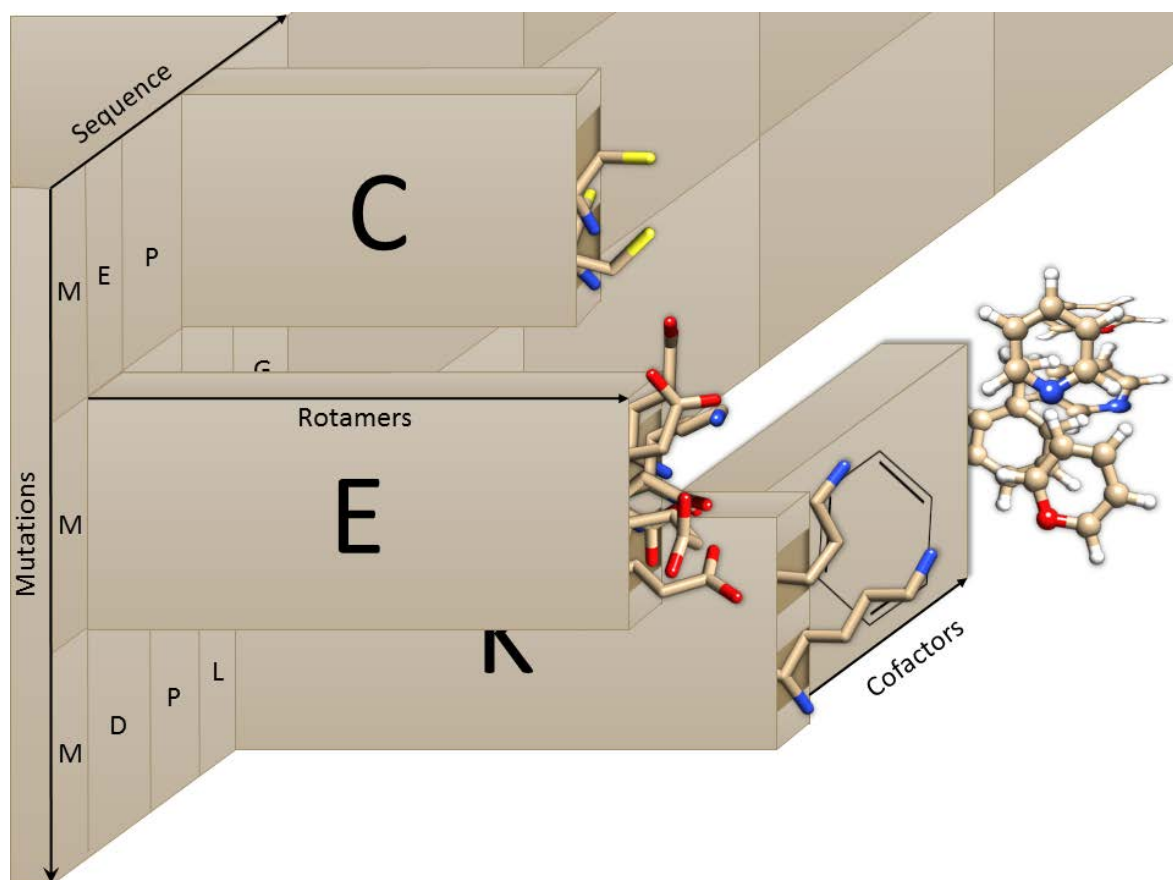
The most popular technique employing this strategy is virtual screening, which essentially performs docking calculations for massive datasets of drug-like compounds. In most implementations, the database must be built beforehand and the chemical space is only explored by trying different entries in the dataset. This is, no algorithmic modification of the structure is done during the simulation. With big enough datasets, sampling should not be a problem, but some studies point out that the chemical space explored this way is very limited.<sup>92</sup>

Some software projects do attempt to explore the chemical space algorithmically,<sup>93-100</sup> but they still struggle to find mainstream usage in the pharma industry. The implemented approaches often resort to chemical synthesis guidelines like CLICK chemistry<sup>101</sup> or graph abstractions.<sup>102</sup> This can be useful in docking calculations that wish to account for some chemical variability in the ligands assessed or rational design of potential inhibitors.

Since protein-ligand docking studies involve at least two molecules, it makes sense to investigate chemical variability in the host. While the theoretical chemical space available in proteins is huge due to the number of atoms alone, fortunately is more constrained: all proteins are chains of the same 20 residues or amino acids.



Hence, assessing possible variations *only* involve changing one particular residue for another of the remaining nineteen. It seems simple at first, but the possible variations explode exponentially with chain length, which means that even for small peptides of around 30 residues it results in an unfathomable number:  $30^{20}$  or around  $3.5 \cdot 10^{29}$  (see fig. 2.1). Subsequently, only a few positions are studied and the variations applied are somehow rational and studied beforehand. Some tools exist to evaluate the potential consequences of a mutation in a protein,<sup>103–106</sup> which can help discard destabilizing changes before doing more intense computations. After all, a single change in a key residue can alter the final structure dramatically.<sup>107</sup>



**Figure 2.1:** In any study on protein-ligand interactions, considering sequence mutation of protein scaffolds, each with its possible sidechain orientations, already produces a combinatorial explosion. Taking different possible ligands into account adds one more dimension to the search space. (Reproduced from *Artificial Metalloenzymes and MetalloDNAzymes in Catalysis*<sup>108</sup>).

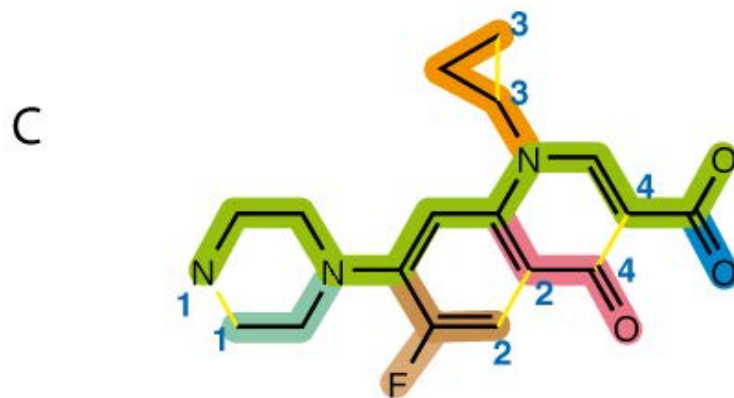
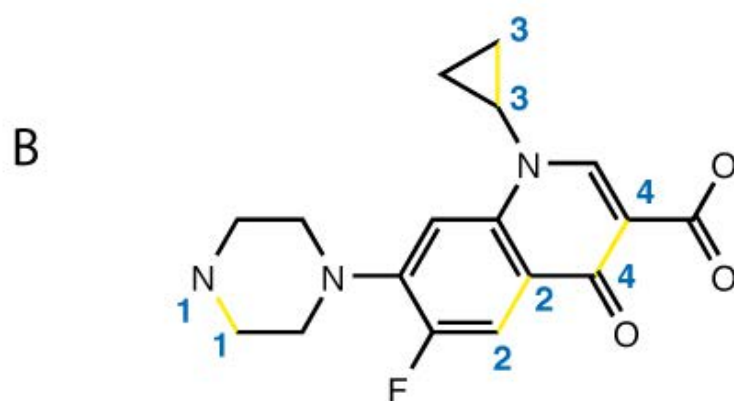
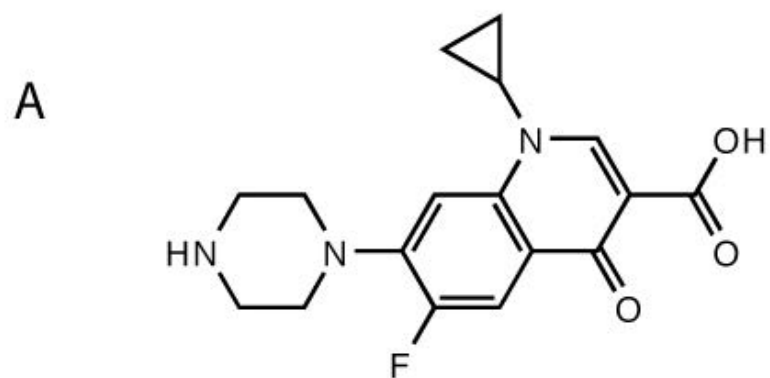
#### 2.4.0.1 CHEMINFORMATICS

There are molecular modeling techniques that do not need to rely on 3D coordinates that information to produce useful results. Cheminformatics are mainly concerned with the creation and maintenance of small compound databases with support for indexing and similarity searches. The main exponent of chemin-

formatics approaches is probably QSAR techniques.

Quantitative Structure-Activity Relationship (QSAR) studies apply classification or regression statistical techniques to predict experimental observables from basic molecular descriptors. The *activity* under study can comprise different variables: biological activity of a drug-like compound, boiling point, potential reactivity, toxicity... All QSAR methods are based on the structure-activity relationship (SAR) assumption: similar compounds will have similar activities. The main issue is how to measure that similarity: number of atoms, functional groups, connectivity... As in all statistics, large datasets are needed to obtain valid results. The first step in QSAR studies is to train the statistical model with a huge library of compounds. Once the model is trained, it can be used to predict the activity of a compound originally not present in the library.

QSAR input data do not provide 3D-structural data (with the exception of the 3D-QSAR variants), but 2D-topological information. These are normally supplied with special character strings called SMILES (Simplified Molecular Input Line Entry Specification),<sup>109</sup> which can describe compounds unambiguously without resorting to explicit coordinates specification. SMILES strings work by enumerating the atoms involved in a molecule with their element symbol, except hydrogen, which is usually implicitly considered. Simple bonds are assumed between linearly adjacent elements. If a ramification occurs, it must be specified with parenthesis. Numeric tags are used to signal the starting and ending atoms of cyclic substructures (see fig. 2.2). For example, butane would be represented as CCCC, while D-glucose would be C(C1C(C(C(C(O1)O)O)O)O)O.



D

N1CCN(CC1)C(C(F)=C2)=CC(=C2C4=O)N(C3CC3)C=C4C(=O)O

**Figure 2.2:** SMILES notation can depict a substituted aromatic ring as linear chains with branches. In this case, 3-cyanoanisole is being written as C0c(c1)cccc1C#N.

## 2.5 BUILDING MODELS FROM SCRATCH

Up to this point, it has been assumed that the molecular modeler had all the needed input data ready for analysis and simulation, but unfortunately that is not always the case.

In QM studies, when only small molecules are studied, manual building of the model can be contemplated as a possibility using a graphical interface,<sup>110,111</sup> but when the model grows, the possible orientation of rotatable bonds or the configuration of the subunits can be a daunting task to solve manually and even a stopping barrier in the research. Additionally, in some cases, the initial structure is only conceived as a 2D depiction with ChemDraw or similar software, which has to be transformed into a 3D model. This is a fairly common process, and as such most software suites include a 2D→3D converter. However, additional refinements are normally required afterwards. If the system features many degrees of freedom, this can be tedious. A common alternative consists of performing high-temperature molecular dynamics simulations to navigate the conformational landscape. Once the researcher is satisfied, the small compound can be directly minimized with quantum mechanics algorithms, but if the size is too large a short MD simulation might be required instead, in this case to *relax* the structure.

In macromolecular studies where the conformational space is untreatable, experimental data is needed beforehand, like X-Ray Crystallography (XRC) or Nuclear Magnetic Resonance (NMR). These techniques provide density maps to which, with adequate refinement and post-processing using specialized software,<sup>112,113</sup> 3D structures are fitted. After validating the quality of the resulting model with ERRAT<sup>114</sup> or similar protocols, these are then usually submitted to specialized databases like Protein Data Bank (PDB),<sup>115</sup> from which the researchers can download the needed files for modeling the required system. Most of the time, files downloaded from PDB must be edited to remove experimental artefacts like duplicate positions of some atoms, protonate the residues at the desired pH by inferring the pKa of each one,<sup>116–118</sup> or reproduce the biological assembly of the structure.<sup>119,120</sup>

While the Protein Data Bank holds almost 140,000 structures readily available for everyone, not every protein is there. Sometimes, the structure is partially missing due to experimental limitations or even totally absent. In those cases, the model must be built from scratch. The folding problem is still unsolved and guessing the tertiary structure out of the sequence of amino acids is very challenging, but workarounds exist to work with available data. For example, if the protein that needs to be modeled has an already characterized homolog in the database, their sequence alignment can be used as a template to reconstruct a good structure. This technique is called homology modeling and its accuracy grows when multiple sequence alignment (MSA) are available. Since the final model is only an interpolation of closely related structures, some external valida-

tion is needed. In MODELLER, one of the most popular packages for homology modeling, several scoring functions to assess the quality are available. Additionally, a series of web services can be found to help in the task.<sup>114,121,122</sup>

Even with a proper protein structure, if the study features custom ligands, obtaining a good candidate for further steps in the protocol is not as simple as one might think. Assuming the required ligand structure is available, a regular protein-ligand docking simulation can provide an initial guess of the complex, but this is usually processed with long MD runs to assess the stability of the interactions. However, if the ligand is not known and must be designed from scratch, there is no consensus strategy. Two common approaches involve: (1) creating a library out of ligands that exhibit the needed chemical features using a reverse pharmacophoric approach, which then would be screened to assess their stability within the protein; (2) applying topology operators during the docking simulation itself using topology operators to dynamically build the ligand out of smaller fragments. The latter has been applied successfully in one of the programs presented in this thesis and will be discussed in chapter 6.

For larger scale systems like viral capsids or lipid bilayers, which can feature millions of atoms, building a starting structure can be daunting at first, but fortunately they all exhibit some kind of symmetry that can be used to assemble the full structure out of the involved subunits with specialized software.<sup>119,120</sup>

## 2.6 OPTIMIZATION METHODS

Most of the procedures used to identify physically sound models of a molecular system stand on finding the way to ally the exploration of wide search spaces (like introducing sensible changes in the 3D coordinates of a compound) and the adequate guiding variables (usually, the potential energy). In mathematics, this is territory of optimization problems for non-smooth surfaces.

An optimization problem consists of, given a function  $f$  that connects a set  $A$  to  $\mathbb{R}$  finding an element  $x_0$  in  $A$  such that  $f(x_0) \leq f(x)$  for all  $x$  in  $A$  (minimization), or such that  $f(x_0) \geq f(x)$  for all  $x$  in  $A$  (maximization).

$$\begin{aligned}
& \text{Given } f : A \rightarrow R \\
& \text{Sought : } x_0 \in A \text{ such that } f(x_0) \leq f(x), x \in A \text{ (minimization)} \\
& \text{or : } x_0 \in A \text{ such that } f(x_0) \geq f(x), x \in A \text{ (maximization)}
\end{aligned} \tag{2.5}$$

$f(x)$  is normally called the objective function, loss function or cost function for minimization problems, utility function or fitness function for maximization problems or, depending on the field, energy function or functional. However, all optimization problems can be expressed as minimization problems, negating  $f(x)$  if the problem is to be maximized.

The domain  $A$  of  $f$  is usually named the *search space* or the *choice set*, and each possible value of  $A$  is called a candidate or feasible solution.  $A$  is normally a subset of  $R^n$ , as defined by equality and/or inequality constraints. The optimization definition can be extended to be subject to inequality and equality constraints, expressed as:

$$\begin{aligned}
& \text{optimize}_x \quad f(x) \\
& \text{subject to } g_i(x) \leq 0, \quad i = 1, \dots, m \\
& \quad \quad \quad b_j(x) = 0, \quad j = 1, \dots, p
\end{aligned} \tag{2.6}$$

, where  $g_i(x)$  are the inequality constraints,  $b_j(x)$  are the equality constraints, and  $m$  and  $p$  are greater than 0. If  $m$  and  $p$  are equal to zero, the definition falls back to the unconstrained optimization problem.

Depending on the form of the function being optimized and the specified constraints, several categories emerge, like convex or nonlinear optimization problems. In convex optimization,  $f$ ,  $g$  and  $b$  are either convex (minimization) or concave (maximization). This includes linear functions, defining the field of linear optimization. Nonlinear optimization deals with functions that cannot be written as linear expressions, which usually makes the problem harder.

Solving this type of problems was initially studied by Fermat and Lagrange, who applied calculus-based formulae to identify optimum solutions. However, not all optimization problems can be solved analytically and, in fact, for some complex problems is usually easier (and faster) to compute numerical solutions iteratively until a convergence threshold is met. This approach was initiated by Newton and Gauss, and since

then many applicable algorithms have been devised throughout the latest decades. A few will be highlighted for illustrative purposes.

### 2.6.1 STEEPEST DESCENT AND CONJUGATE GRADIENT

To go down a smooth mountain, one simply takes steps in a direction towards the valley. There's a lot of possible directions, but skilled mountaineers usually take the fastest: the steepest side. Climbing down that mountain can be expressed as finding the minimum of a convex three-dimensional function, so figuring out which direction we should take in every step is a matter of finding the gradient of that function at that point.

A gradient is the  $n$ -dimensional generalization of a single-variable derivative, so instead of returning a scalar, it returns a vector. If derivatives gave us the rate of change of a function, gradients will tell the direction in which the function will experience the greatest change. In this three-dimensional problem, the gradient vector will point to the next step. By taking little steps in the direction pointed by the gradient, we will eventually get to the minimum.

This is what the steepest descent (SD) algorithm does, but it can progress very slowly in almost *flat* regions of the function. A similar method named conjugate gradient (CJ) uses a similar approach, but the search direction is computed in a smarter way, making sure that the direction is orthogonal to the previous step gradient and the current one. It requires more operations, but the performance towards the optimum is better and usually worthy.

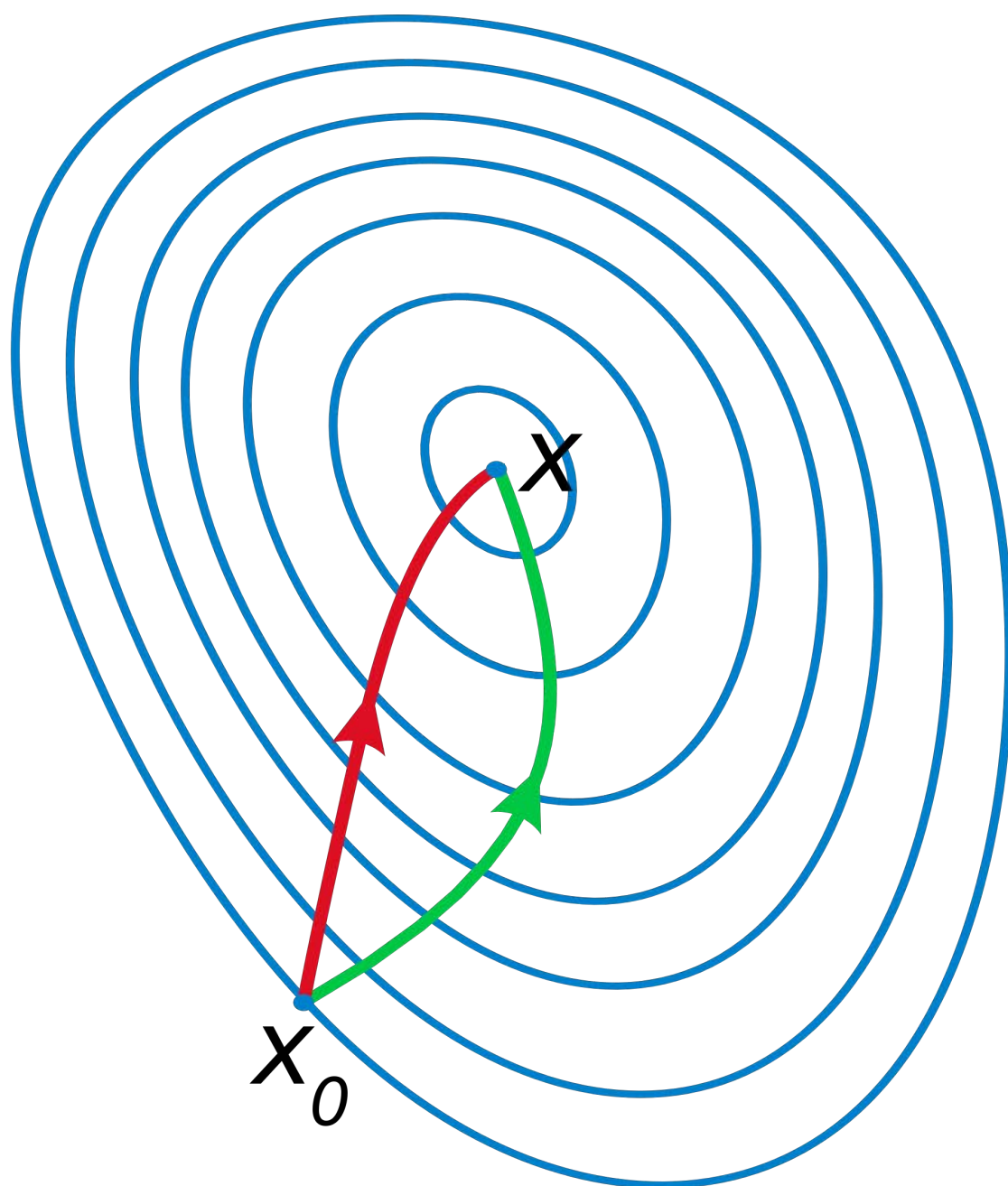
Since these methods only compute  $f(x)$  and  $f'(x)$  (first derivatives), they are called first-order methods.

### 2.6.2 THE NEWTON AND QUASI-NEWTON ALGORITHMS

Newton numerical algorithms are similar to SD and CJ methods, but compute an additional differentiation step to use the information provided by the curvature of the function. This makes each iteration more expensive, but with certain functions fewer iterations might be needed (see fig. 2.3).

Second derivatives can be generalized as Hessian matrices for problems of higher dimensions, but this can get very expensive to compute. As a result, several derived methods, called quasi-Newton methods, include alternative methods to compute it or supply equivalent information, like directly computing the inverse with numerical methods, updating it with successive gradient vectors... Due to its performance, one of the most popular quasi-Newton algorithms is the BFGS algorithm (for Broyden,<sup>123</sup> Fletcher,<sup>124</sup> Goldfarb<sup>125</sup> and

Shanno<sup>126</sup>) and its limited memory version L-BFGS,<sup>127</sup> widely used in energy minimization of molecules.



**Figure 2.3:** A comparison of gradient descent (green) and Newton's method (red) for minimizing a function (with small step sizes) starting with  $X_0$ . Global minimum is  $X$ . Newton's method uses curvature information to take a more direct route.



### 2.6.3 HEURISTIC AND META-HEURISTIC METHODS

While all these numerical methods are very different in nature, they still perform the same kind of tasks: exploration, evaluation and selection. This is, they generate a candidate solution (exploration), solve the equations and assess how far they are from the exit condition (evaluation). Selection is often so trivial in scalar functions that is not even considered as a separate step.

In a simplified example, where we must find  $f(x) = 0$  with  $f(x) = ax + b$ , *generating new candidate solutions* would simply consist of assigning new values to  $x$ . While this can be done randomly until the solution is found, it is usually more interesting to use a smarter approach. This what the gradients and hessian approaches provide: educated guesses towards finding the optima. However, they still require an equation to be available. When several variables are analyzed and the relationship between them is not differentiable or, simply, unknown, other type of algorithms must be employed, like heuristic or meta-heuristic. This kind of methods make very few assumptions about the problem being solved, making them suitable for a variety of optimization areas.

#### 2.6.3.1 MONTE CARLO METHODS

Monte Carlo methods are useful for studying problems that are characterized by a huge number of degrees of freedom but can be interpreted probabilistically. Since the expected value of an integral can be approximated by the empirical mean of a random sample, these methods allow to obtain numerical results by randomly sampling the search space. In the Metropolis variant, the sample is refined iteratively with random modifications that are either accepted or rejected depending on the new value of the sample or a random acceptance ratio.

For example, to minimize the potential energy of a molecule, random states can be generated by introducing small perturbations to the atomic positions that follow a Boltzmann distribution. The energy of the new states is evaluated and either accepted or rejected by comparing their energies with current mean of the sample. For example, those with smaller energies are usually included and accepted in the ensemble. For those with higher energies, they can still be included with some probability that depends on the chosen acceptance ratio. Being a Markov chain, the probability distribution for the next iteration will be reparametrized with the state of the current sample and the process will continue iteratively until convergence.

### 2.6.3.2 EVOLUTIONARY ALGORITHMS

Evolutionary algorithms (EA) can be explained as an extension to Monte Carlo's: they also employ random generation of solutions as starting points, but following iterations employ biology-inspired heuristics to localize next candidate solutions. In each iteration, the *population* of feasible solutions (individuals) are evaluated in the optimization environment, and each one is assessed a fitness score. Like in the Evolution theory, only the fittest will be allowed to survive (included in the sample for the next iteration).

Genetic algorithms (GA) are a special type of EA that implement *evolutionary* heuristics inspired on chromosomic changes. By mimicking chromosomes during the meiosis, candidate solutions can exchange some of their variables (mating or recombination), and some can experience a random change in one or more variables (mutation). By iterating over this *reproductive* cycle, fitter and fitter solutions will be obtained.

Besides EA, new metaphor-inspired algorithms are constantly developed. Starting in 1983 with Kirkpatrick's Simulated Annealing (SA),<sup>128</sup> it began to grow in the 90s with Ant Colony Optimization<sup>129</sup> and Particle Swarm Optimization,<sup>130</sup> and exploded in the 2000s and 2010s. Last developments have been attracting criticism because they seem to hide the lack of novelty behind an attractive metaphor.<sup>131–134</sup>

### 2.6.4 MACHINE LEARNING

Artificial Intelligence and Machine Learning are very popular computer science fields these days. Globally, they are algorithms that can *learn* from their own *experience* by extracting patterns and relationships out of the supplied data. They can be studied as non-linear statistical data modeling tools.

One of the hottest branches of Machine Learning are Artificial Neural Networks and, especially, Deep Learning. The implemented algorithms in these categories mimic the way neurons work in the brain. Like all mathematical functions, each *neuron* produces an output that depends on the input. Many neurons are grouped together in layers and these layers are concatenated, having the output of one layer fed as the input of the next one. Layers can back-propagate, and modify the input of previous layers, like the feedback mechanism of the brain. Ultimately, this construction generates a huge set of self-adjusting equations that can optimize wide ranges of observations. For example, they are actively used in speech recognition, computer vision or artificial intelligence. The excitement produced by its success in other areas made it permeate towards some areas of science where its application is controversial and less *fancy* algorithms like traditional statistic methods are even better performers.<sup>135</sup>

## 2.7 MULTI-OBJECTIVE OPTIMIZATION

Usually, our minds are wired to think in scalar functions and values. This is, functions that return scalar magnitudes or *single values*. If  $f(x)$  points to a scalar space, selecting  $f(x_0)$  vs  $f(x_1)$  is just a matter of seeing which value is smaller (minimization) or greater (maximization). However, if the function returns  $n$ -dimensional data, navigating towards the optimum is not so intuitive. Since there is more than one target value, conflicting decisions might arise.

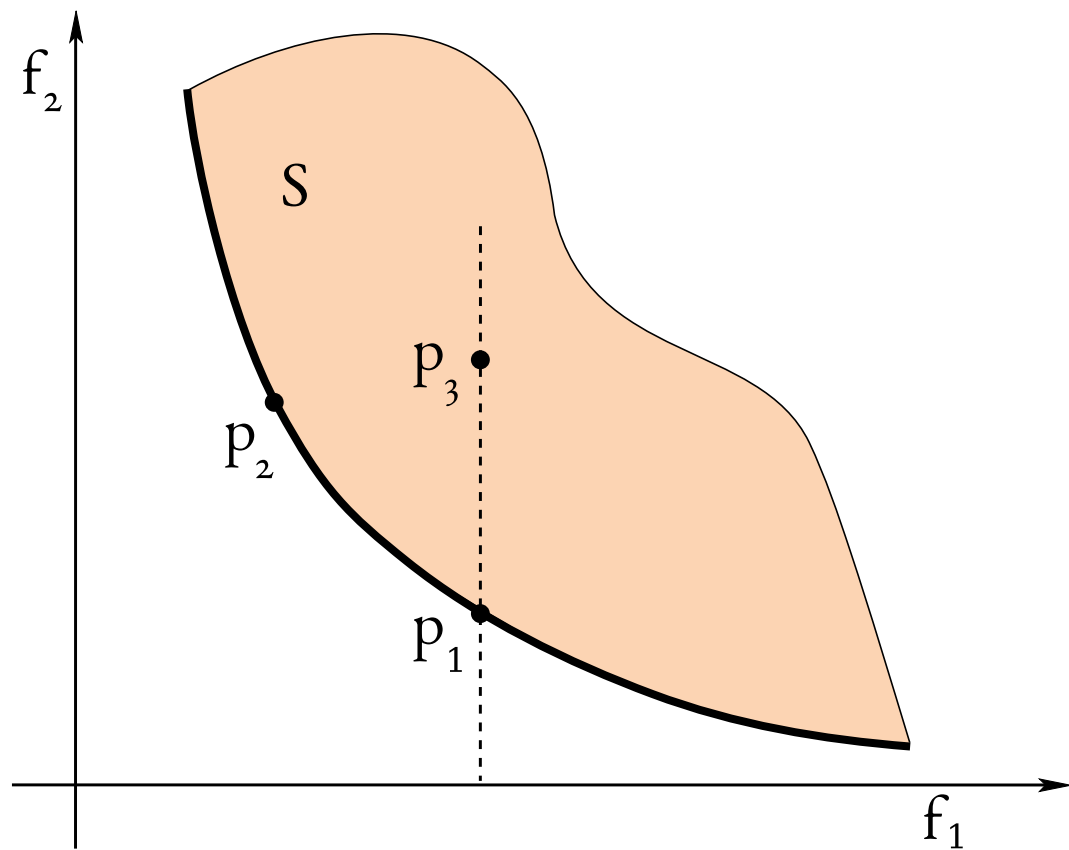
Say we need to minimize a function that takes a vector in  $R^3$  and returns another vector in  $R^3$ . Reaching the origin by minimization would mean obtaining a vector with all three values equal to zero. What if the first element in the vector is smaller, but the second is greater? One possibility is to construct a general function out of those functions, like the Euclidean distance until the origin. In more complex cases, simple weighted linear sum might work, but the weights must be carefully chosen for each case; otherwise, convergence problems might appear.<sup>136</sup>

One alternative which does not involve a dimensionality reduction ( $R^3$  to  $R$  in the previous example) is the Pareto optimality criterion, enunciated by Wilfried F. Pareto in his studies of income distribution in the 1900s. It is based on Pareto *dominance*: a solution  $a$  is set to dominate solution  $b$  if it solves at least one of the objectives better than  $b$ , without losing to  $b$  in any of the remaining objectives.<sup>137</sup> By enumerating a high number of solutions and comparing them in terms of Pareto-dominance, a reduced set of non-dominated solutions can be found (see fig. 2.4). When no more non-dominated solutions can be found, that set is said to be Pareto-optimal and constitutes the Pareto front: the solutions to the problem.<sup>¶</sup>

Under this scheme, finding the solutions to a multi-objective problem is only a matter of increasingly enriching the Pareto front with non-dominated solutions. Without defining the importance of each objective, all of them will be equally good solutions. In other words, multi-objective optimization algorithms do not propose a single optimum solution, but a set of good trade-offs between the variables under consideration.<sup>138</sup>

---

<sup>¶</sup>Optimization processes like this are more common than they appear. At the supermarket, all clients decide on the trade-off between price and quality every day. Normally, humans solve this by setting a cutoff on one of the variables. For example, a maximum budget is set. However, if all possibilities are considered, the resulting solutions would range from the cheapest possible product to the most expensive one, including all the good enough (non-dominated) combinations in between. However, if a new product is added to the catalog and is cheaper than its competitors without a decrease in quality, that new product will dominate all other products with the same quality but higher price.



**Figure 2.4:** For a two-dimensional problem where  $f_1$  and  $f_2$  must be minimized, the Pareto front can be identified with a convex curve. In this example,  $p_1$  and  $p_2$  are non-dominated solutions that are part of the Pareto front (wide line).  $p_3$  is dominated by  $p_1$  and  $p_2$  because  $p_1$  has a lower value in the  $f_2$  axis without worsening the value in the  $f_1$  axis, and  $p_2$  has lower values in both axes.



# 3

## Objectives

**M**ULTISCALE molecular modeling employs different modeling techniques and levels of theory, per definition. However, resorting to such a vast variety of software tools means they do not usually play well together. Being conceived by teams with different background and focus, this end up resulting in three common symptoms:

- Most molecular modeling tools are designed as standalone pieces not meant to be part of broader, multistage protocols.
- They present unintentionally opinionated abstractions and problem-solving strategies that force users to recontextualize their problem for each tool.
- The files required are almost never compatible, which results in non-trivial format conversions or manual input, especially if data exchange is needed.

Subsequently, when a researcher faces a multiscale protocol, a series of technical issues unrelated to the scientific problem arise: files are not properly converted, software dependencies are not updated, the operating system is not supported anymore... Molecular modeling is difficult enough by itself; there is no need to put additional barriers in the way.

The main motivation behind this thesis is to provide new software solutions to make technical and scientific barriers easier to overcome when it comes to molecular modeling and multiscale protocols. Several tools will be presented in the next chapters, each focusing on a specific part of the multiscale funnel. Two of them constitute the main projects of this thesis:

- GaudiMM, described in chapter 4, is a multi-objective optimization platform to provide reasonably

sound models meant to be used as starting structures for subsequent stages down a multiscale protocol.

- Tangram suite, described in chapter 5, is a collection of graphical interfaces for UCSF Chimera to bridge diverse molecular modeling tools in a single, intuitive user experience. This chapter also includes command-line utilities that were started as helper tools and ended up becoming independent projects on their own.

Finally, in chapter 6, a collection of illustrative cases will be described in detail to prove their usage and applicability. These include toy examples that showcase the potentiality of GaudiMM, and a detailed computational insight on the counter-intuitive experimental observations found in multivalent enzyme inhibition studies.

# 4

## GaudiMM

**T**HERE is an implicit restriction in multiscale approaches due to their own design. They are based on a sequential series of steps, which are chained one after another to answer the initial question. Each step must be resolved separately, which can potentially become a bottleneck or even a blocking step if the results are not successfully obtained.

Instead of forcing a sequential protocol around a complex molecular problem, an alternative approach could be devised. If the panoply of existing modeling methods could be recruited on demand to work simultaneously on the same study, all of them could contribute to solve the problem, multiplying their strengths and compensating their weaknesses. Building this feature set into a robust and flexible platform would be very desirable for drafting molecular hypotheses and sketching proofs of concept.

GaudiMM is here presented to become such a platform. It takes the expressiveness and flexibility of Python to create a molecular design platform with unprecedented versatility. The rationale behind its concept can be summarized in three points:

1. Its modular implementation allows to encapsulate separate methods in isolated entities that can work together through a well-defined programmatic interface, which also allows fast development of new extensions.
2. It makes a clear distinction between the three main stages of any optimization process (exploration, evaluation and selection), which suggests a flexible way of rationalizing molecular modeling problems.
3. It does not require prior knowledge of the importance of the variables that affect the system thanks to its multi-objective optimization capabilities.



As a result, solving a molecular modeling problem is only a matter of choosing the appropriate modules in terms of which variables should be explored (cartesian coordinates, chemical spaces...) and which properties should be measured (geometries, energies...). In some cases, some rigor can be sacrificed in honor of obtaining good enough results to start working with. In other cases, the combination of methods will work synergistically towards the design of a novel methodology.

Following sections will describe: (1) the algorithmic and (2) implementation details of the platform, (3) how different combinations of modules allow diverse molecular modeling tasks, and (4) how to analyze the proposed results.

**Table 4.1:** GaudiMM: technical datasheet.

GAUDI MM	
<i>Description</i>	A modular optimization platform for molecular design
<i>Requirements</i>	Python, UCSF Chimera, OpenMM, IMP, DSX, ProDy...
<i>License</i>	Apache 2
<i>Download</i>	<a href="https://github.com/insilichem/gaudi">github.com/insilichem/gaudi</a>
<i>Documentation</i>	<a href="https://gaudi.readthedocs.io">gaudi.readthedocs.io</a>
<i>Citation</i>	J. Comput. Chem. 2017, 38, pp 2118–2126. DOI: 10.1002/jcc.24847

## 4.1 ALGORITHMIC DETAILS:

### MULTI-OBJECTIVE OPTIMIZATION & NSGA-II

GaudiMM is built on top of a multi-objective genetic algorithm (MOGA), NSGA-II, developed by K. Deb.<sup>139</sup> It has been thoroughly tested and benchmarked in well-characterized multi-objective problems and is considered a prototypical MOGA.

As other optimization methods, this algorithm can be described in three main stages (exploration, evaluation and selection) that are executed iteratively until an exit condition is met (usually, convergence or maximum steps). Generating new candidate solutions or individuals is considered within the *exploration* stage, and can be achieved by random attribute assignation or combining previously existing individuals. In the *evaluation* stage, the candidates are assessed with different functions or objectives, each returning a scalar that represents a fitness score for that objective. Finally, the selection stage collects all the individuals and compares their vectorial scores to select the best individuals according to the Pareto dominance criterion (see chapter 2).

In more detail, NSGA-II starts with the generation of a random set of potential solutions (*individuals*) which

comprise the so-called *initial population*. This first set of individuals is then evaluated with one or more *objectives* and each individual is assigned a *fitness* score vector, the elements of which are the result of those cost functions. At this point, a small subset of the population is submitted to a round of random modification of parameters (*mutation*) or exchanging some of their attributes (*recombination*), and are then assessed by the same cost functions. Being random, the results of these variations can be better or worse than their preceding counterparts (parents). Finally, both the offspring and the parental generation ( $\mu + \lambda$  strategy) compete in the selection tournament, which will rule which ones will replace the initial population. After a number of iterations, the initial population will have evolved and, eventually, will end up providing reasonable solutions to the problem that represent a compromise between the analyzed variables (see fig. 4.1).

## 4.2 IMPLEMENTATION

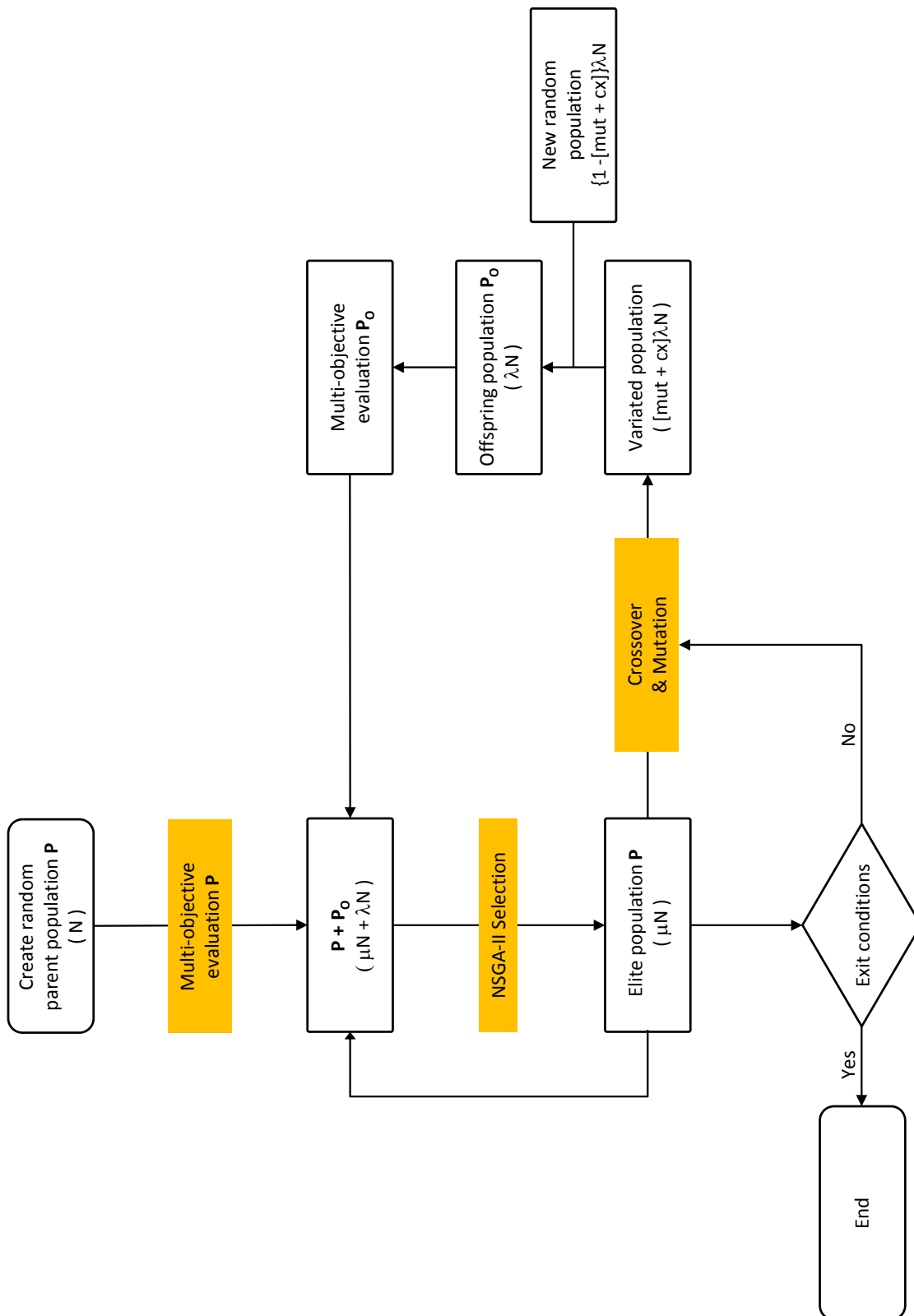
The underlying algorithm is very present in how GaudiMM has been implemented and how it is used. Learning to model with GaudiMM means having a clear understanding on the different stages involved in the algorithm, specially exploration and evaluation.

### 4.2.1 OF INDIVIDUALS AND GENES: THE EXPLORATION STAGE

The initial step of all the iterations in the algorithm is the exploration, which is responsible for the generation of new candidate solutions. A candidate solution is defined by a list of attributes, each representing the state of a molecular property. Generating new solutions simply involves changing the value of one or more attributes in that list.

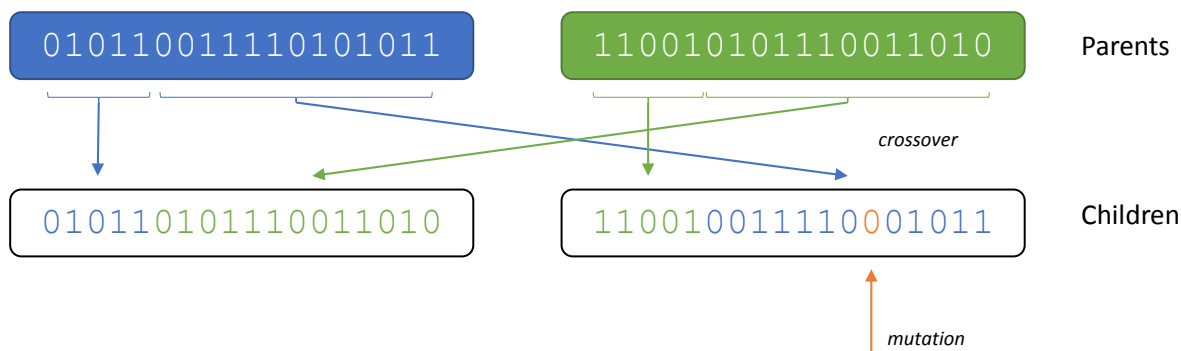
Since GaudiMM is based on a genetic algorithm, the implementation follows the same biologist terminology. In GaudiMM any candidate solution is encoded in a special object called `Individual`. All `Individual` objects in the simulation are defined by the same high-level attributes, which are called *genes*. In the same fashion, the state of each gene is defined by its *allele* attribute. Depending on the gene, the allele can be a list of numbers, a path to a file, a matrix...

For example, a typical optimization problem is finding the dihedral torsion that gives the minimum energy in the ethane molecule. The `Individuals` featured in this example would only need exploring a single variable, the torsion angle of the C-C bond, with values ranging from 0 to 360°. In GaudiMM-speak, the gene would be the bond *rotator* and the allele the different angles.



**Figure 4.1:** Flowchart of the modular NSGA-II multi-objective genetic algorithm (MOGA) implemented in GaudiMM.  $N$  is the number of individuals in the initial population  $P$ . Values  $\lambda$  and  $\mu$  are related to the number of children produced at each generation and the number of individuals selected for the next generation, respectively. Together they control the offspring population size,  $P_0$ . Constants  $mut$  and  $cx$  are the probabilities associated to mutation and crossover.

The key part of genetic algorithms is the implementation of variation operators as part of the exploration stage. Instead of merely trusting randomness, existing solutions are combined in hopes of obtaining a better child solution. These two operations are called *mutation* and *crossover* or mating, mimicking what happens in the cell nucleus at the chromosomal level.



**Figure 4.2:** Mutation and crossover operations introduce variability in the parental population.

Taking all these requirements into account, genes in GaudiMM are programmatically defined by four functions (*express*, *unexpress*, *mutate* and *mate*) and an attribute (*allele*). Additional methods and attributes can be defined to support these required elements, if needed. Since each gene is a clearly separate entity, the *Individual* object can feature more than one gene, and one gene can be present more than once with different parameters.

This adds an unprecedented versatility when configuring a GaudiMM calculation: the user can decide which molecular features must be explored for every case. For conformational searches it might be enough with the *Torsion* gene, but for protein-ligand docking the *Search* gene will be required too. Additionally, if the built-in genes do not fulfill the requirements of the simulation, new ones can be written and added to GaudiMM thanks to its modular architecture and well-defined programmatic interface.

This is, genes are more than simple *allele* attribute holders: they are high-level abstractions of operators that can make reversible changes in a molecule based on the value of its allele. Like in Biology, changes in the allele are only visible if the corresponding gene is being *expressed*. In those terms, GaudiMM genes encompass both the allele and the expression mechanism. In the previous example, when the allele changes the torsion gene needs to update the coordinates of the atoms affected by the dihedral rotation, and only those. To make changes consistent, it might also need to *unexpress* or undo those changes to the original state. These changes can happen in the topology or coordinates of an associated molecule.

#### 4.2.1.1 TOPOLOGY MODIFIERS

Genes that fall in this category perform modifications on the atoms that conform the molecular structure and/or their connectivity. For example, they could increase the length of a ligand linker, change the metal element of a metallic cofactor or mutate some residues in a peptide sequence.

- **MOLECULE.** It is the main gene, as it will be used to load molecular structures from files (PDB, Mol2, XYZ or any other input format supported by UCSF Chimera). All other genes depend on the initial topology and coordinates provided by one or more Molecule genes. In addition to loading files, the path parameter supports loading from a directory, whose contents determine the final behavior:
  - If the directory contains molecule files, the allele will be set to one of them randomly for each individual. This allows GaudiMM to deal test a library of compounds against certain criteria; i.e. virtual screening.
  - If the directory contains subdirectories which, in turn, contain molecules files, the gene will sort those subdirectories by name and then pick one molecule from each, in that order. The chosen molecules will constitute the allele and will be chained linearly as specified in the accompanying meta file, which lists the serial number of the potential donor and acceptor atoms.
- **MUTAMERS.** Given a residue position in a protein structure, it can replace its sidechain to any other natural amino acid specified in the configuration. Useful to study site mutations.

#### 4.2.1.2 COORDINATES MODIFIERS

Genes that fall in this category only alter the positions of the atoms involved in a molecular structure. They can modify the full structure, like a rigid translation or rotation of the molecule, or only a part, like the sidechain orientation of a protein residue.

- **TORSION.** It helps explore the flexibility of small molecules by performing bond rotations in the selected Molecule objects, if they exhibit free bond rotations.
- **SEARCH.** It performs rigid transformations on Molecules (translation and rotation). A radius parameter can be set to limit the search sphere range. If the radius is zero, the molecule will not be translated but can freely rotate around the anchor atom, which is useful for covalent bond emulation.

- **ROTAMERS.** It allows to explore side-chain conformations in protein residues by applying Dunbrack's<sup>140</sup> or Dynameomics<sup>141</sup> rotamer libraries.
- **NORMALMODES.** Given a `Molecule` object, it calculates normal modes with elastic network methods and applies the resulting collective motions as possible variants of the initial coordinates set.
- **TRAJECTORY.** Given a molecular dynamics trajectory file, it can retrieve random frames and apply the resulting coordinates to any `Molecule` object.

**Table 4.2:** List of genes implemented in GaudiMM.

NAME	DESCRIPTION	DEPENDS ON
Molecule	Load and build structures	UCSF Chimera
Rotamers	Explore side chain flexibility	UCSF Chimera
Mutamers	Explore mutation of residues	UCSF Chimera
NormalModes	Explore collective motions	ProDy <sup>142</sup>
Search	Translation and rotation of Molecules	UCSF Chimera
Torsion	Dihedral rotation of bonds	UCSF Chimera
Trajectory	Load frames from MD trajectories	MDTraj <sup>143</sup>

#### 4.2.2 OF ENVIRONMENTS AND OBJECTIVES: THE EVALUATION STAGE

After generating candidate solutions, these must be evaluated with the optimization criteria. In genetic algorithms, this is usually called assessing the fitness of the individuals: fitter individuals are more qualified to survive in the environment.

Mimicking these concepts, the GaudiMM implementation creates an `Environment` object that list the optimization criteria, each represented by an `Objective` entity. Objectives are also independent units that can be instantiated multiple times in the same `Environment`, but the defined interface is simpler than in genes: a *weight* attribute defines the optimization type (maximization or minimization), and a function named `evaluate` that takes an `Individual` object and returns a numerical value as result. What the `evaluate` function does behind the scenes does not actually matter as long as a number is produced: calculating a distance between two atoms, retrieving a parameter from a database, computing the potential energy with an external MM library...

As a result, GaudiMM ships with a rather diverse set of objectives, combining 3<sup>rd</sup> party packages and custom developments in the same distribution. Together they cover all kinds of energetic, geometric and spatial measurements, allowing to use different levels of theory at the same time in a seamless workflow. Any geometric or energetic parameters that could describe a molecular system can be used as objectives to drive the GA exploration. This allows us to turn the tables on routine protocols based on computing energetic opti-

mizations and then analyzing the results in hopes of finding a suitable model that fits the intended restraints; i.e. those same analysis tools can guide the optimization process from the beginning.

#### 4.2.2.1 GEOMETRY MEASUREMENT

- **ANGLE.** Given three atoms, this objective calculates the angle between those. By minimizing the difference between the measured angle and the target one, the final angle can be optimized. It will calculate the dihedral if four atoms are specified.
- **DISTANCE.** If two atoms are provided, this objective calculates the distance between. By minimizing the difference against a target value, the structure can be optimized to fulfill that requirement. It also supports calculating distances to groups of atoms by taking the centroid of the group.
- **INERTIA.** This objective calculates the inertia tensors of two structures and returns the sine of the smallest angle formed between any of the possible pairings. It can be useful to align ligands along the major axis of a protein.

#### 4.2.2.2 SPATIAL MEASUREMENT

- **SOLVATION.** Solvent-Accessible Surface Area (SASA) and Solvent-Excluded Surface Area (SESA) are two common techniques to describe the solvation of a structure. It can be used to optimize structures in terms of exposure of inside pockets or their folding. By maximizing SASA or SESA, the structure will tend to open up; by minimizing those values, the trend will be towards a more compact conformation.
- **VOLUME.** This objective calculates the volume occupied by a structure. It does so by computing the solvent-exposed surface of the structure, which is then considered as a polyhedron of thousands of triangular faces.

#### 4.2.2.3 ENERGY CALCULATION

- **DSX.** DrugScoreX is a knowledge-based docking scoring function developed by Neudert & Klebe.<sup>84</sup> It is specially designed to compute interaction energies between protein structures and small compounds. This objective is a Python wrapper around the DSX executables and input files.
- **ENERGY.** This objective allows to calculate the potential energy of a structure with the Molecular Mechanics force fields implemented in OpenMM. Parameters must be provided for custom residues.

- **LIGSCORE.** Another docking scoring function developed by Sali<sup>144</sup> which allows to obtain protein-ligand interaction energies. While the parent project, IMP,<sup>145</sup> is a C++ project with Python bindings, the LigScore function is only exposed through an executable. This objective can call that binary and parse the resulting energies from the output.
- **VINA.** AutoDock Vina<sup>13</sup> is a popular open-source package to perform protein-ligand docking. This objective calls the Vina executable in score-only mode to calculate the interaction energies between a protein and a ligand.
- **GOLD.** This commercial software suite is one the most used solutions to calculate accurate docking poses. With this objective, all the scoring functions exposed in GOLD<sup>146</sup> can be used as guiding evaluators in GaudiMM: PLP, GoldScore, ChemScore... License is needed for this to work.
- **NWCHEM.** This objective provides a way to run quantum mechanics calculations in this popular open-source software suite.<sup>147</sup> Provided a template input-file, this objective will insert the appropriate coordinates, charge and multiplicity. While all methods implemented in NWChem are potentially usable, only semi-empirical ones are recommended in terms of speed; specially for large structures.

#### 4.2.2.4 HIGH-LEVEL CHEMICAL DESCRIPTORS

- **CONTACTS.** This objective can calculate two type of distance-based energy descriptors. When the *hydrophobic* mode is chosen, this objective will maximize potentially attracting interactions between close enough atoms by applying a Lennard-Jones-like scoring function. If the *clashes* mode is chosen, it will minimize the steric hindrance of the structure by minimizing the volumetric overlap of the Van der Waals spheres of atoms that are too close.
- **HBONDS.** This objective uses geometrical criteria to calculate the number of hydrogen bonds between potential donors and acceptors.
- **COORDINATION.** By applying a type of computer vision algorithm called Point Set Registration, this objective can identify potential coordination geometries around a metal center. It returns the RMSD similarity between the first coordination sphere and the ideal polyhedron: the lower the value, the better the geometry.

#### 4.2.3 OF TOURNAMENTS AND TRADE-OFFS: THE SELECTION STAGE

Once the Individuals have been assigned a fitness score, these values must be compared to assess how good of a solution they make. In multi-objective optimization problems there is no *best* solution in usual terms.



**Table 4.3:** List of objectives implemented in GaudiMM.

NAME	DESCRIPTION	DEPENDS ON
Angle	Optimize angle of three atoms, or dihedral of four atoms	UCSF Chimera
Contacts	Minimize steric clashes, maximize hydrophobic interactions	UCSF Chimera
Coordination	Optimize coordination geometry of metal center	In-house <sup>148</sup>
Distance	Optimize distance between two or more atoms	UCSF Chimera
DSX	Docking scoring function	DrugScoreX <sup>84</sup>
Energy	Minimize molecular mechanics potential energy	OpenMM <sup>11</sup>
HBonds	Detect hydrogen bonds	UCSF Chimera
Inertia	Align axes of inertia of two or more molecules	In-house
LigScore	Docking scoring function	IMP <sup>144</sup>
NWChem	Launch NWChem QM calculations	NWChem <sup>147</sup>
Solvation	Measure solvent accessible solvent area	UCSF Chimera
Volume	Measure volume occupied by molecule	UCSF Chimera

Instead, a set of trade-offs between the involved (and usually conflicting) variables is required. NSGA-II solves this by following the Pareto optimality criterion explained in chapter 2, which will iteratively enrich the Pareto optimal set with the *best* candidates of the population. However, when more variables (objectives) are added to the optimization, the Pareto front grows in dimensionality and enriching the Pareto optimal set can get difficult. Deb et al. do not recommend more than three objectives for NSGA-II, but several extensions to the algorithm (MONSGA-II, NSGA-III) exist to improve this situation. Higher dimensionality will also involve a larger number of possible solutions (even when Pareto-optimality is reached).

To ensure a rich Pareto front in constructed, NSGA-II includes a crowding parameter, and GaudiMM provides structural similarity comparisons when scores are very close to each other, resulting in a good compromise between diversity and number of solutions proposed.

#### 4.2.4 THE CODE BEHIND: PYTHON AS GLUE

GaudiMM started by hooking `deap`<sup>149</sup> evolutionary algorithms into UCSF Chimera. Using Python as the main language allowed to design a modular architecture that conceptually emphasizes the different stages of optimization, while focusing on the reutilization and addition of existing codebases. It is difficult to think of a different language that could have provided a working proof-of-concept in that little time.

All the code is object-oriented and features a well-documented programmatic interface, alleviating the process of writing new genes and objectives. The educational value of this technical decision was not obvious until degree and master students began to collaborate in the project as part of their final dissertation (see appendix B).

After years of development, UCSF Chimera is still the main library behind the scenes. In fact, to our knowledge, GaudiMM is one of the few projects that relies on it for calculation purposes and not strictly for visualization. This interactive 3D viewer offers lots of analysis tools and robust molecular abstractions that allowed us to implement most of GaudiMM genes and objectives in few lines of code. However, everything has a price and UCSF Chimera was not designed to be used as a library in other projects; instead it expects external projects to be executed within UCSF Chimera interface. To overcome this limitation, a separate package named PyChimera was developed. With PyChimera, other Python libraries can be used together with UCSF Chimera, which allowed to reuse code from other projects in GaudiMM. That way, MM energies can be computed with OpenMM, Normal Modes Analysis calculated with ProDy, and more (see tables 4.2 and 4.3). Further examples of integration are given in chapter 5, where PyChimera has been instrumental in the development and distribution of new graphical interfaces.

### 4.3 USAGE: FROM RECIPES TO MOLECULAR MODELING TASKS

GaudiMM does not make any assumptions on the molecular modeling task to be performed. Setting up a calculation is a matter of choosing the appropriate genes and objectives (see tables 4.2 and 4.3). Like ordering off a menu, each combination of those can be considered a *recipe*. Since each gene and objective is a separate module that can be instantiated as many times as needed, this confers lots of flexibility.

All calculations normally start by configuring one or more `Molecule` genes to load the structures under study. On top of the `Molecule` genes, the user can choose additional genes to introduce certain types of variability, like the internal flexibility of a small compound (`Torsion` gene) or 3D spatial exploration (`Search` gene). Additional genes can target one or more `Molecule` genes, either partially or globally.

The set of genes will simply generate different variants of the starting model, which can potentially include non-feasible structures. As a result, after choosing which genes to apply, the user must decide which variables will guide the optimization of the structure by choosing one or more objectives. For example, it is common to request a `Contacts` gene to minimize the steric clashes that can arise from moving a small molecule around a bigger one. If more requirements are needed, like maximizing the number of hydrogen bonds, the corresponding objectives can be added too.

It must be remembered that the objectives simply assign a score to a candidate solution. It is up to the selection step to favor the promotion of candidates that satisfy the optimization criteria (i.e., low number of clashes with good hydrogen bonds) and discard those that do not. As an example, a trivial molecular

modeling task will be explained.

### 4.3.1 TUTORIAL: OBTAINING A CYCLIC ALKANE

Building a cyclic alkane seems like a trivial task, but depending on the number of bonds involved, it can soon become a tedious process. With GaudiMM, it can be done in a single calculation: only two genes and two objectives are needed. The hypothesis is that there exists a set of dihedral torsions that can connect the ends of a linear decane without steric clashes.

First, a starting 3D structure is needed. For practical purposes, a linear decane can be built directly with UCSF Chimera by issuing the command ``open smiles:CCCCCCCCC`` and saved as a Mol2 file with ``write decane.mol2``. This file can be loaded in GaudiMM with a `MoLecule` gene by setting its location as the value of the argument `path`.

The second gene is `Torsion`, which will detect rotatable bonds in the decane and apply the rotations instructed. However, the `Torsion` gene does not know nor care about steric clashes or minimum distances needed for a covalent bond. It will simply generate arbitrary sets of rotations.

Detecting one that can provide a structure compatible with a cyclodecane is responsibility of the evaluation stage. For example, to discard candidates with bad steric clashes, these should be minimized with the `Contacts` objective. In order to locate a structure compatible with a cyclodecane, a second objective is needed: a `Distance` minimization between the end carbon atoms of the decane. By setting the target distance as 1.5 Å, the linear decane will be forced to anneal itself.

This configuration is enough to run a simple multi-objective optimization process. Since the `Contacts` objective only analyzes the volumetric overlap of the van der Waals spheres of the atoms and UCSF Chimera provides a basic library of van der Waals radii for all elements, no additional parameterization is needed. After running the program over this input file (see fig. 4.3), GaudiMM will generate solutions compatible with the satisfaction of both criteria, which can be assessed with the accompanying graphical interface (see section 4.4).

However, after the first attempt, the user might realize that some results are indeed decane conformations whose ends are 1.5 Å apart, but not in the expected orientation. In other words, they do not respect the  $sp^3$  tetrahedral geometry. To fix it, an additional `Angle` objective set to match  $109.5^\circ$  between the end atoms and one of their neighbors would suffice. The final recipe can be consulted in table 4.4.

```

# GaudiMM input files are formatted in YAML
# This is a comment
output:
  path: ./results
  name: cyclodecane-experiment

genes:
- name: Decane
  module: gaudi.genes.molecule
  path: decane.mol2

- name: Torsion
  module: gaudi.genes.torsion
  target: Decane
  anchor: Decane/1

objectives:
- name: Clashes
  module: gaudi.objectives.contacts
  which: clashes
  weight: -1.0 # minimize
  probes: [Decane]
  radius: 5.0

- name: Distance
  module: gaudi.objectives.distance
  weight: -1.0
  probes: [Decane/1]
  target: Decane/10
  threshold: covalent # 1.5 A for C-C

- name: Angle
  module: gaudi.objectives.angle
  weight: -1.0
  threshold: 109.5
  probes: [Decane/1, Decane/10, Decane/9]

```

**Figure 4.3:** Minimal GaudiMM input file for the optimization of linear decane into a cyclodecane.

**Table 4.4:** Final recipe for the cyclodecane example.

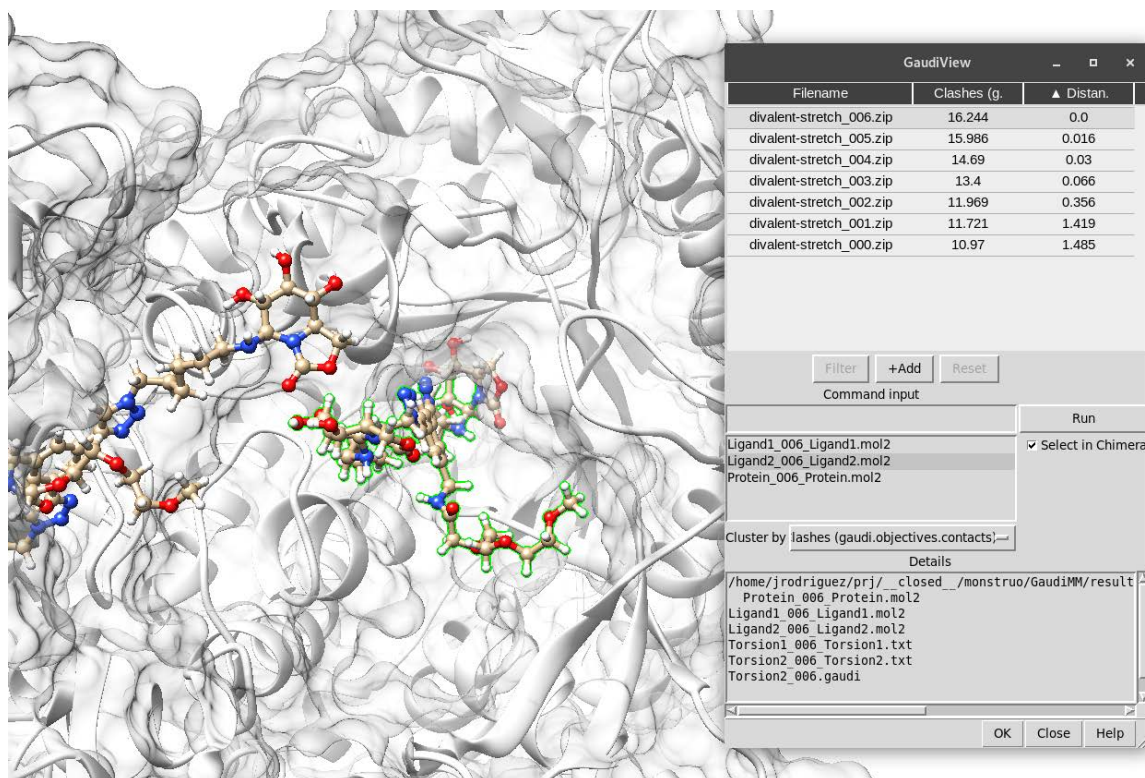
GENES	
Molecule	Load the starting linear decane structure
Torsion	Explore rotations in rotatable bonds
OBJECTIVES	
Contacts	Minimize steric clashes
Distance	Bring terminal carbon atoms within 1.5 Å
Angle	Force terminal carbon atoms to match a 109.5° angle

Albeit useless, this toy example illustrates the flexibility of the paradigm proposed in GaudiMM. For more practical use cases, please refer to models detailed in chapter 6.

#### 4.4 ANALYZING THE RESULTS OF MULTI-OBJECTIVE OPTIMIZATION

In multi-objective optimization (see section 2.7), ultimately choosing which solution is the *best* is up to the decision maker: the researcher. Some strategies to make that decision involve reducing the fitness vector to a scalar using an adequate function. However, since that function is usually not characterized in tentative molecular modeling tasks, a UCSF Chimera extension has been developed GaudiView along GaudiMM to aid in that decision in a more interactive manner.

GaudiView will list the proposed solutions along with the fitness of each objective in spreadsheet-like dialog. Upon clicking each entry, the UCSF Chimera canvas will load and render a 3D interactive depiction of the structure. The table can be sorted by columns and filtered by threshold criteria, which can reduce the complex surface of the Pareto front to the *interesting parts* (according to the decision maker) dynamically. Since the renderization of molecular structures is delayed until the corresponding rows are selected, the interface can show thousands of results with low memory usage. Integrative analysis can be performed on the flow with other tools included in UCSF Chimera thanks to the built-in command line widget, which is executed on each selection change event. Appendix D contains more details on this tool.



**Figure 4.4:** Analysis of a GaudiMM dual docking calculation with GaudiView. Each row of the table represents one candidate solution that will be depicted in the 3D canvas upon selection.

## 4.5 CONCLUSIONS & FURTHER WORK

The development of GaudiMM was motivated by the need of applying simple descriptors in complex biomolecular systems featuring residues beyond the natural amino acids: metallic cofactors, oligosugar-derivatives and partially characterized organic molecules. The main idea was to at least have *some* results around a hard-to-model structure, instead of saying that it could not be done. Even with low accuracy methods, GaudiMM soon started to prove that the approach is good enough to provide starting points valid for further refinement and processing with more accurate methods; in other words, GaudiMM is a good entry point for multiscale protocols. This and other examples of its potential applications, including how it has been used in real cases of research, will be detailed in chapter 6.

These observations have made clear that GaudiMM provides a mental framework suitable for implementing proof-of-concept multiscale protocols and explaining basic concepts of molecular modeling to newcomers in the field.

That said, there is room for improvement in the performance area. Genetic algorithms are easily paralleliz-

able by design, but depending on UCSF Chimera for most functions means that communication across processes could be expensive in terms of memory usage and synchronization overhead. Since the calculations rarely involve more than a few hours, the focus shifted towards the implementation of new modules rather than optimizing the speed of the new ones. However, it is one of the main milestones of the GaudiMM v2.0 roadmap.

# 5

## Python-based molecular modeling workflows

**R**ECRUITING different techniques to compose a multistep protocol is the very essence of multiscale molecular modeling. In addition to the scientific challenge itself, a technical barrier can arise: putting all the software to work together. To solve it, the researcher resorts to accumulated expertise in combining different file formats or, in some fortunate cases, even automating the conversions with scripting languages. Switching from program to program can be confusing and distracting, especially if those programs were not meant to be used together—a common situation in advanced molecular modeling. In those cases, some might prefer using a single cohesive user experience, normally in the form of a graphical interface (see chapter 1 for more details). The first part of this chapter will present the Tangram suite, a collection of graphical extensions for UCSF Chimera written in Python and Tk.

Of course, not all tasks benefit equally from a graphical interface. Some can be further improved by providing smart command-line tools. The remaining part of the present chapter will introduce complementary developments designed to improve the workflow and daily routine of computational chemists and molecular modelers alike. Both graphical and command-line developments (including GaudiMM) are presented in table 5.1.

### 5.1 IMPLEMENTATION OF A COMMON INTERACTIVE CANVAS: TANGRAM

Stemming from the former Midas program, UCSF Chimera is presented as *a highly extensible program for interactive visualization and analysis of molecular structures and related data, including density maps,*



**Table 5.1:** The full InsiliChem Molecular Modeling software suite.

GRAPHICAL INTERFACES	
TANGRAM <sup>150</sup>	A collection of more than 15 UCSF Chimera graphical extensions for integrative molecular modeling
ESIGEN WEBUI <sup>151</sup>	Automated generation of Supporting Information HTML5 drag & drop interface
COMMAND-LINE TOOLS	
GAUDI MM <sup>152</sup>	Modular multi-objective molecular optimization platform
PYCHIMERA <sup>153</sup>	Use UCSF Chimera modules in any Python 2.7 project
OMMPROTOCOL <sup>154</sup>	GPU-accelerated Molecular Dynamics protocols with OpenMM
GARLEEK <sup>155</sup>	Gaussian's ONIOM extended with Tinker MM force fields
ESIGEN <sup>151</sup>	Automated generation of Supporting Information documents from the command-line (batch mode)
EASYMECP <sup>156</sup>	A modern workflow for J. N. Harvey's MECP program <sup>157</sup>

*supramolecular assemblies, sequence alignments, docking results, trajectories, and conformational ensembles.*

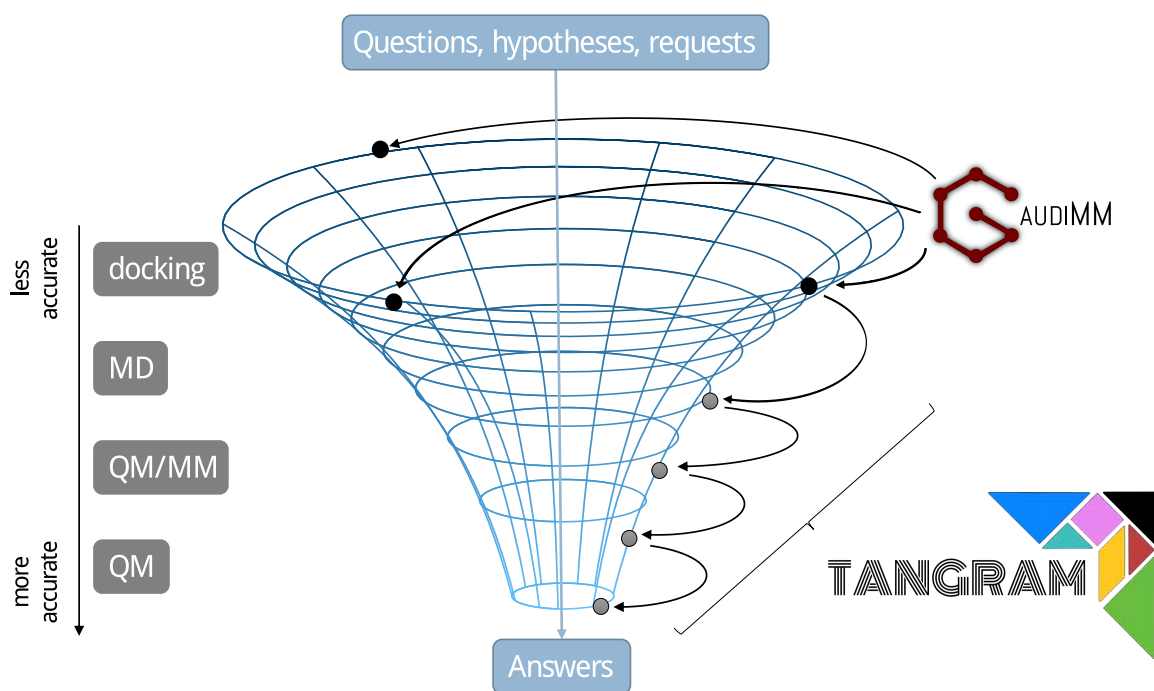
It consists of an interactive 3D visualizer built on top of C++ core with Python bindings, which is responsible of providing much of that promised *extensibility*. GaudiMM, commented in chapter 4, heavily uses UCSF Chimera as a backend library, but being a command-line tool, does not need any graphical interaction.

After developing GaudiView as the results viewer for GaudiMM (see section 4.4 and appendix D), the potential of having simple but powerful graphical interfaces for common modeling tasks became evident. In this section, we present Tangram, a set of extensions designed to add new pieces to the arsenal of molecular modeling tools already present in UCSF Chimera (see fig. 5.1).

**Table 5.2:** Tangram Suite: Technical datasheet.

TANGRAM SUITE	
<i>Description</i>	Graphical interfaces for UCSF Chimera
<i>Requirements</i>	UCSF Chimera, Python, PyChimera
<i>License</i>	MIT
<i>Download</i>	<a href="https://github.com/insilichem/tangram">github.com/insilichem/tangram</a>
<i>Documentation</i>	<a href="http://tangram-suite.readthedocs.io">tangram-suite.readthedocs.io</a>
<i>Citation</i>	(Submitted)

Tangram is composed of independent UCSF Chimera extensions that can play together through the interactive molecular canvas. This is, each extension can be used separately, but complex workflows can be implemented by using them sequentially. This distantly mimics the principles described in the UNIX philosophy: each extension should only do one thing and do it well.<sup>158</sup> As a result, some extensions in this package might look simple, but their true power arises when used together, like the pieces of a tangram puzzle. Hence the



**Figure 5.1:** Molecular modeling methods can be depicted in 3D funnel, where the width represents the sampling capacity and accuracy is depicted with depth. Less accurate methods with high sampling capacity would be depicted at the top, while most accurate methods would be at the bottom of the funnel. In this sense, GaudiMM can help access the entry of the funnel, while the Tangram interfaces will connect further steps down the accuracy scale.

name.

The different *tans* or components of Tangram can be of very different nature. Some provide interactive methods for setting up heavy calculations in external programs, like quantum mechanics in Gaussian or molecular dynamics in OpenMM. Others rely on the 3D viewer to depict properties of molecular structures as calculated previously in other software, turning UCSF Chimera in an even more versatile analysis tool. Some will wrap well-known executables meant for standalone usage and present the results in the UCSF Chimera canvas interactively, reducing the entry-barrier substantially. The following subsections will describe the Tangram components developed for multiscale modeling, listing the rationale and features implemented. Examples of integrative protocols using some of them will be provided in chapter 6. Additional extensions devoted to integrative analysis are collected in appendix D. The full list can be consulted in table 5.3.

**Table 5.3:** Full list of Tangram extensions.

CALCULATION SETUP FOR MULTISCALE MODELING	
QMSETUP	QM and QM/MM calculations setup, with Gaussian <sup>31</sup> and Garleek <sup>155</sup>
MMSETUP	Setup MD calculations with OpenMM <sup>11</sup> and OMMProtocol <sup>154</sup>
DUMMYMETAL	Parameterize metal ions for MM with the CaDAs <sup>159</sup> approach by placing dummy atoms in the coordination vertices
NORMALMODES	Perform interactive Normal Modes Analysis with ProDy <sup>142</sup>
REVINA	Resubmit failed AutoDock Vina <sup>13</sup> jobs without reconfiguring the GUI
INTERACTION ANALYSIS	
GAUDIVIEW	Lightweight visualization of results coming from GaudiMM <sup>152</sup> and GOLD <sup>146</sup>
NCIPLUGUI	Setup calculations for NCIPlot <sup>160</sup> and visualize them on-screen
PLIPGUI	Depict protein-ligand interactions, as calculated with PLIP <sup>161</sup>
STRUCTURAL ANALYSIS	
3D-SNFG	Intuitive visualization of saccharydic residues with the Symbol Nomenclature For Glycans <sup>162,163</sup>
ORBITRAJ	Display temporal volumetric data along a molecular trajectory
PoPMuSiCGUI	Depict potential site mutations in proteins as predicted by PoPMuSiC <sup>106</sup>
PROPKAGUI	Analyze the expected pKa values of protein residues with PropKa 3.1 <sup>117</sup>
SUBALIGN	Align potentially different molecules based on partial matches of substructures with RDKit <sup>164</sup>

## 5.1.1 MULTISCALE MODELING WITH TANGRAM

### 5.1.1.1 QMSETUP

QMSetup helps prepare Gaussian input files from UCSF Chimera for QM and ONIOM calculations (see sections 2.2.1 and 2.2.3). In GaussView,<sup>110</sup> setting up even the most common tasks would require going through scattered dialogs and tabs. QMSetup has been designed to provide a simpler workflow from a single dialog. Additionally, while UCSF Chimera is not as intuitive as GaussView for building small molecules, with QMSetup it shows several usability advantages, especially when macromolecules are present. Some highlights include:

- In UCSF Chimera, selection commands are hierarchical and can be extended from atoms to residues, chains and subunits with a single key stroke. This is really useful for selecting layers in ONIOM jobs or choosing which atoms should be frozen in an optimization, both options present in QMSetup.
- Some multiscale protocols involve setting up QM/MM jobs from different frames of a molecular dynamics trajectories. The different frames are just different coordinates sets of the same topology, so instead of creating separate input files one by one, a single one needs to be created. The remaining ones can be created automatically by updating the first one with the adequate coordinates. This is

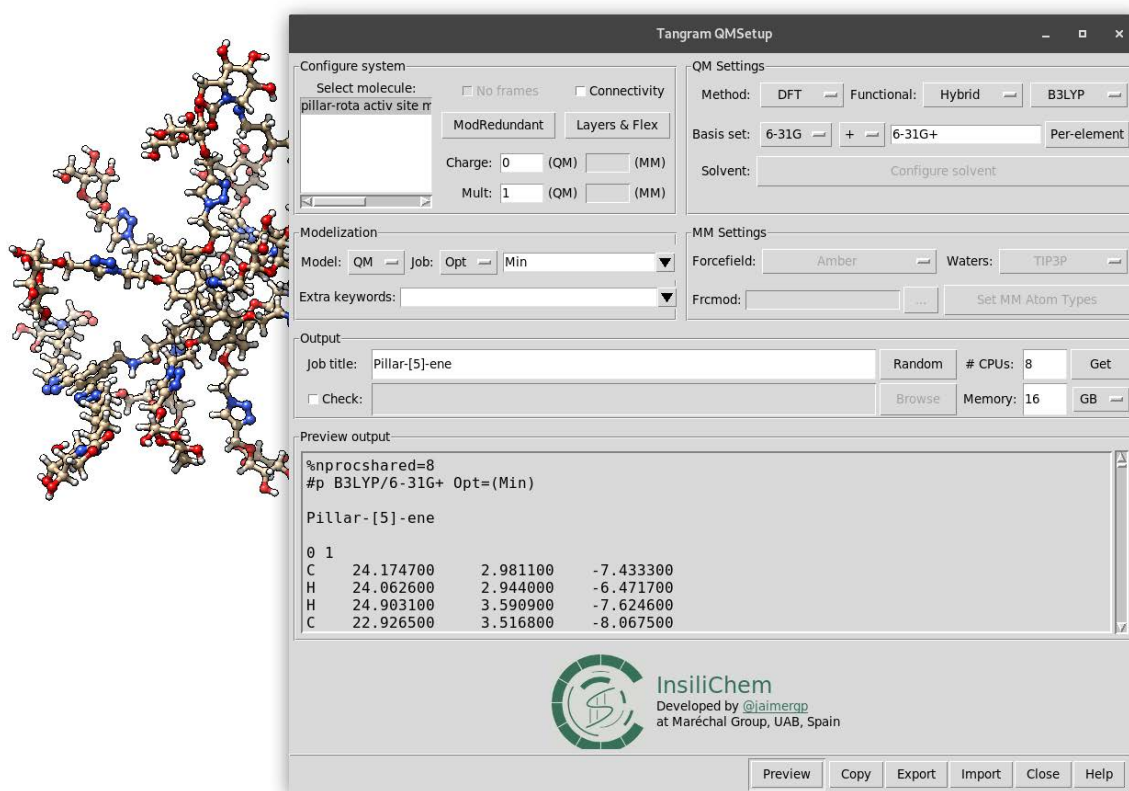


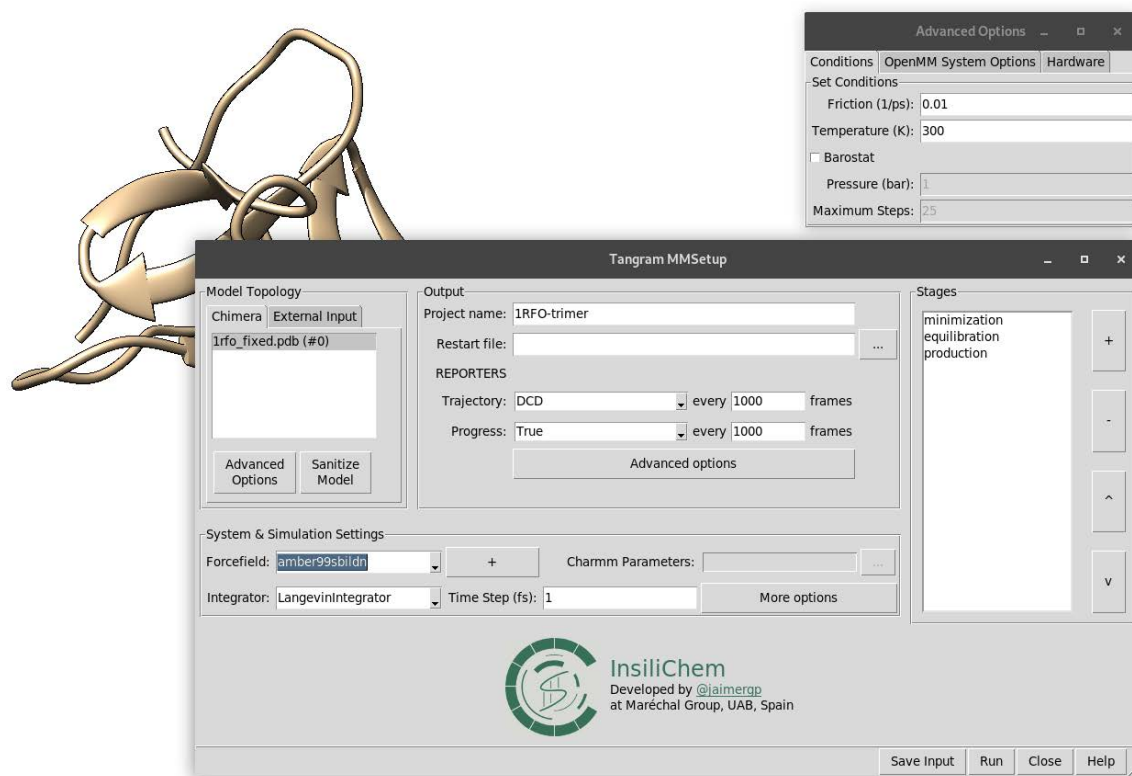
Figure 5.2: Tangram QMSetup interface allows to create QM and ONIOM jobs for Gaussian.

possible with QMSetup *Replica* option.

- In organometallics, exotic elements are used frequently. For these species, special basis sets are usually needed. Advanced users know about the Basis Set Exchange (BSE)<sup>165</sup> online platform and use it to locate the needed basis sets. QMSetup provides an offline interface to this dataset and handles the insertion in the input file automatically. This saves the hassle of copy-pasting the results and worrying about the adequate number of blank lines.

### 5.1.1.2 MMSETUP

MMSetup provides a graphical interface for setting up Molecular Dynamics simulations (see section 2.2.2) in UCSF Chimera using OpenMM behind the scenes. It recognizes opened molecules and offers different methods to prepare the final structure that will undergo the simulation (see fig. 5.3). For example, OpenMM's PDBFixer<sup>166</sup> can be used to add hydrogens and missing residues. Even missing loops can be modeled with this integration. Once the structure is prepared, the simulation protocol must be configured with its individual stages: minimization, equilibration and production by default. Then, the user can choose



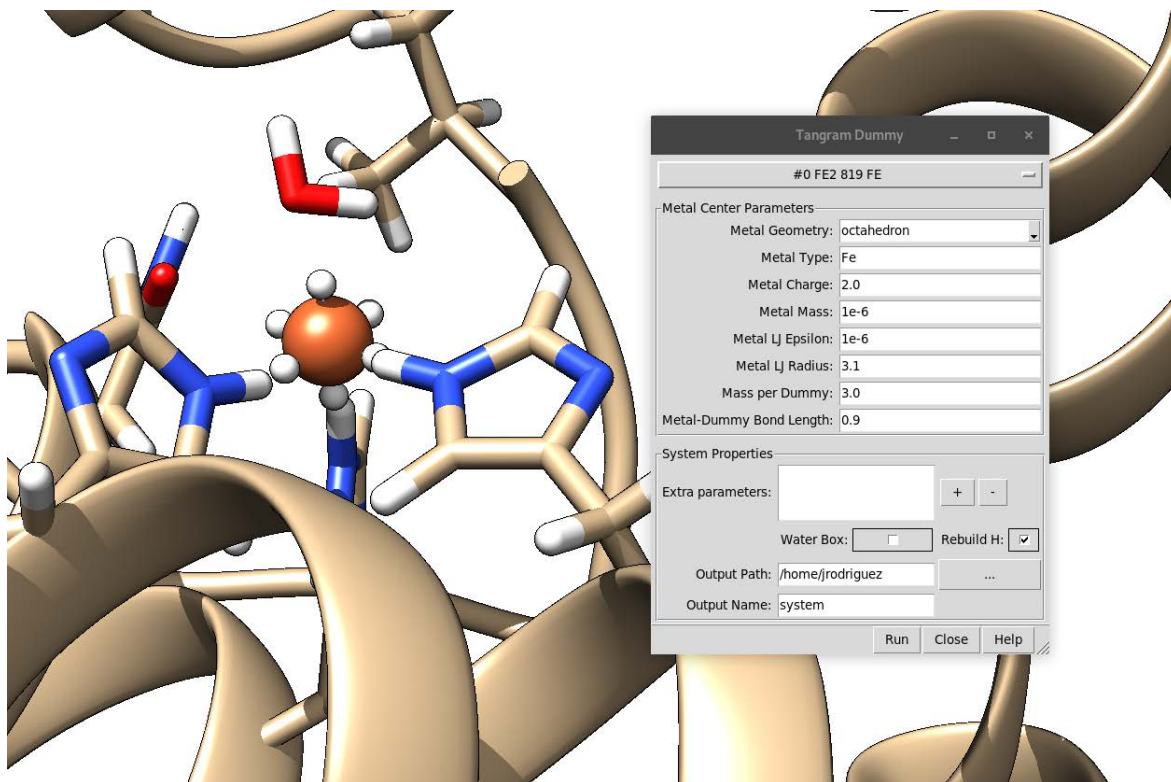
**Figure 5.3:** Tangram MMSetup can configure OpenMM calculations in UCSF Chimera, which can be run directly on-screen or in a separate job.

between running in situ and observing the evolution in real time (ideal for educational purposes) or creating an input file that can be run separately in cluster computers for speed.

### 5.1.1.3 DUMMYMETAL

In Molecular Mechanics, dealing with residues foreign to default force fields is one of the most difficult tasks. They require custom parameterization that in some cases can involve more complex calculations than the Molecular Dynamics simulation itself. When they are obtained, it is difficult to reuse them in other structures that also feature that residue because the connectivity or oxidation state might have changed. This is particularly painful if the new residue contains a metallic species.

For non-metallic organic compounds, Antechamber<sup>167</sup> routines are usually enough. However, for metal ions, the process is more intricate. Most methods proposed to generalize this process use high-level calculations in a reduced model, like Seminario's method derived MCPB.py routines,<sup>168</sup> but there are alternatives that skip those calculations by implementing virtual positions around the metal ion: the *Cationic Dummy*



**Figure 5.4:** With Tangram DummyMetal, metal centers can be easily modeled in MM force fields by following the CaDAs approach.

*Atoms* (CaDAs) approach.<sup>159</sup>

In the CaDAs approach, the metal ion is wrapped with positively-charged dummy atoms placed at the vertices of its expected coordination geometry. While the main idea is simple, building these systems accurately by hand is often disregarded for its difficulty. The DummyMetal extension can take a molecular structure, adapt the metal center with the CaDAs method (see fig. 5.4) and generate Amber-compatible PRMTOP, INPCRD and FRCMOD files. Since OpenMM can load Amber files seamlessly, the resulting files can be loaded in MMSetup to launch an MD simulation right away.

#### 5.1.1.4 NORMALMODES

Normal Modes Analysis methods are routinely used to study structural dynamics of molecules. Structural variability can be inferred from experimental data or computed MD simulations with principal component analysis (PCA), but it can be also computed with elastic network models (ENM) like the Gaussian or anisotropic network models (GNM and ANM, respectively).

This extension reuses part of the visualization functionality already implemented in UCSF Chimera exten-

sions previously developed by Muñoz Robles,<sup>169</sup> but ditches MMTK<sup>170</sup> and calculates ENMs with ProDy,<sup>142</sup> a more modern Python library specifically designed to compute protein dynamics. The resulting interface will list the calculated frequency vectors and animate the corresponding collective movements.

Since the interface itself is decoupled from the code that calls ProDy routines in the background, the collective vectors can be obtained from Gaussian QM `freq` jobs as well, if desired.

## 5.2 OPTIMIZING WORKFLOWS FROM THE COMMAND-LINE

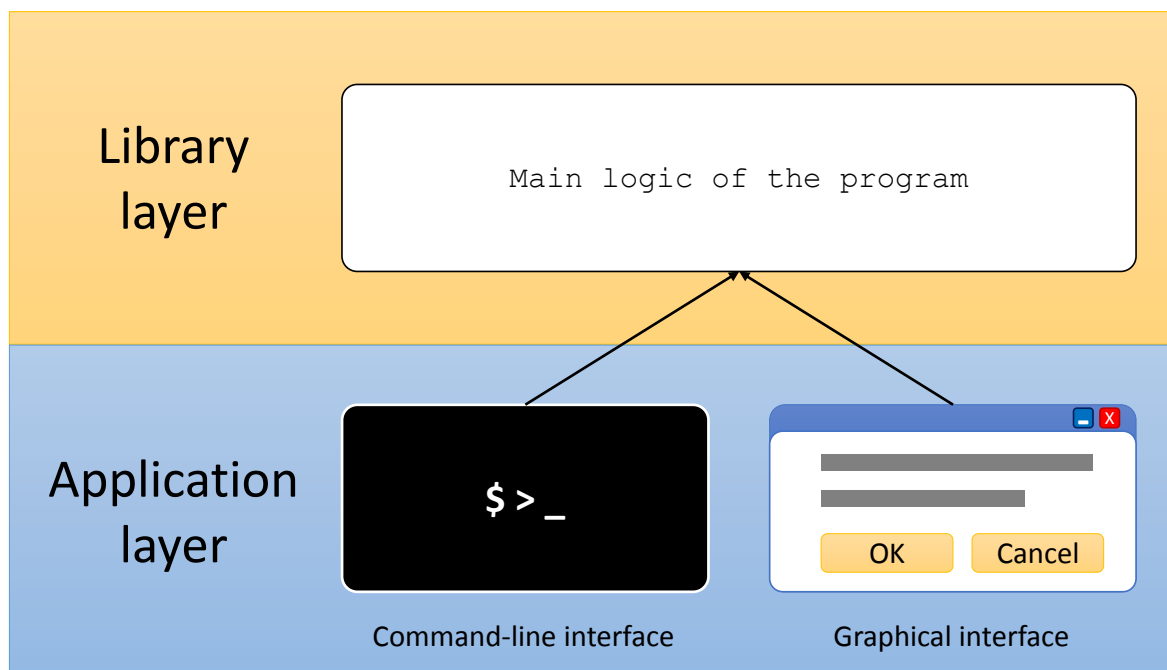
While a graphic interface can help with interactive tasks, there are other parts of the workflow of a molecular modeler that cannot benefit directly from a GUI. This type of software can be regarded as *backend* code that provides new, unsupervised calculation methods or, in other cases, an improved workflow that allows to perform the same calculation in an easier way.

Conceiving a project for command-line usage does not mean that a different interface can be built on top of that backend. In fact, MMSetup is just an interface around OMMProtocol, which in turn it's a user-friendly application around OpenMM. In QMSetup, the QM/MM support for additional force fields is provided by Garleek, which handles the Gaussian-Tinker programmatic interfacing. These two tools —OMMProtocol and Garleek— do not rely on UCSF Chimera and can be used as standalone command-line applications. However, since they are built with a decoupled architecture where the Python API is separate from the command-line interface (see fig. 5.5), they can support the aforementioned graphical interfaces.

In this section, the motivation, features and implementation of five different packages will be discussed: (1) PyChimera, (2) OMMProtocol, (3) Garleek, (4) ESigen, and (5) EasyMECP. Unlike GaudiMM, discussed in the previous chapter, they do not provide novel molecular modeling methods, but they do make them easier to use by automating repetitive tasks or abstracting away the technical details. This, ultimately, ends up saving the user some precious time.

### 5.2.1 PYCHIMERA

Most of the extensions listed in the previous section relies on libraries developed by 3<sup>rd</sup> parties that are not present in the UCSF Chimera Python distribution. Installing new packages inside UCSF Chimera is possible, but not very straight-forward. Additionally, some packages required by Tangram need long compilation times that would constitute a high entry barrier. To ease the process, the full Tangram suite can be installed



**Figure 5.5:** By structuring code in separate responsibility layers, new interfaces can be added easily without modifying the core logic.

with a single executable that is available in the central code repository.\*

This is possible thanks to the `conda` package manager,<sup>171</sup> which allows to redistribute compiled libraries and applications easily. However, since both `conda` and UCSF Chimera provide their own Python 2.7 distribution, they do not play well together. To solve this problem, the preloading code originally present in GaudiMM, which was needed to call the `gaudi` executable directly from the command-line, was extracted into a separate package and extended to connect UCSF Chimera Python distribution with any other one —be it the system-provided one, or virtual environments like `conda`'s or `pipenv`'s.

This new package was called PyChimera.<sup>153</sup> It does not try to alter the original UCSF Chimera installation; it only allows to load new packages from other locations outside the Chimera installation. For that reason, most Tangram extensions (those with external dependencies) will only work if a patched UCSF Chimera instance is loaded with the special `tangram` command.

PyChimera also includes some features particularly useful for developers, like exploring the UCSF Chimera codebase from augmented Python interpreters (IPython,<sup>172</sup> Jupyter Notebooks<sup>173</sup>) or providing auto-completions and help messages in advanced text editors (Sublime Text, Visual Studio Code). PyChimera was accepted for publication in *Bioinformatics*<sup>153</sup> and is the most popular package uploaded in the InsiliChem

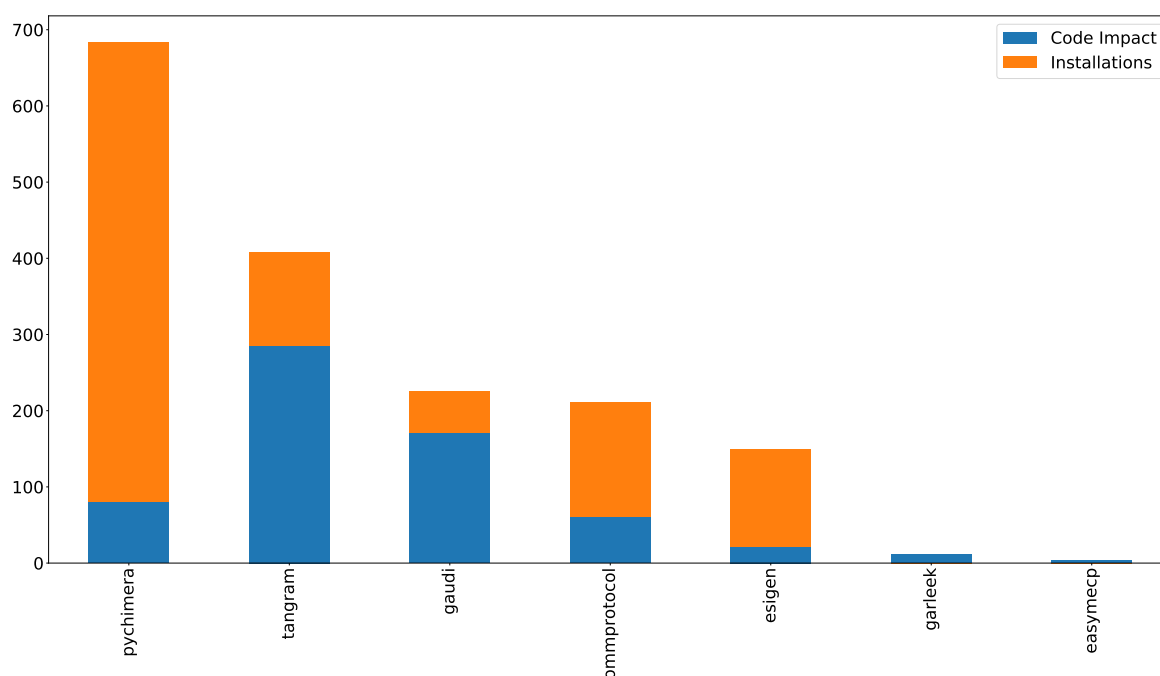
\*<https://github.com/insilichem/tangram/releases>



**Table 5.4:** PyChimera: Technical datasheet.

PYCHIMERA	
<i>Description</i>	Import UCSF Chimera modules in external Python projects
<i>Requirements</i>	Python, UCSF Chimera
<i>License</i>	LGPL
<i>Download</i>	<a href="https://github.com/insilichem/pychimera">github.com/insilichem/pychimera</a>
<i>Documentation</i>	<a href="https://pychimera.readthedocs.io">pychimera.readthedocs.io</a>
<i>Citation</i>	Bioinf. 2018, 34 (10), pp 1784–1785. DOI: 10.1093/bioinformatics/bty021 <sup>153</sup>

repositories (see fig. 5.6).



**Figure 5.6:** Popularity of InsiliChem packages developed during this Ph.D. Thesis, measured as the sum of *Code Impact* (registered visits and interactions in the source code repository), and *Installations* (downloads of compiled packages and requests from command-line installers, such as `conda` or `pip`). For ESigen, number of unique web app users is also listed in Installations. PyChimera is a clear highlight above the rest.

## 5.2.2 GPU-ACCELERATED MOLECULAR DYNAMICS, THE EASY WAY: OMMPROTOCOL

Molecular Mechanics (MM) and Molecular Dynamics (MD) (see section 2.2.2) are widely used in structural biology since they allow observing evolution of large biomolecules along time with affordable timescales and computational resources. This is particularly true after the popularization of General-Purpose Graphic Processing Units (GPGPUs) and their usage for calculations beyond graphics renderization. While long-

established MM suites like Amber,<sup>12</sup> Gromacs<sup>174</sup> or CHARMM<sup>70</sup> have been progressively implementing GPU acceleration in their code for some years now, a relatively recent project caught the community attention with its performance, flexibility, open-design and availability: the free OpenMM library.<sup>11</sup>

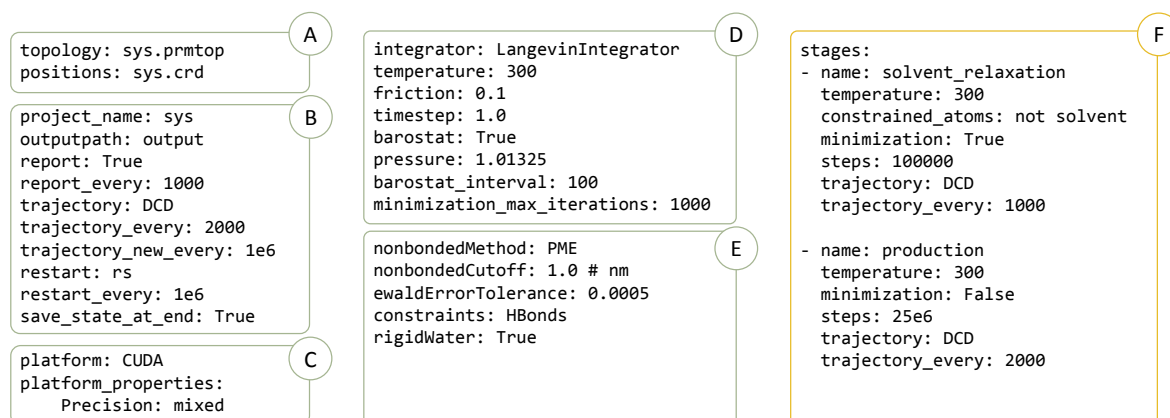
OpenMM presents a layered API designed for easy reutilization of its code in other projects. In fact, to use OpenMM, one is expected to write Python scripts to configure the simulation. These scripts are not harder to write than input files for other suites; they just happen to use that scripting language. That said, it could be easier. Users should not need to care about missing parenthesis, import statements or ending quotes. OMMProtocol was conceived to overcome this barrier by providing an easy to read and easy to write input file that abstracts away all the key underlying configuration steps with the concept of *protocol*: each input file contains all the stages involved in the simulation (like minimization, equilibration or production), avoiding the hassle of chained restarts.

**Table 5.5:** OMMProtocol: Technical datasheet.

OMMPROTOCOL	
<i>Description</i>	GPU-accelerated Molecular Dynamics simulations
<i>Requirements</i>	Python, OpenMM, ParmEd, MDTraj, openmoltools, pandas, matplotlib, jinja2
<i>License</i>	LGPL
<i>Download</i>	<a href="https://github.com/insilichem/ommprotocol">github.com/insilichem/ommprotocol</a>
<i>Documentation</i>	<a href="http://ommprotocol.readthedocs.io">ommprotocol.readthedocs.io</a>
<i>Citation</i>	(Submitted)

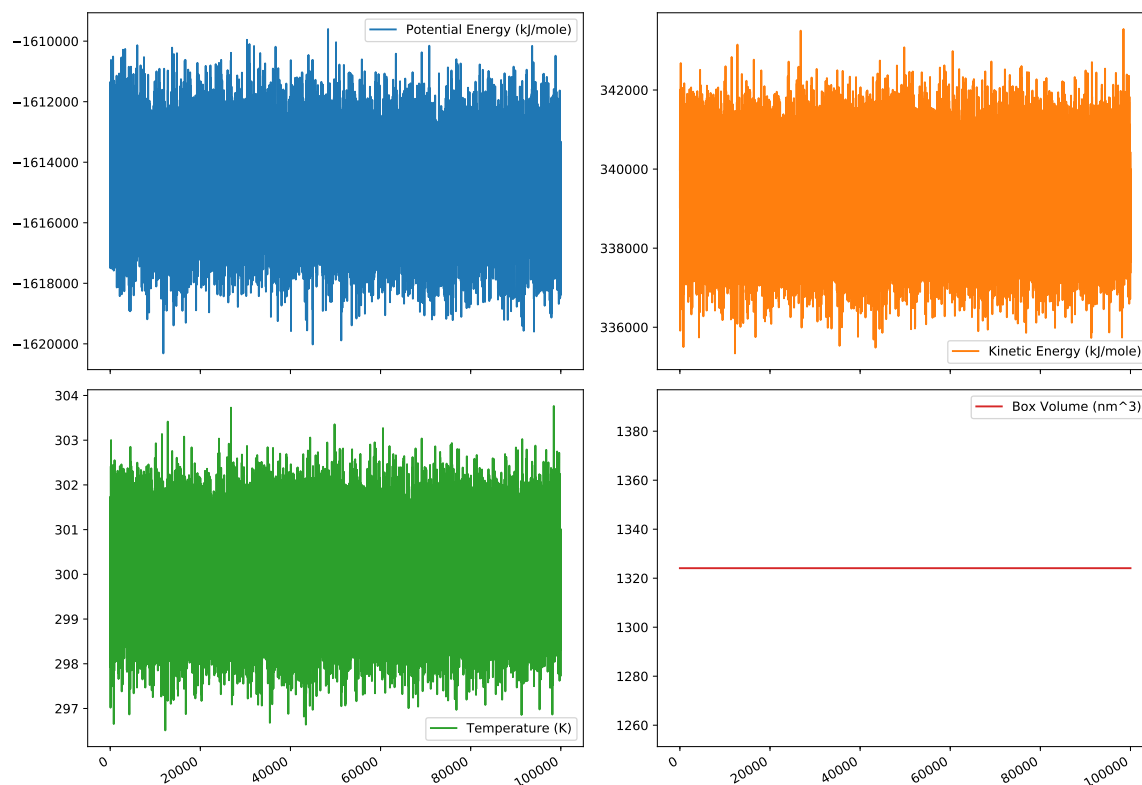
With OMMProtocol, setting up GPU-accelerated MD simulations can be as easy as loading a PDB structure, choosing one of the force fields provided and specifying the number of steps. Since default values have been chosen sensibly for compatibility with most popular cases, there is no need for added complication. That said, users are encouraged to review these parameters and adapt them to their specific needs by following the accompanying documentation and input file examples (see fig. 5.7). More details can be found in the submitted manuscript.<sup>154</sup>

OpenMM default input format compatibility is extended with even more file types by integrating other libraries together, like MDTraj,<sup>143</sup> ParmEd<sup>175</sup> or openmoltools.<sup>176</sup> This means that existing structure preparation workflows do not need to be disrupted: OMMProtocol will load Amber's PRMTOP, Charmm's PSF and Gromacs' TOP files seamlessly.



**Figure 5.7:** OMMProtocol files are formatted in YAML. Configuration keys can be specified in any order, but they have been grouped in this figure for convenience. Section A contains the structural data of the system to be simulated: the topology key is always required. Section B groups options related to file output. Section C controls the hardware to be used. Section D and E specify the conditions of the simulation. Finally, section E lists all the stages to be simulated in this protocol. Each entry, marked with a starting dash, can override any of the global options specified in sections B-E. Usually, only constraints, minimization, temperature and simulated steps will be modified here, since every other parameter is normally constant during the full protocol.

Finally, OMMProtocol is complemented by a second utility called `ommanalyze`, that drafts support for trajectory analysis protocols following the same spirit as OMMProtocol. This part of the project is only a stub so far, but it already provides automated, constant-memory RMSD analysis, and energy, temperature, and volume plots (see fig. 5.8).



**Figure 5.8:** OMMAnalyze can parse progress reports, written in the background as .log files, to plot the evolution of the potential and kinetic energies, the system temperature and the volume occupied. Since this data is readily available in the .log file, no expensive calculation of the magnitudes is needed. The opened dialog is interactive and can be used to zoom in the data, slice interesting parts and save high-resolution screenshots.

### 5.2.3 EXTENDED QM/MM FOR GAUSSIAN: GARLEEK

Gaussian<sup>31</sup> is one of the most popular QM packages and is still actively developed since its first release in the 70s. After almost 50 years, this package has been accumulating more and more features over time, and all of them are requested in the same counter-intuitive input file. While several alternatives exist with a comparable feature set and an easier workflow, even for free,<sup>147</sup> Gaussian is still king on many research groups.

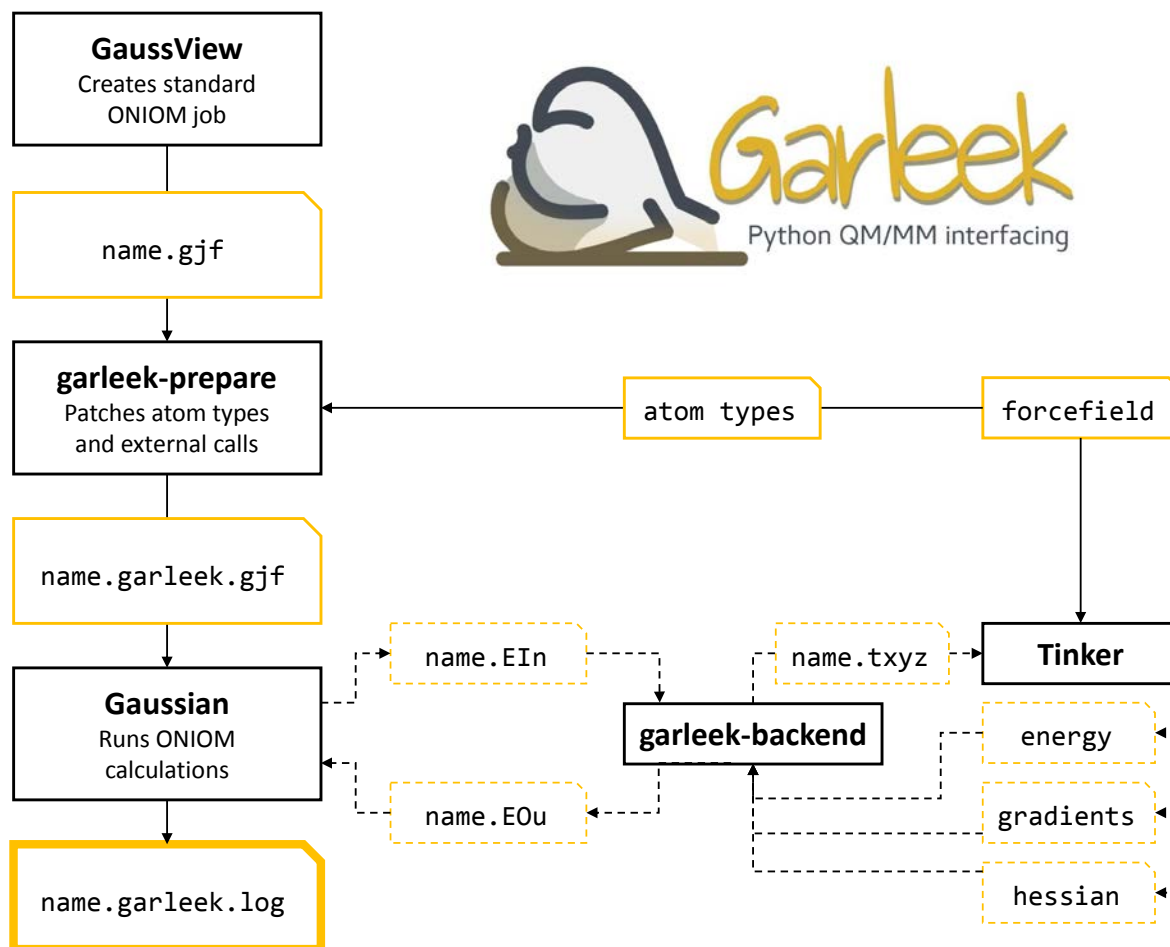
One of the features already included in Gaussian is the ONIOM method,<sup>16</sup> already described in chapter 2. This hybrid method splits a system in layers seeking to combine high-level calculations for specific regions that require very accurate modeling, with low-level theories that will deal with the remaining parts of the system. Most common applications usually use a QM method like DFT for the *high* layer and an MM method for the *low* layer. For this case, Gaussian provides a built-in MM engine suitable for calculations with only three force fields: Amber,<sup>177</sup> UFF<sup>178</sup> and Dreiding.<sup>179</sup> Fortunately, for those users that need other force fields, a communication protocol with 3<sup>rd</sup> party software is provided through the `external` keyword.

**Table 5.6:** Garleek: Technical datasheet.

GARLEEK	
<i>Description</i>	Additional MM support for Gaussian ONIOM jobs
<i>Requirements</i>	Gaussian, TINKER, Python, NumPy
<i>License</i>	MIT
<i>Download</i>	<a href="https://github.com/insilichem/garleek">github.com/insilichem/garleek</a>
<i>Documentation</i>	<a href="https://garleek.readthedocs.io">garleek.readthedocs.io</a>
<i>Citation</i>	Journal of Computational Chemistry, 2018. (In press)

Garleek is a Python package born after a collaboration with Dr. Ignacio Funes and Prof. Feliu Maseras. Garleek is designed to use this protocol to delegate the MM calculations to any other MM suite (see fig. 5.9). In the present state, it has full compatibility with all TINKER-provided force fields, like Amber99SB,<sup>180†</sup> CHARMM,<sup>181</sup> AMOEBA,<sup>182</sup> MMFF<sup>183</sup> or MM3.<sup>184</sup> Since the underlying architecture implemented in Garleek provides a straight set of guidelines, adding more MM packages is as easy as possible, thus avoiding reinventing the wheel. Garleek has been described in Journal of Computational Chemistry's Special Issue<sup>155</sup> dedicated to the memory of Prof. Dr. Keiji Morokuma.

<sup>†</sup>Gaussian does include the Amber forcefield, but an outdated version (94 and 98).



**Figure 5.9:** ONIOM workflow with Garleek. Black-border boxes describe programs, yellow-border boxes describe files. Dashed borders and lines describe temporary files created and removed on demand. The standard workflow involves creating a standard ONIOM input file (`name.gjf`) configured which is then patched to be Garleek-compatible with the `garleek-prepare` command, generating a copy (`name.garleek.gjf`). Gaussian runs this file and calls `garleek-backend` when necessary, which handles the communication with Tinker binaries for the MM calculations. The results are written to `name.garleek.log`.

#### 5.2.4 AUTOMATED ELECTRONIC SUPPORTING INFORMATION GENERATOR: ESIgen

Any scientific text must convey well-written ideas that make no room for ambiguous interpretation, but at the same time it should be easy to read. Handling such apparently conflicting ideas with ease is one of the reasons why good scientific communication is considered a hard task. One of the approaches to keeping the reader interested without losing correctness is to maintain a concise and direct style, which usually means taking all the technical details off the main text and supplying them in an accompanying document. Sometimes disregarded, Supporting Information (SI) and its electronic-only variant (ESI) are key to science reproducibility.

Computational chemistry, as all fields related to structural studies of molecules, tends to generate huge

amounts of data that should be inserted in the ESI: 3D depictions, coordinates, energies, and other characteristics of the structures involved in the molecular process under study. While most experienced users end up building scripts that dig throughout the output files searching for the relevant data, this is not the case for users without programming experience or time. In this section, we present ESIgen,<sup>151</sup> a Python project designed to automate the generation of technical reports suitable as ESI documents or internal communication between researchers. Initially conceived as a simple command-line script, it soon grew into a Python library that supports two interfaces simultaneously: (1) a web application and (2) a command-line executable.

**Table 5.7:** ESIgen: Technical datasheet.

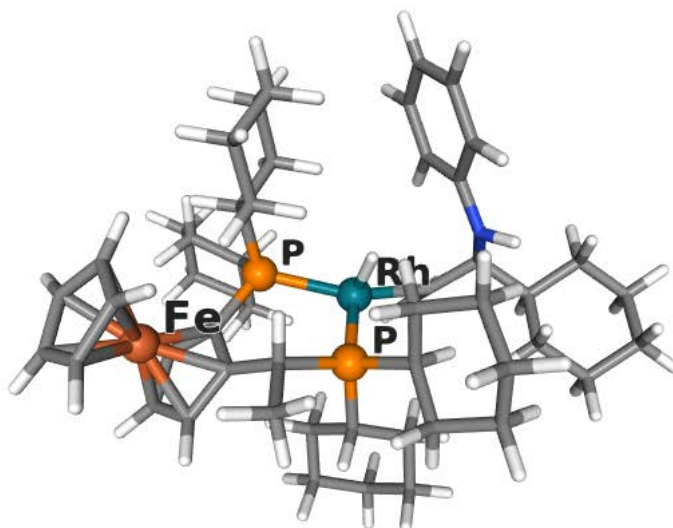
ESIGEN	
<i>Description</i>	Automated technical reports for computational chemistry calculations
<i>Requirements</i>	Python, cclib, jinja2, flask
<i>License</i>	LGPL
<i>Download</i>	<a href="https://github.com/insilichem/esigen">github.com/insilichem/esigen</a>
<i>Documentation</i>	<a href="https://esigen.readthedocs.io">esigen.readthedocs.io</a>
<i>Citation</i>	J. Chem. Inf. Model., 2018, 58 (3), pp 561–564. DOI: 10.1021/acs.jcim.7b00714 <sup>151</sup>

The drag-and-drop web application is meant for quick one-off usages where the user can inspect the structure interactively with the included 3D viewer<sup>185</sup> (see fig. 5.10). A public web app demo can be found at , which demonstrates how the web content can be seamlessly exported to DOI-citable repositories like Zenodo<sup>186</sup> or FigShare<sup>187</sup> or downloaded to disk in several formats (PDF documents, plain text, or even JSON<sup>‡</sup> programmatic objects). The command-line executable `esigen` allows to process several computational chemistry logfiles in batch with a single action. It will generate only plain-text files meant for further typesetting in text processors like Microsoft Word or LaTeX.

Both interfaces are based on the same usage principle: the user only needs to write a report template listing the requested fields as placeholders. The supplied template is then filled in with the requested data. Behind the scenes, ESIgen uses `cclib`<sup>188</sup> to parse the computational chemistry logfiles, which means that it is compatible with wide array of suites out of the box, like Gaussian,<sup>31</sup> NWChem<sup>147</sup> or ORCA.<sup>189</sup> Several examples are included within the package, covering most common cases.

<sup>‡</sup>JavaScript Object Notation

## Rh\_aleno1\_p\_Hext.ok



### Requested operations

```
opt freq=noraman gen scrf=(cpcm,solvent=dichloroethane) integral=ultrafinegrid m06 pseudo=read
```

### Relevant magnitudes

Datum	Value
Charge	1
Multiplicity	1
Stoichiometry	C <sub>51</sub> H <sub>77</sub> FeNP <sub>2</sub> Rh(1+)
Number of Basis Functions	1242
Electronic Energy (Eh)	-2960.09761781
Sum of electronic and zero-point Energies (Eh)	-2958.921229
Sum of electronic and thermal Energies (Eh)	-2958.864781

**Figure 5.10:** ESIgen can be used via a web interface and from the command line. When using the web interface (a demo is available at <http://esi.insilichem.com>), the user only needs to upload the quantum chemistry calculation output files to the server and select the data to report. After processing the file, an interactive HTML5 preview of the 3D structure can be displayed along the requested data so the user can manually find the best orientation for a static depiction.



### 5.2.5 EASY MECP CALCULATIONS

Minimum Energy Crossing Points (MECP) are defined as the consensus conformation of a molecular system that can feature low-energy minima in different spin states. A strategy to calculate them computationally was proposed by J. N. Harvey in 1998,<sup>157</sup> using Gaussian, GAMESS and custom Fortran routines orchestrated by shell scripts. The method and its related source code has been used widely across several research groups since then. However, setting up the MECP procedure involves recompiling the Fortran binary for each system, since its memory allocation requires the manual specification of the number of atoms. Convergence thresholds and other hardcoded values are scattered all over the source code, which does not allow easy access to these parameters. All these technical difficulties should not concern the user.

**Table 5.8:** EasyMECP: Technical datasheet.

EASYMECP	
<i>Description</i>	Simplified MECP calculations with Gaussian
<i>Requirements</i>	Python, gfortran, Gaussian
<i>License</i>	LGPL
<i>Download</i>	<a href="https://github.com/jaimergp/easymecp">github.com/jaimergp/easymecp</a>
<i>Documentation</i>	<a href="https://github.com/jaimergp/easymecp">github.com/jaimergp/easymecp</a>
<i>Citation</i>	(In preparation)

Developed during the collaboration with Dr. Funes and Prof. Dr. Maseras, EasyMECP is a self-contained Python script without added dependencies that takes care of all these steps automatically. The user only needs to provide a slightly modified Gaussian input file that specifies both spin states. Under the hood, EasyMECP still uses the original Fortran code, so convergence of results is guaranteed by design. Unit-tests are provided to support this claim. Additional conveniences have been implemented such as the generation of the optimization trajectory or the automated calculation of the often-needed thermochemistry of the converged MECP structure.

## 5.3 CONCLUSIONS & FURTHER WORK

The UNIX philosophy essentially restates that subdividing a problem in smaller chunks helps in solving that problem. Simple units responsible of single tasks are easier to understand and compose together into something bigger. This approach helped devise Tangram as a cohesive suite instead of a convoluted collection of dissimilar tools. By integrating tightly with the UCSF Chimera interactive canvas, all of them can work collaboratively.

However, UCSF Chimera starts to show its age and, while PyChimera allows to use it together with more modern tools, it is only a patch and cannot be considered a definite solution. A simpler integration in the vivid Python ecosystem would be desirable. This is being solved in the new, promising version of UCSF Chimera, UCSF ChimeraX,<sup>190</sup> which provides an online repository of one-click installable extensions called the *Toolshed*. ChimeraX is built on top of the same C++/Python premise, but uses the new Python 3 instead of Python 2 (which will stop receiving updates in 2020) and a different GUI library, Qt, which is easier to work with and its results are better-looking. ChimeraX is still very young and its feature set cannot be compared to the classic Chimera, but in the future this will be no longer the case. When that time comes, it will be possible to convert Tangram over to the new ChimeraX. Thanks to its modular design, the small pieces of this big puzzle could be migrated one by one, little by little, as soon as ChimeraX offers the needed features.

Developing new software is easier than changing how people work daily, though. Scientific community has proved to be very conservative about how they work, which is very paradoxical taking into account that science is all about progress. Some would argue that it is about progress, but in small steps. All the work involved in providing command-line utilities that work smarter and faster can be useless if nobody is going to use them.

For that reason, some of the tools presented in this chapter do not try to change things too much, too fast. For example, several alternative, easier-to-use MECP implementations can be found online,<sup>191–193</sup> but people still use Harvey's. EasyMECP is only a wrapper around the time-tested Harvey's original code. It does not try to change how it works; it just changes how you use it.

Another example can be found in the Supporting Information (SI) documents. In the future, SI will consist of digital repositories that are constantly updated and discussed, as some services like Zenodo<sup>186</sup> or Figshare<sup>187</sup> already provide. However, this complementary data is still being submitted as PDF documents, which are good for paper printing but not so much for data sharing. ESIgen does allow to generate PDF files from your data, but only after recommending the usage of data formats easier to share and reproduce. That will not prevent people from doing *what they have always done*, but sometime in the future, slowly but surely, we might get there.



# 6

## Benchmark & Application

**O**VER the previous chapters, several software developments have been presented. They try to fill different gaps in the multiscale modeling toolbox, be it the entry point to the funnel (GaudiMM) or to connect different stages down below (Tangram, OMMProtocol, Garleek...). In this chapter, several case studies where these programs have been used will be presented. Potential scenarios where they would be welcome will be also introduced as additional examples of applicability.

### 6.1 GAUDI MM AS A VERSATILE MOLECULAR MODELING TOOL

While GaudiMM's approach to molecular modeling can be daunting at first, once the key concepts are settled, configuring a calculation is straight-forward: it is a matter of which set genes and objectives to use. A particular combination of genes and objectives can be considered a *recipe* that can be adjusted for one study and reused in similar ones just by changing the involved structures. The following *recipes* will showcase common uses of GaudiMM.

#### 6.1.1 FROM STANDARD TO MORE EXOTIC DOCKINGS

Classic protein-ligand docking studies devote to finding the correct orientation and position of a small molecule (the ligand) within the cavity of a bigger one (normally, a protein). It usually cares about supramolecular recognition only, which means that analyzed interactions are mostly non-bonded. As a result, covalent bonds and coordination geometries are usually left out. However, a lot of systems do exhibit this type of recognition. In our group, we work with metallodrugs and artificial enzymes, two fields where these phenomena play an important role. As a result, we had big interest in considering these aspects in our docking

calculations.

In fact, GaudiMM was initially devised as an extensible protein-ligand tool, but later grew into a multi-purpose molecular modeling platform. The transition was smooth because the idea of docking can be further abstracted as simply *finding structural geometries compatible with certain requirements*, which is in turn a very specific type of restrained optimization problems. In this section, we will present how GaudiMM can perform several types of docking.

### 6.1.1.1 FLEXIBLE PROTEIN-LIGAND DOCKING: A BENCHMARK

GaudiMM capabilities for flexible protein-ligand docking were studied in its first publication,<sup>152</sup> where it was benchmarked against three different datasets\* using four genes and two objectives (see table 6.1 for more details).

All the entries in each dataset were analyzed with full torsion flexibility on the ligand, which could move and rotate within 12 Å of the crystallographic position. The results were analyzed considering the best RMSD of each calculation against the crystallographic reference structure. The correct binding pose was considered successfully reproduced if the RMSD was under than 3.0 Å. Despite not being the target usage of GaudiMM, the recipe reported success rates of up to 57.6% (see table 6.2), a value comparable to several works in the literature.<sup>194</sup> With more efforts directed at optimizing the exploration stage (especially the variation operators on torsion and orientation), the number of hits would highly increase and compete with other docking software.

**Table 6.1:** Recipe applied in the docking benchmark.

GENES	
Molecule	Load the protein
Molecule	Load the ligand
Torsion	Explore internal flexibility of the ligand
Search	Move the ligand within 12 Å of its starting point
OBJECTIVES	
Contacts	Minimize steric clashes
Contacts	Maximize hydrophobic interactions (target distance thresholds adapted)
LigScore	Minimize scoring function value

\*GOLD dataset (100 entries), ChemScore dataset (166 entries) and the CCDC Astex dataset (305 entries). Available at <https://www.ccdc.cam.ac.uk/support-and-resources/downloads>.

**Table 6.2:** Success rate<sup>†</sup> of a LigScore GaudiMM recipe against four benchmark datasets.

RMSD <sub>max</sub>	Benchmarked dataset		
	CCDC Astex <sup>a</sup>	GOLD <sup>b</sup>	ChemScore <sup>c</sup>
2.5 Å	41.64%	45.45%	45.45%
3.0 Å	51.80%	57.58%	51.52%

<sup>†</sup>Success was considered if at least one solution with LigScore score < 0 had an RMSD against the XRD structure within the given threshold. <sup>a</sup>305 entries. <sup>b</sup>100 entries. <sup>c</sup>166 entries.

### 6.1.1.2 COVALENTLY RESTRAINED DOCKING OF SEVERAL LIGANDS AT ONCE

As explained, sometimes there is an interest beyond non-bonded recognition, like when the ligand is covalently attached to some part of the protein. While there is no specific gene to implement a covalent bond, it can be mimicked through a `Search` gene configured to perform only rotation (translation can be disabled if `search radius = 0`). As this null `Search` gene will consider all possible rotations from that point, an `Angle` objective between the involved atoms in the covalent bond is recommended so that the resulting rotation matches the expected geometry of the new bond.<sup>†</sup> If more covalent interactions are needed, those can be modeled with a `Distance` objective set to bring the involved atoms within their covalent range (automatically calculated with the `covalent` keyword).

Additionally, taking advantage of the fact that genes and objectives can be instantiated multiple times,<sup>‡</sup> two or more `Molecule` genes can be set to open one ligand each. The compounds will be loaded simultaneously and they will compete to find their place in the protein(s). For this strategy to work, any related genes that are acting on the ligands (i.e. `Search` or `Torsion`) must be replicated accordingly.

This strategy allows competitive multi-ligand docking, something which is directly not possible in most docking software suites. It is true that it can be mimicked by performing sequential studies, in which the first ligand is docked separately and then the resulting solutions (protein *plus* first ligand) are fed as the host structure of the second ligand. However, in that context they would not be competing for the protein space simultaneously, per se: one of them has *priority* access. The procedure should be repeated to consider all possible orderings in order to be fair. With GaudiMM, this is not necessary since they are competing during the whole simulation.

<sup>†</sup>For example, for a carbon atom: 109.5° for  $sp^3$ , 120° for  $sp^2$ , 180° for  $sp^1$

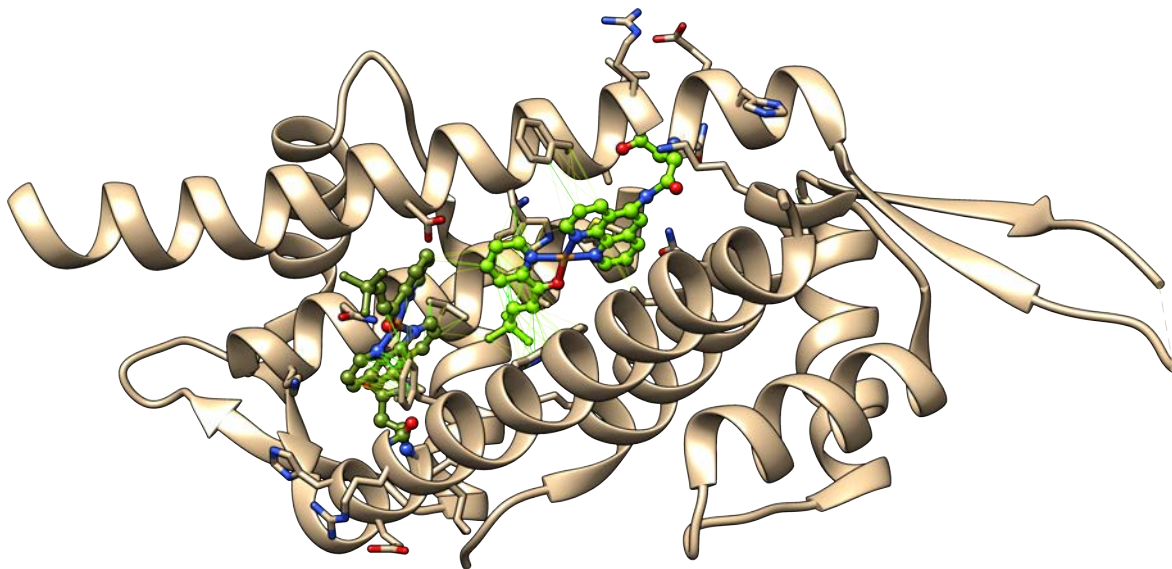
<sup>‡</sup>For example, in the previous sections, protein and ligand molecules were loaded with separate instances of the `Molecule` gene

Both concepts —covalent bond emulation and competitive docking— were tested during the design of an artificial metalloenzyme featuring a copper-containing phenanthroline cofactor covalently attached at the position M89C of the dimeric *Lactococcal multidrug resistance Regulator* (LmrR) protein. While the final structures in that work (manuscript under preparation) propose a single cofactor, having two ligands simultaneously attached to both monomers was also considered at the beginning of the study. To assess that possibility, a GaudiMM calculation was set up to see if two covalently attached ligands can fit within the dimeric interface (see table 6.3 for details).

**Table 6.3:** Recipe applied for the LmrR competitive docking calculations.

GENES	
Molecule	Load the LmrR protein (dimer)
Molecule	Load a copy of the Cu-Phn cofactor (ligand) and anchor it into position 89 of monomer A
Molecule	Load a copy of the Cu-Phn cofactor (ligand) and anchor it into position 89 of monomer B
Torsion	Explore internal flexibility of the ligand A
Torsion	Explore internal flexibility of the ligand B
Search	Allow free rotation of ligand A from its anchor, but no translation ( <i>radius</i> = 0)
Search	Allow free rotation of ligand B from its anchor, but no translation ( <i>radius</i> = 0)
OBJECTIVES	
Contacts	Minimize steric clashes
Contacts	Maximize hydrophobic interactions (target distance thresholds adapted)
Angle	Force an angle of 109.5° in anchor point of ligand A
Angle	Force an angle of 109.5° in anchor point of ligand B
DSX	Maximize docking scoring function to select stabilizing interactions
Solvation	Minimize solvent-accessible surface area (SASA) so the ligands are forced to interact within the protein rift instead of avoiding clashes in the exterior area

Under this scheme, several satisfactory solutions were obtained (see fig. 6.1) and submitted to a MD simulation to test their stability. Unfortunately, the trajectories did not report any long-lasting interaction between the ligands inside the protein and this alternative model was discarded. It must be highlighted that no experimental information was available besides the protein structure. With GaudiMM, new models could be obtained which, at least, suggested an idea on how these systems could be.



**Figure 6.1:** One of the candidate structures proposed by GaudiMM. The two Cu-containing phenanthroline cofactors (in green) were covalently attached to position 89 in both monomers of the dimeric LmrR protein. While the results were promising, the subsequent MD simulation proved that that particular conformation did not feature long-lasting interactions.

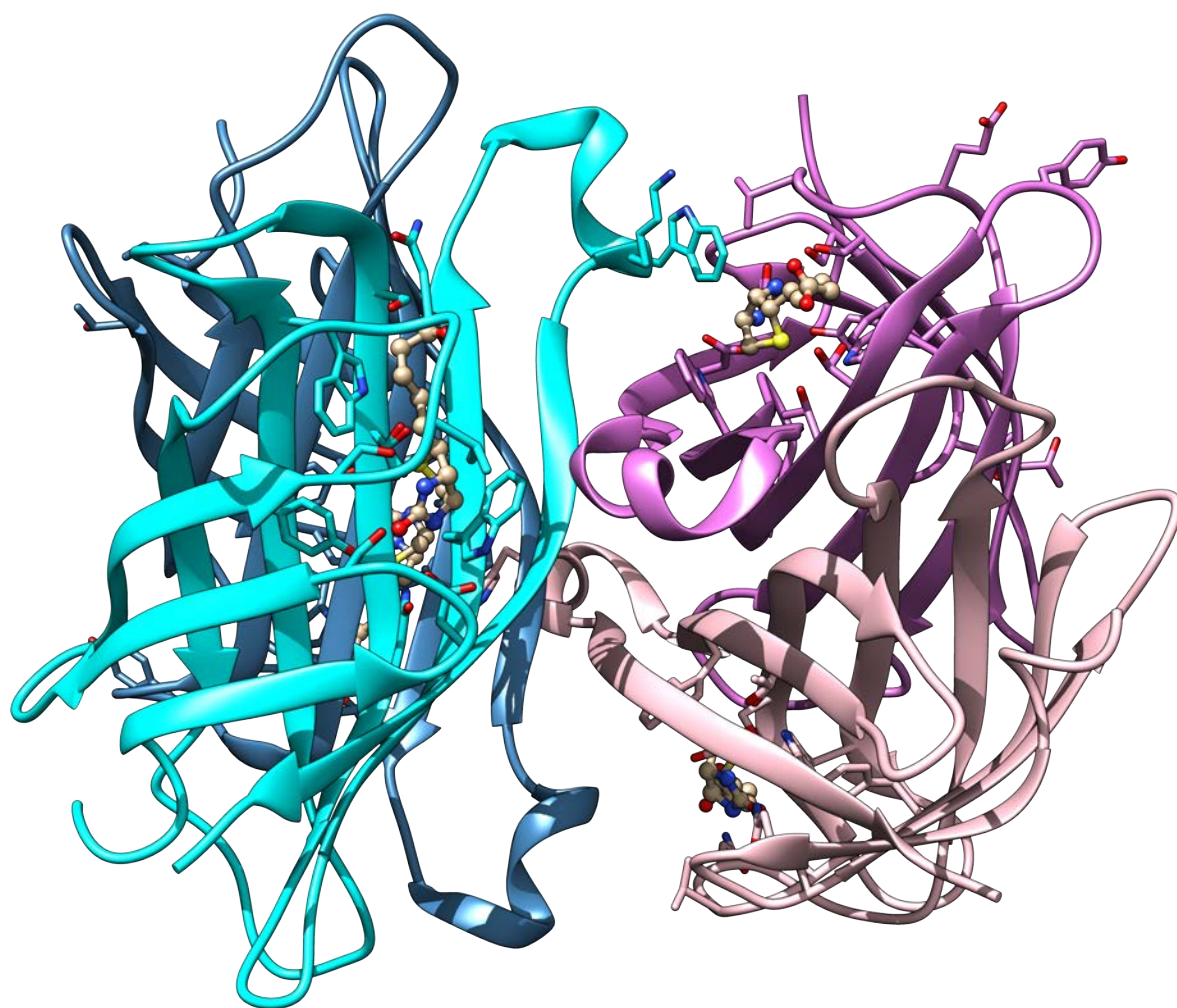
### 6.1.1.3 DYNAMIC DOCKING & LINKER-LENGTH OPTIMIZATION

The LmrR system is part of one of the most successful strategies to bring new stereoselective reactivities into existing enzymes: introducing cofactors that can anchor to the host and expose the catalytic region in a particular way. The anchoring can take place via a covalent bond (like in the previous case), but it can also be established through non-bonded interactions. In the latter case, biotin-derived cofactors have been particularly popular due to its high affinity to avidin and its bacterial counterpart, streptavidin, much easier to produce.

Streptavidin is composed of four monomers arranged as a dimer of dimers, each able to host a biotin molecule (see fig. 6.2). One possible strategy to fix a catalytic cofactor within the protein is to build a dibiotin derivative that could anchor to both binding sites simultaneously, holding the catalytic cofactor in the dimeric interface. This way, only one side of the cofactor is exposed to the medium, bringing higher enantioselectivity.

Ward and collaborators were trying to build this hypothetical ligand, but simply bonding the copper cofactor to one biotin on each side would not result in a compound able to reach both sites: a linker or spacer was needed to connect the biotins to the cofactor. The question is: which is the optimal length so the resulting ligand reaches both sites? This is where GaudiMM could help.





**Figure 6.2:** Streptavidin is composed of a dimer of dimers. Each monomer is capable of hosting one biotin molecule. Given its high affinity for biotin, it is a popular system for artificial enzyme design.

As briefly described in chapter 4, the *Molecule* gene can also be configured to build new molecules by chaining fragments found in a given directory structure. The subdirectories are sorted alphabetically and one fragment is randomly picked from each. The chosen fragments are concatenated following the subdirectory order, using the atoms configured as connectors (first and last atom in the file, by default).

In principle, the dibiotin ligand could be constructed out of five fragments: *biotin A + linker A + cofactor + linker B + biotin B*. However, since biotin exhibits a high affinity for streptavidin, it can be assumed that it will stay in its crystallographic binding site. This allowed to simplify the calculations: instead of having a 5-fragment construction, the biotins were fixed into their crystallographic binding site and a 3-fragment *linker + cofactor + linker* dynamical ligand was used instead (see fig. 6.3). To assess if a candidate construct was long-enough to reach both sites, one of the linkers was anchored to one of the biotins following the null-sphere procedure described in section 6.1.1.2. Then, a distance minimization objective was applied to the

other linker to push it into the second binding site, where a second biotin was placed to accept the simulated covalent bond. All the bonds in the linkers were allowed to freely rotate with the Torsion gene; biotin and cofactor bonds were considered frozen (see table 6.4).

**Table 6.4:** Recipe applied for the Streptavidin-dibiotin system.

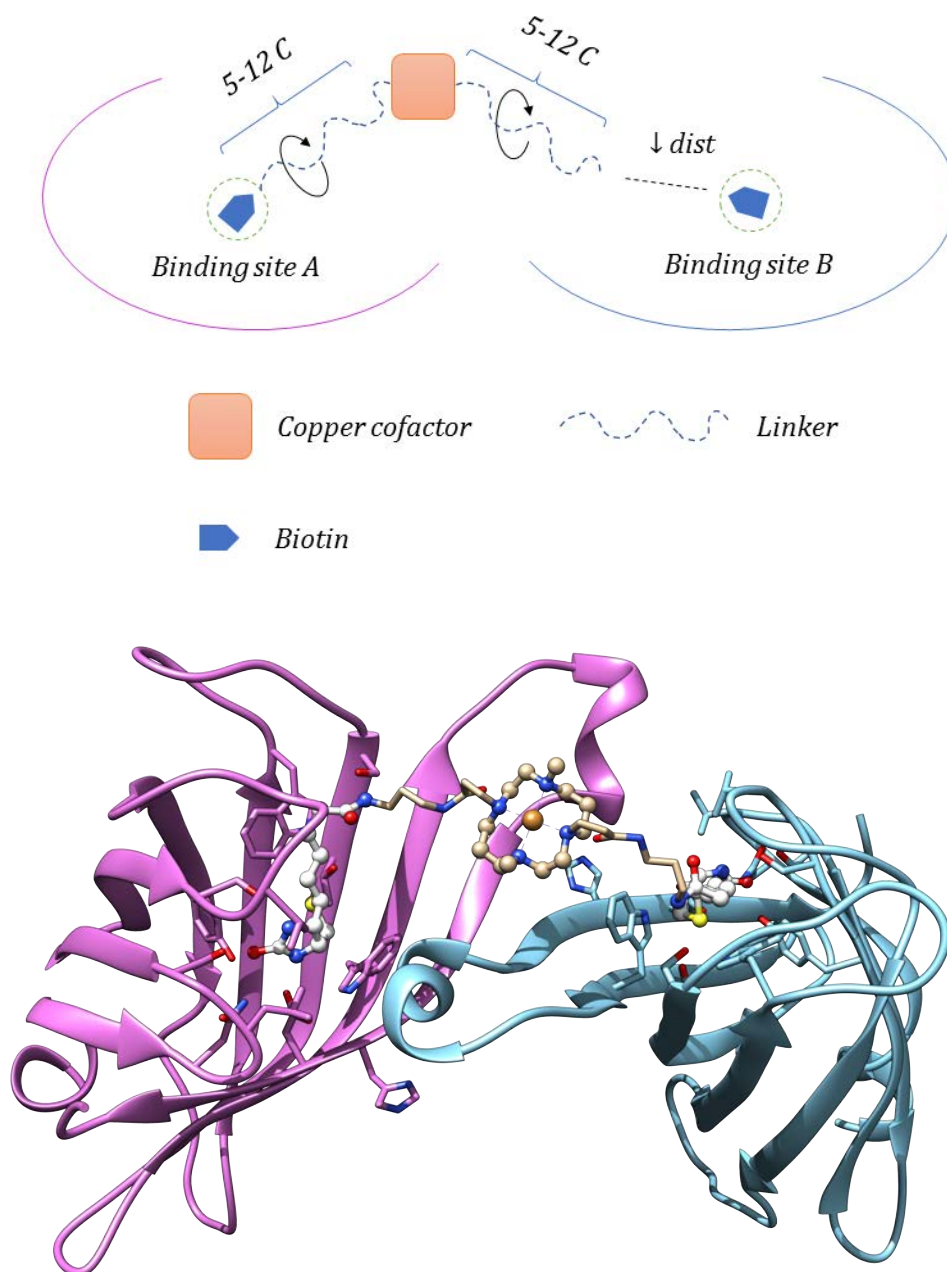
GENES	
Molecule	Load the streptavidin dimer, with biotin ligands already frozen in their binding sites
Molecule	Load a 3-fragment directory to build different versions of the <i>linker A-cofactor-linker B</i> construction. Linker library included linear alkanes ranging from pentane up to dodecane. The cofactor was DFT-minimized
Torsion	Explore the flexibility of the linkers (bonds of biotins and cofactor were considered frozen)
Search	Allow free rotation of the ligand from its anchor (end of biotin A), but no translation ( $radius = 0$ )
OBJECTIVES	
Contacts	Minimize steric clashes between protein and ligand
Angle	Force an angle of $109.5^\circ$ in anchor point of linker A with biotin A
LigScore	Maximize docking scoring function to select stabilizing interactions
Distance	Minimize the distance between the end of linker B and the end of the biotin B, so the two biotins are connected by the ligand

The results showed that a linker compatible with the length of a linear heptane would be enough to reach both binding sites simultaneously. This helped guide the synthesis of the dibiotin ligand (an amide group had to be introduced in the linker, but the suggested length was respected). The resulting ligand exhibits micromolar affinity with streptavidin (manuscript in preparation). Finally, our proposed model also contributed in the refinement process of the X-Ray structure. One of the most interesting parts is that the copper cofactor did not need any type of parameterization for the simulation to work. Simple descriptors like van der Waals overlap can be enough for complex modeling.

### 6.1.2 METAL IONS: ORGANIZATION & BINDING SITE PREDICTION

In two of the previous examples, a metal ion was present in a nonstandard residue. GaudiMM was able to deal with them because the recipes applied did not take any special considerations. All atoms were treated as different-radius spheres connected to other spheres. In some cases, this strategy can be successful, but in others special treatment might be necessary.

This section will present the applications of a novel strategy to deal with metal ions in molecular modeling. Instead of resorting to complex parameterization exercises (see appendix C for further details) like those



**Figure 6.3:** To figure out the optimum linker length for desired dibiotin ligand, a 3-fragment construction was prepared: linker+cofactor+linker. The first linker fragment was anchored to the biotin fixed in binding site A. The copper cofactor and the second linker were appended to its tail. By analyzing the torsions in both the first and second linkers, the end atom in the second linker can reach the biotin already fixed in binding site B. The solution (bottom) proposed a linker compatible with 7-carbon linear alkane.

expected in Molecular Mechanics, some properties of the metal ions can be described with geometry measurements.

#### 6.1.2.1 RESTRAINED CONFORMATIONAL ANALYSIS FOR ALZHEIMER'S B-AMYLOID PEPTIDE

Conformational analysis can be described as a range of techniques focused on taking an input molecular structure and generating coordinates sets (conformers) compatible with a set of criteria, normally energetic and geometric. This definition is broad enough to be used for other type of studies, such as the aforementioned docking variants, but is also a direct reflection on how GaudiMM impose a clear separation of concerns when it comes to exploration and evaluation. Once again, generating those new coordinate sets is a matter of choosing the right *genes*, and deciding if they are compatible with some criteria or not is up to the *objectives*.

To perform conformational analysis only one operation is needed: modify the coordinates set sensibly. This can be performed in a high-temperature molecular dynamics simulation, but parameters would be needed beforehand. For simple cases, GaudiMM's Torsion gene can be employed: all bonds lengths will be restrained to those in the input structure, but dihedral exploration will be performed on those bonds considered rotatable. Using the Contacts objective can help minimize the steric clashes and additional constraints like distances and angles can be imposed so the proposed solutions are compatible with a given geometry. The resulting calculation could be regarded as a restrained conformational analysis, very useful for finding initial structures of unparametrized small molecules you want to study with higher levels of theory, such as QM.

For some peptides, the same Torsion applied in small molecules can be applied to test the torsion angles of the CO-NH peptide bonds. This would result in the exploration of the backbone flexibility. This idea can be coupled with the Rotamers gene to assess the conformational variability of the sidechains. For the evaluation, full MM energy can be calculated with the Energy objective. Additional restraints are supported via the corresponding objectives: Angle, Distance, Surface, Volume...

In the GaudiMM manuscript, an unbound Alzheimer's  $\beta$ -amyloid structure was processed following this strategy in an attempt to reproduce experimental Zn-bounded NMR models (PDB ID 1ZE9<sup>195</sup>).

Starting with the unfolded peptide (PDB ID 1ZE7<sup>196</sup>), its backbone torsions were analyzed in hope of finding a combination of rotations that could be compatible with those in the Zn-bounded form. The only evaluators used were: (1) a Contacts minimization to rapidly discard structures with abundant steric clashes,

(2) an Energy minimization with the Amber 99 SBILDN force field for more accurate values, and (3) a Volume objective configured to match the average volume occupied by the 20 Zn-bound NMR structures reported in PDB ID 1ZE9 (15854 Å<sup>3</sup>), intrinsically showing a possible pre-organization of the isolated peptide. Even though the Zn ion was not explicitly considered, proposed structures were in agreement with the experimental ones, with backbone RMS deviations in the range of 3.5 Å (see fig. 6.4).

### 6.1.2.2 FINDING METAL BINDING SITES IN BIOMOLECULES

Metal-protein and metal-peptide interactions are not unusual at all: around 30% of the human genome encodes for metal-containing biomolecules.<sup>197</sup> As a result, having the ability to describe and predict these interactions becomes an interesting exercise in molecular modeling.

GaudiMM features an objective specifically designed for that purpose: the *Coordination* objective. The idea was initially described in Mujika et al.,<sup>198</sup> where a multi-objective docking procedure was used to locate possible octahedral coordination sites of an aluminum ion within pre-optimized structures of Alzheimer's  $\beta$ -peptides. The octahedral geometry could be described by using several distances, angles and dihedral objectives<sup>§</sup> set to match those in an ideal octahedron: coordination bonds of around 2 Å, 90° for the donor-metal-donor angle, and a dihedral of 109.5° for compatible sp<sup>3</sup> geometries (see table 6.5).

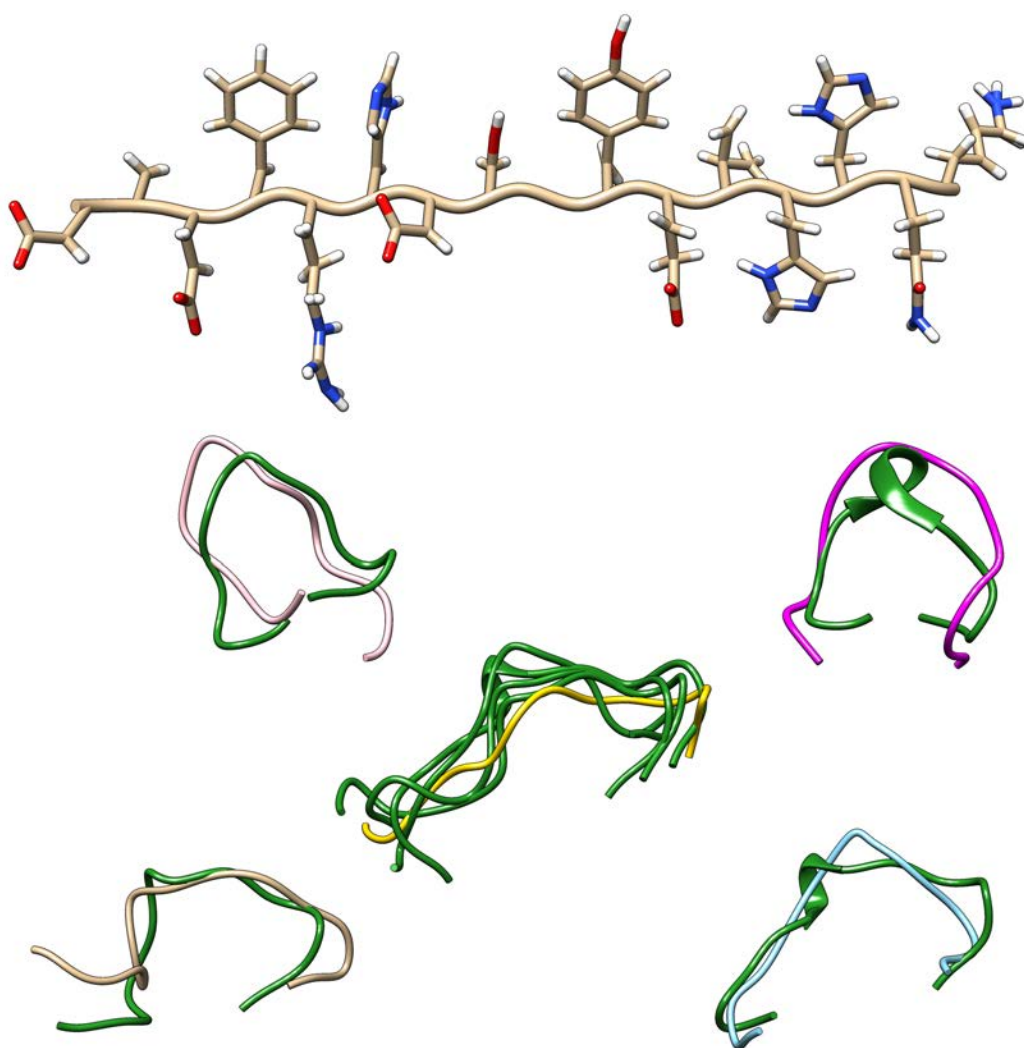
**Table 6.5:** Recipe applied for the Al(III)-amyloid complexes.

GENES	
Molecule	Load the preoptimized peptide, without the aluminium
Molecule	Load the bare aluminium ion
Search	Allow free translation of the aluminium ion within 5 Å
OBJECTIVES	
Contacts	Minimize steric clashes (stops the aluminium from getting too close)
Distance (x3)	Optimize the distance from the aluminium ion to the (three) closest oxygen atoms
Dihedral (x3)	Align the dihedral angles between the coordinating residues and the aluminium position

After validating the utility of this proof-of-concept, a first generalization of the method was implemented as a separate objective in GaudiMM, and tested in some illustrative cases for its publication (see section 6.1.2.3).

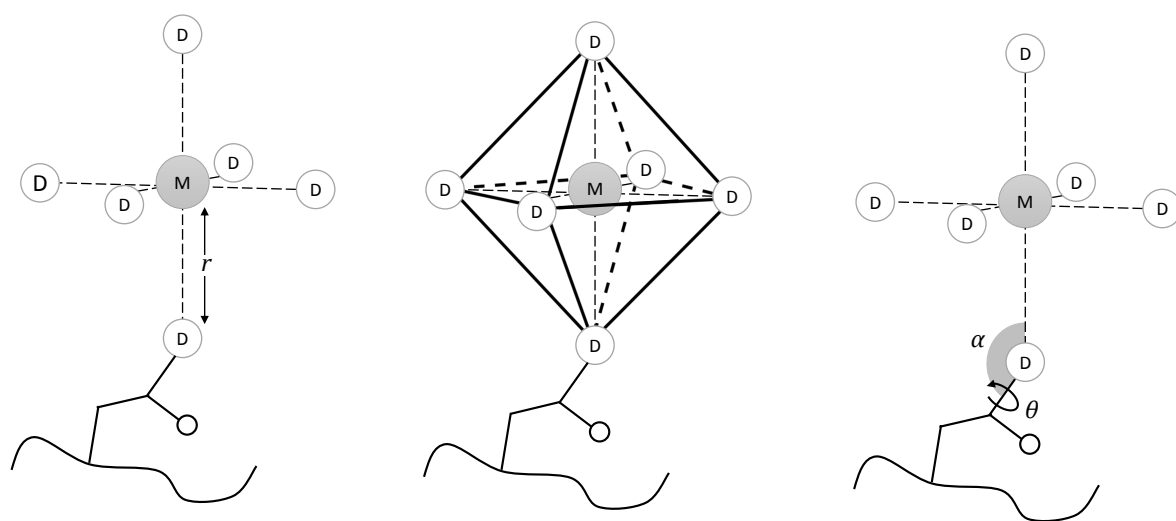
In short, the *Coordination* objective scans the surroundings of the selected metal ion for suitable donor atom types (like terminal sp<sup>3</sup> oxygen atoms in aspartic acid). If sufficient donors are found within 3.0 Å,

<sup>§</sup>This was done with a very early version of GaudiMM, when the notion of objective was in development. As a result, some of the modules here listed do not have an exact correspondence with the current ones.



**Figure 6.4:** Unfolded peptides can adopt feasible folded structures under a given volume only by exploring backbone torsion angles and force field energy minimization objectives. Low-energy solutions were further aligned to their best NMR conformation matches (in green).

their positions with respect to the metal center are compared to those of the vertices of the corresponding ideal polyhedron. This comparison is performed with a point-set registration method called Coherent Point Drift,<sup>199</sup> which admits missing points. This means that geometries with vacant vertices can be considered seamlessly. If the comparison is successful, a RMSD value is returned and two additional checks are calculated: (1) the directionality of the hypothetical coordination bond is assessed through the absolute difference of the sines of the angles and dihedrals of the involved atoms against the ideal ones, and (2) the ideal distance is compared to the measured distance and the absolute difference is. All these terms are summed linearly, which should result in a value of zero for perfect geometries. By minimizing this function over a few generations, coordination geometries can be identified.



**Figure 6.5:** In the Coordination objective, a metal ion  $M$  queries its surroundings looking for potential coordinating atoms (donors,  $D$ ). If the number of donors is enough, a Coherent Point Drift registration is performed to match the ideal polyhedron and the directionalities of the bonds are checked.

A posterior refinement of this objective has been extensively reviewed recently,<sup>148</sup> where we discuss different modifications of the initial score function. The most promising is benchmarked against a dataset of 106 high-quality X-ray metal-containing proteins representing diverse metallic species with octahedron-derived geometries. The protocol considers a 20 Å radius for the search sphere and a rigid protein structure. With these updates, the initial success rate increased from an initial 86% to a final 100%. If flexibility of sidechains is considered with the Rotamer's gene, the success rate retains a value of 87.5 %, even with the added search dimensionality.

## 6.1.2.3 FOLDING METAL-BOUND SIDEROPHORES

Siderophores are key compounds in the metabolic access to iron species, involved in oxygen transport and other vital processes. Since  $\text{Fe}^{3+}$  has bad solubility in water, these are responsible for their chelation and intake. Enterobactin is the strongest siderophore known ( $K = 10^{52} \text{M}^{-1}$ ) and is primarily found in Gram-negative bacteria. An iron-free 3D structure can be found in PubChem,<sup>200</sup> which substantially differs from its metal-bound form found in bacterial proteins, like *E. coli*'s FepB.<sup>201</sup>

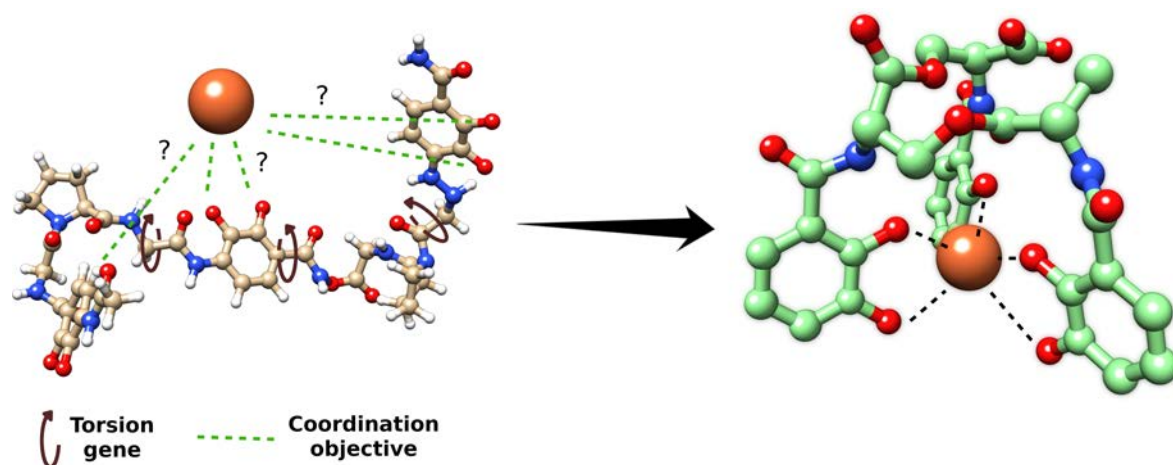
To assess the guiding capacities of the Coordination objective, an illustrative case was proposed in GaudiMM's original publication. The task was to fold the iron-free form into its metal-bound form. A Torsion gene was configured to explore rotatable bonds of the unfolded siderophore structure, and a Search gene was instructed to move the iron ion within a radius of 5 Å.

The main driver of the optimization was the Coordination objective, which was set up to find octahedral geometries by analyzing terminal oxygen atoms. This strategy successfully reproduced three of the four structures found in *E. coli*'s FepB. RMSD values were under 1.0 Å in all cases (see table 6.6 and fig. 6.6).

**Table 6.6:** Recipe applied for the enterobactin exercise.

GENES	
Molecule	Load the unfolded enterobactin <sup>200</sup>
Molecule	Load a bare iron ion
Torsion	Explore the flexibility of the rotatable bonds in enterobactin
Search	Move the iron within 2.5 Å
OBJECTIVES	
Contacts	Minimize steric clashes in the system
Coordination	Match an octahedral geometry by querying the terminal oxygen atoms of the enterobactin rings
Distance	Since UCSF Chimera does not support dihedral torsions in closed rings, the central ring of the enterobactin was deliberately opened so their torsion bonds could be explored. This distance keeps the ring functionally closed while that happens





**Figure 6.6:** GaudiMM can transform an unfolded apo-enterobactin siderophore into its metal-bound form as found in *E. coli* FepB by exploring its free-torsion bonds to find an octahedral coordination geometry around an iron ion.

## 6.2 CASE STUDY:

### MULTISCALE MODELING OF MULTIVALENT ENZYME INHIBITORS

GaudiMM was conceived to be the entry door to the multiscale funnel, while the tools presented in chapter 5 fill in other gaps down the funnel. In this section, a real example on how all these developments work together and allow to create integrative workflows is presented. It is a deep computational insight on the work described in *Pillar[5]arene glyco(mimetic)rotaxanes for the functional interrogation of multivalency responsive glycosidases*,<sup>202</sup> a collaboration with the IIQ-CSIC in Seville, Spain.

#### 6.2.1 INTRODUCTION TO MULTIVALENT ENZYME INHIBITION

Biological molecules are usually classified in four separate families: nucleic acids, lipids, saccharides and proteins. Far from being separate entities, they can be found together in many processes. A particularly interesting combination is when proteins and saccharides work together.

Glycoside hydrolases (or glycosidases) can specifically recognize glycosides and catalyze the hydrolysis of their intermonomer bonds in complex sugars. They are key in the degradation of natural polymers like starch (amylase) or cellulose (cellulase), pathogenesis, anti-bacterial activity (lysozyme) and normal cell function. Other examples are lectins, which exhibit high affinity for specific saccharidic residues through non-bonded interactions. They are involved in biofilm formation,<sup>203</sup> immune response<sup>204</sup> or even antineoplastic activities,<sup>205</sup> to name a few. Studying mechanisms of inhibition would allow to develop new antibacterial

techniques or avoid biofilm formation, for example.

Most common strategies for enzyme inhibition involve designing a mimetic ligand that can block the binding site of the substrate through a key-and-lock mechanism preventing the enzyme from performing any further action. For lectins and glycosidases, glycomimetic compounds like iminosugar-containing<sup>¶</sup> ligands can be employed. The first iminosugar characterized, 1-deoxynojirimycin (DNJ) was isolated from a natural source and showed to be an  $\alpha$ -glucosidase inhibitor with anti-diabetic and antiviral activities. Since then, more iminosugars have been described in the literature.

Recently, the general intuition around the classical lock-and-key inhibition mechanism was questioned in a study that described 2000-fold inhibition enhancement towards the Jack bean  $\beta$ -mannosidase (JbM) when twelve copies of DNJ were displayed around a  $C_{60}$  fullerene.<sup>206</sup> The strategy, termed Multivalent Enzyme Inhibition (MEI), was then observed in other systems. Knowing that JbM has a very accessible binding pocket and is multimeric in solution provides an understandable rationale behind this enhancement: a single  $C_{60}$  construct can block several sites at once. However, this idea cannot explain why monomeric enzymes with deep and narrow binding pockets are also inhibited by bulky multivalent conjugates. Additionally, it has been observed that putative sugars also inhibit their corresponding catalytic enzymes if they are multivalently displayed in a conjugate, defying the assumption of specificity and non-promiscuity glucosidases are thought to exhibit.

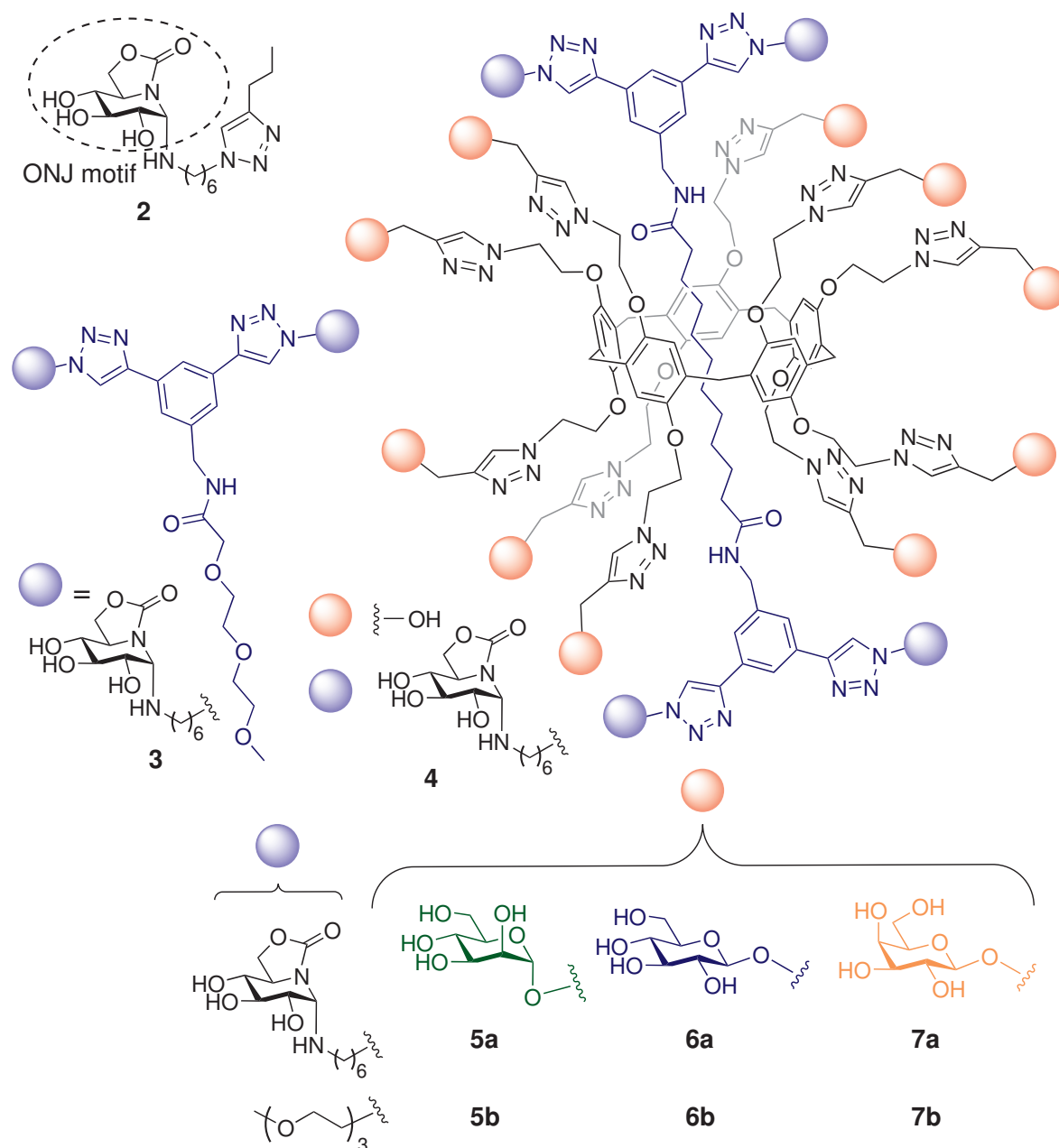
## 6.2.2 EXPERIMENTAL RESULTS

To shed light on these counterintuitive observations, García-Fernández et al. designed a series of two-component pillar[5]arene rotaxane conjugates that exhibited both glycomimetic and non-glycomimetic (putative) residues. The first component is a H-shaped central axel, which exhibits four  $sp^2$ -iminosugar-type<sup>[12]</sup> 6-oxa-5*N*,6*O*-oxomethylidenojirimycin (ONJ) residues, one on each on the stop caps. The second component is the pillar[5]arene, which displays ten moieties of glucose, mannose or galactose, depending on the variant, one on each of its ten rims (see fig. 6.7).

To test the contribution of the multivalent saccharides, two reduced models that did not include the pillar[5]ene were also considered: 1) a monovalent ONJ compound only bearing the hexyltriazolyl aglycone segment present in the rotaxane, and 2) a divalent ONJ compound, which emulates one of the stoppers halves of the central axel of the rotaxane.

---

<sup>¶</sup>Iminosugar moieties are standard saccharides whose oxygen atom in the ring has been replaced by a nitrogen atom.



**Figure 6.7:** Different inhibitors variants tested against the monomeric ScGH13 and dimeric TmGH1 proteins. In the computational studies, only divalent model (number 3) and rotaxane variant 5a were tested. Reproduced from Nierengarten, 2018.<sup>202</sup>

All these inhibitors and their controls were tested and measured on two glycosidases for which crystallographic data evidenced the existence of deep binding pockets: the monomeric GH13  $\alpha$ -glucosidase from *Saccharomyces cerevisiae* (*ScGH13*, yeast maltase) and the dimeric GH1  $\beta$ -glucosidase from *Thermotoga maritima* (*TmGH1*). The inhibition power was measured experimentally and reported in  $\mu\text{M}$  units (see table 6.7). Further details are out of the scope of this dissertation but can be found in the manuscript.<sup>202</sup>

**Table 6.7:**  $K_i$  (in  $\mathcal{M}$ ) for the different inhibitors tested against the monomeric ScGH13 and dimeric TmGH1 proteins. Reproduced from Nierengarten et al., 2018.<sup>202</sup>

$K_i^a$	ONJ		Pillar-[5]-arene			
	Monovalent	Divalent	Regular	D-manno	D-gluco	D-galacto
ScGH13 <sup>b</sup>	$4.3 \pm 0.5$	$2.5 \pm 0.3$	$2.3 \pm 0.1$	$33 \pm 2$	$16 \pm 1$	$13 \pm 1$
TmGH1 <sup>c</sup>	$109 \pm 5$	$3.0 \pm 0.2$	$7.1 \pm 0.3$	$6.3 \pm 0.3$	$16 \pm 2$	$27 \pm 2$

<sup>a</sup> Measured in  $\mu\mathcal{M}$ . <sup>b</sup> Monomeric. <sup>c</sup> Dimeric

For the monomeric ScGH13, results were not surprising: the monovalent ONJ is a strong inhibitor with  $K_i = 4.3 \pm 0.5\mu\mathcal{M}$ , and the divalent ONJ shows a 1.7-fold enhancement explainable by statistics alone: two iminosugar moieties are available in each molecule. The pillar[5]ene variant, featuring four iminosugars, should have increased that value again but stayed at  $2.3 \pm 0.1\mu\mathcal{M}$ , maybe due to steric impediments. The trend was not the same in the dimeric TmGH1, though. The monovalent ONJ model was a very weak inhibitor ( $K_i = 109 \pm 5\mu\mathcal{M}$ ), but the divalent variant showed a 36-fold enhancement ( $K_i = 3.0 \pm 0.2\mu\mathcal{M}$ ). The pillar[5]ene variants did not improve this  $K_i$  but did not cancel it either.

This surprising enhancement in the inhibition power of the divalent ONJ compound towards the dimeric TmGH1 could not be explained by stoichiometry alone. One hypothesis was to consider that the dimeric conformation observed in the X-Ray data was not the only one present in solution, leaving room to the idea that the divalent ONJ compound was able to reach the binding sites of two hypothetical monomers at the same time and force a blocked dimerization. However, for that to be possible the di-ONJ ligand must be long enough, something that was not entirely safe to assure. If a single di-ONJ ligand was not enough, could two di-ONJ molecules occupy a binding site each and then stabilize each other via the free ONJ end? What about the hydrophobic tail? Is the same interaction profile feasible for the pillar[5]arene variants? At this point, computational insights were requested in hopes of finding structural models that could help explain the experimental observations. The questions posed are summarized now:

- Can a single di-ONJ ligand reach both sites? If not, how long should it be?
- Can two di-ONJ ligands occupy both sites simultaneously? Do they interact?
- Does the pillar[5]ene compound fit in the dimer? Can it occupy both sites comfortably? If so, in which conformation?

### 6.2.3 COMPUTATIONAL APPROACHES TOWARDS AN EXPLANATION

A project like this, in which computational insights are needed to clarify experimental observations, is perfect to prove how GaudiMM allows to formulate (bio)chemical hypothesis as optimization problems solvable by a multi-objective genetic algorithm. Each question might need a slightly different resolution strategy, which will be described in the corresponding subsections, but the main protocol remains the same:

1. The problem is formulated as a GaudiMM *recipe* and run.
2. All the candidate structures are analyzed interactively with GaudiView and any needed Tangram extensions. Interaction profilers (see appendix D) are particularly useful at this stage.
3. The best candidates are checked for stability with long molecular dynamics trajectories (more than 100 ns). The protocol involves using explicit solvent and full-atom treatment using the GPU acceleration provided by OMMProtocol.

#### 6.2.3.1 CAN A SINGLE DIVALENT LIGAND REACH BOTH SITES?

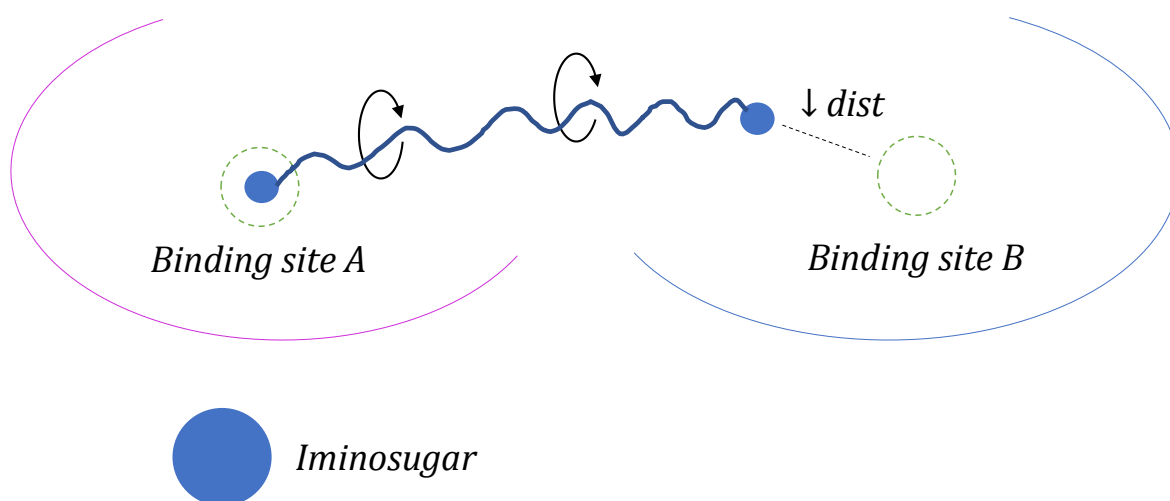
IF NOT, HOW LONG SHOULD IT BE?

The  $K_i$  of the divalent ligand and the rotaxane against TmGH1 cannot be explained by stoichiometry alone and it was suggested that both binding sites are reached simultaneously. However, it was not entirely clear if the di-ONJ ligand was long enough to reach them both. When the iminosugar residues are set to be as far as possible from each other, they can get 30 Å apart. In the crystallographic structure of TmGH1,<sup>207</sup> the crystalized ligands are 40 Å apart in a straight line. Additionally, it must be considered that the dimer interface is curved and a greater length must be covered in order to reach both sites in an energetically feasible manner. Alternatively, another dimerization structure could happen in solution and that could be analyzed with a combination of protein-protein docking and solvated molecular dynamics.

To assess the first possibility, a GaudiMM calculation was set up following this strategy: one of the iminosugars was fixed to the crystallographic site of one dimer, and the other iminosugar was instructed to reach the binding site in the opposite dimer by exploring the dihedral torsions of its rotatable bonds (see fig. 6.8). To discard steric impediments, unfavorable clashes were minimized. This can be achieved with the recipe detailed in table 6.8.

**Table 6.8:** Recipe used in the evaluation of a single di-ONJ ligand. The ligand was positioned in such a way that one of the terminal iminosugars matched the crystallographic structure of the ligand in the original 2WBG protein structure.

GENES	
Molecule	Load the protein model obtained after cleaning the PDB structure 2WBG (waters and ligands removed)
Molecule	Load the divalent ligand using a Mol2 file obtained with ChemCraft
Torsion	Explore the free rotations of the ligand molecule
OBJECTIVES	
Contacts	Minimize steric clashes
Distance	Bring the free iminosugar end of the ligand as close as possible to the center of mass of the original crystallographic ligand in the X-ray structure <sup>207</sup>

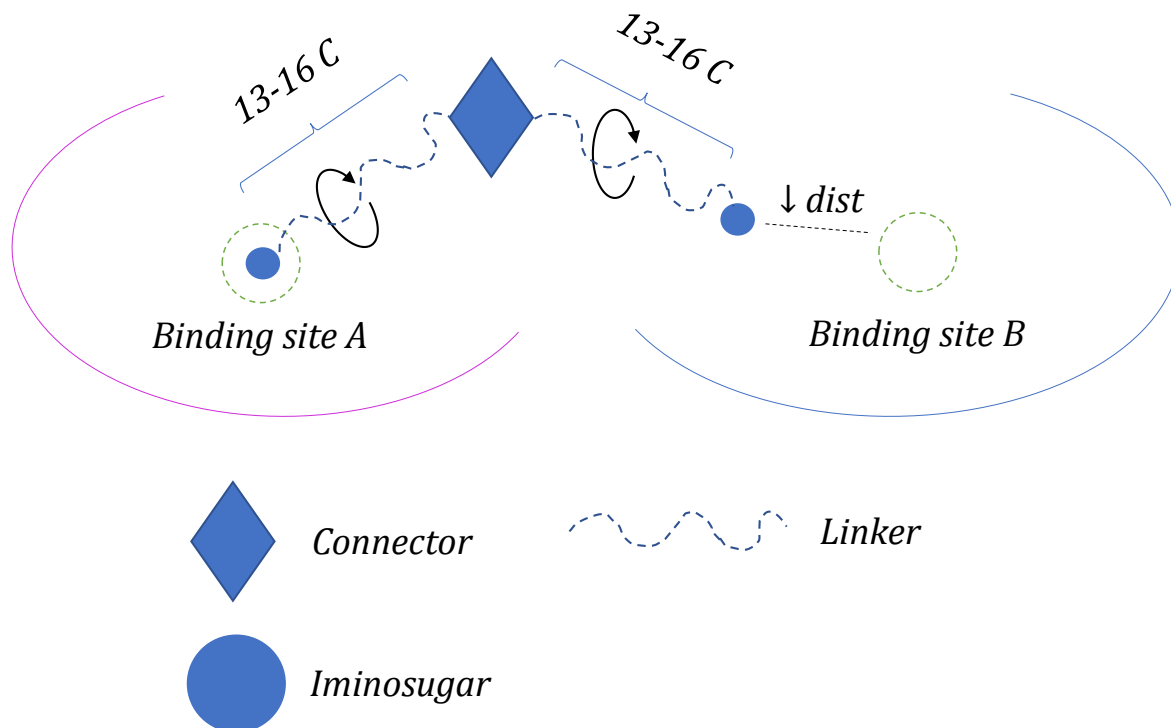


**Figure 6.8:** To assess if a single divalent ONJ ligand can reach both binding sites of TmGH1 simultaneously, one iminosugar end was fixed in the binding site of one monomer (left) and the other end was instructed to minimize its distance to the other binding site (right) by exploring the free-torsion bonds rotation.

The results of this preliminary calculation confirmed that the synthesized divalent ligand was not long enough to reach the both sites. If one site was forced to be occupied by one of the iminosugars, the other end will stay at a distance of 12 Å, even considering severe steric impediments. Seeing that a single divalent ligand cannot occupy both sites simultaneously, an inhibition mechanism by sliding is proposed: a single divalent molecule must be able to switch from one site to the other, taking advantage of having an iminosugar on both ends.

This observation makes the next question obvious: how long should it be then? The ligand is, simply put, two iminosugars connected by two chains of 10 atoms each. If we consider longer linkers, it might be possible to reach both. To assess that possibility, a library of linkers ranging from 13 to 16-carbon linear alkanes was

constructed and the same protocol was applied, but changing the `Molecule` gene to consider a dynamic construction of molecules as explained in section 6.1.1.3 (see fig. 6.9 and table 6.9). The fragments can be chained as *iminosugar + linker + hydrophobic connector + linker + iminosugar*.



**Figure 6.9:** To guess the optimum length of the divalent linker, GaudiMM was instructed to build ligands with linkers ranging from 13 to 16 carbon atoms, explore their free-torsion bonds rotations and minimize the distance to binding site B.

The evaluation part is the same: the non-frozen iminosugar will be forced to reach the other dimer, but only long enough ligands will be able to do so. The results show that linkers longer than 13 carbons are able to reach both sites comfortably without the need to slide from one to the other.

### 6.2.3.2 CAN TWO DIVALENT LIGANDS OCCUPY BOTH SITES SIMULTANEOUSLY?

#### DO THEY INTERACT?

A second hypothesis would consist of considering that two di-ONJ ligands can occupy both sites of the same dimer simultaneously. If that is the case, they could even stabilize each other by interacting at the dimer interface via their free iminosugar moieties (which can form hydrogen bonds through their hydroxyl groups) or the coupling of their hydrophobic tails.

To assess that possibility, two molecules were superposed against the crystallographic binding sites of the

**Table 6.9:** Recipe used in the evaluation of a stretchable di-ONJ ligand. The ligand was positioned in such a way that one of the terminal iminosugars matched the crystallographic structure of the ligand in the original protein structure.<sup>207</sup>

GENES	
Molecule	Load the protein model obtained after cleaning the PDB structure 2WBG (waters and ligands removed)
Molecule	Construct variants of the di-ONJ ligand using five fragments: <i>ONJ</i> + <i>linker</i> + <i>hydrophobic tail</i> + <i>linker</i> + <i>ONJ</i>
Torsion	Explore the free rotations of the resulting ligand construction
OBJECTIVES	
Contacts	Minimize steric clashes
Distance	Bring the other iminosugar end of the ligand as close as possible to the center of mass of the original crystallographic ligand in the X-ray structure <sup>207</sup>

dimer structure,<sup>207</sup> which features an analog inhibitor compound suitable for structural alignment. Then, dihedral torsions of the rotatable bonds of the ligand were analyzed looking for a combination that could have them interact at the interface. This interaction was implemented as a distance minimization between a carboxylic oxygen of one ligand and a carboxylic hydrogen of the other (see table 6.10 and fig. 6.10).

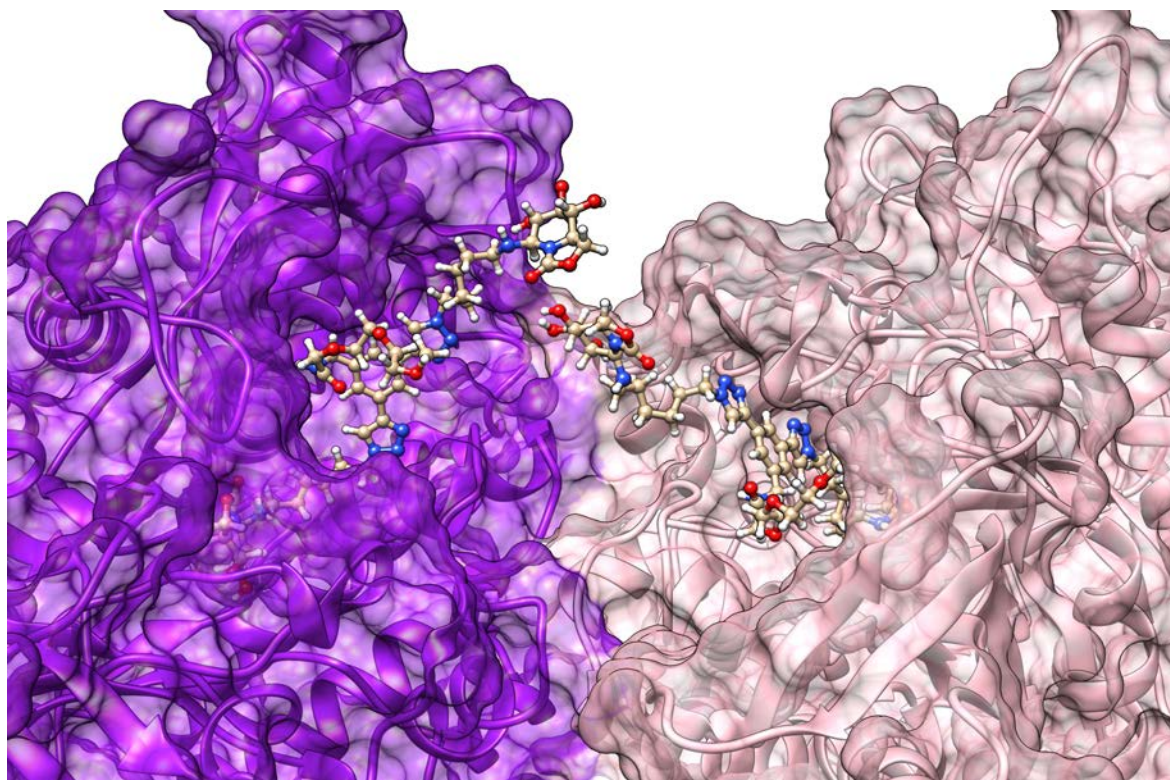
**Table 6.10:** Recipe used in the evaluation of a two di-ONJ ligands. The ligand was positioned in such a way that one of the terminal iminosugars matched the crystallographic structures of the ligand in the original X-ray protein structure.

GENES	
Molecule	Load the protein model obtained after cleaning the PDB structure 2WBG (waters and ligands removed)
Molecule	Load one copy of the di-ONJ ligand using a Mol2 file obtained with ChemCraft, positioned in the binding site of monomer A
Molecule	Load another di-ONJ molecule, but positioned in the binding site of monomer B
Torsion	Explore the free rotations of the di-ONJ copy in monomer A
Torsion	Explore the free rotations of the di-ONJ copy in monomer B
OBJECTIVES	
Contacts	Minimize steric clashes
Distance	Bring the free iminosugar ends of both ligand copies close together so they are able to form a H-bond

The analysis showed that this interaction is structurally feasible, which was further confirmed by an explicitly solvated, full-atom molecular dynamics trajectory: the interaction remained stable for more than 100 nanoseconds. Simulating this system (100,000+ atoms) for such a long period can take months with ordinary CPUs, but thanks to the GPU acceleration implemented in OpenMM and OMMProtocol (see section 5.2.2), these trajectories could be obtained within a week.<sup>||</sup>

<sup>||</sup>An equivalent protocol in the commercial, GPU-accelerated version of Amber installed in our facilities would





**Figure 6.10:** Two divalent ONJ models were anchored to their respective binding sites, and their free-torsion bonds were explored to find a pose where the two free iminosugar ends could interact at the interface via a H-bond.

While the computational model offers an answer to whether this structure is possible or not, this specific protocol cannot answer whether this interaction is favored. To assess that possibility, broader sampling would be needed (like metadynamics), which demanded more computational time than the available within the submission deadline. Additionally, there is no experimental information to support this hypothetical interaction: the stoichiometry suggested leans towards 1:1, and not 2:1.

### 6.2.3.3 DOES THE ROTAXANE COMPOUND FIT? CAN IT OCCUPY BOTH SITES? HOW?

The pillar[5]ene variants exhibit a slightly worse  $K_i$  but still comparable to the di-ONJ compound, so one would expect a similar interaction profile. The divalent ligand has been shown that it cannot reach both sites of a dimer, suggesting that its inhibition mechanism might be based on a sliding motion between the binding sites. However, the divalent ligand only represents half of the H-shaped component of the rotaxane compound. This has two conflicting consequences: (1) the H-shaped component is larger and could use the iminosugars of opposed axels to reach both binding sites, and (2) the volume of the crown component might work against this interaction through steric impediment. This raises two possibilities:

---

have taken two weeks.

1. The rotaxane interacts with the protein via iminosugars on the same axel.
  - (a) This interaction strategy does not offer any advantage over the divalent binding (its iminosugar-iminosugar range distance is the same).
  - (b) The steric impediments of the crown component are easier to solve, since rest of the structure would remain facing the outside part of the structure.
2. The rotaxane interacts with the protein via iminosugars on different axels.
  - (a) The iminosugar-iminosugar range distance is far greater and could enable accessing both sites simultaneously.
  - (b) The steric impediments of the crown are far greater, since the structure would be now in a less ideal orientation.

To assess both possibilities, the protein-rotaxane structure was analysed with GaudiMM following the same recipe as the single divalent molecule docking in section 6.2.3.1: one iminosugar was fixed in one binding site and the second one was instructed to get close to the second binding site with a distance minimization objective by exploring the free torsion of rotatable bonds. Steric clashes were minimized through a Contacts objective. See table 6.11 for more details.

In the binding mode A (same axel), the structure did not reach the second binding site, as expected; not even tolerating severe clashes. In binding mode B (different axels), the H-shaped component could reach both sites comfortably. There were clashes, but not as bad as expected: they were mainly due to internal clashes of the rotaxane. Given the unusually high number of freely rotatable bonds (176 in this case), more iterations would have been needed to optimize them out. However, that was not necessary, since the purpose of these GaudiMM calculations was to obtain a *good enough* structure to use as the starting point of the next step in the multiscale protocol.

Once parameterized with Antechamber,<sup>167</sup> two candidate structures of each binding mode were submitted to a molecular dynamics analysis with OMMProtocol. Both revealed stable bindings to their respective sites, with additional stabilization of the structure via internal cross-interactions.

The unsurprising results observed for binding mode A (same axel) agree with the experimental evidences. It exhibits the same binding profile as a single divalent molecule, compatible with the sliding mechanism, hence the comparable  $K_i$  values. The slight difference might be due to the entropic stabilization of the crown-component via secondary binding sites.

Unfortunately, while the binding mode B (different axels) showed a promising interaction profile, there is

**Table 6.11:** Recipe used in the evaluation of the pillar[5]ene ligand. The ligand was positioned in such a way that one of the terminal iminosugars matched the crystallographic structure of the ligand in one of the monomers of the original 2WBG protein structure.

GENES	
Molecule	Load the protein model obtained after cleaning the PDB structure <sup>207</sup> (waters and ligands removed)
Molecule	Load the structure of the pillar[5]ene as obtained through a preliminary 3D model in ChemCraft
Torsion	Explore the free rotations of the pillar[5]ene
OBJECTIVES	
Contacts	Minimize steric clashes
Distance	Bring one of the free iminosugar ends (depending on the case studied, from the same axel or from the one across) closer to the binding site in monomer B

no experimental evidence to back it up. If this binding mode was feasible, a higher  $K_i$  should be observed, but that is not the case.

#### 6.2.4 DISCUSSION & FURTHER WORK

This joint study was an excellent opportunity to show how GaudiMM can be a valuable asset for both experimental and computational communities. The computational feedback has provided illustrative models on what can be happening at the molecular level and even proposed alternative explanations to be confirmed experimentally. This can be argued in three points.

First, GaudiMM can provide results directly applicable to the wet-lab. The different di-ONJ variants tested opened doors to synthesizing ligands of optimum length that would explain how the di-ONJ ligand exhibits that excellent inhibition power without incurring in additional costs. Doing this experimentally would have involved more steps of synthesis and tests, only to discard most of the candidate ligands. With this computational framework, this can be obtained within a day. Of course, this does not replace the experimental data; it just reflects that computational assessment can at least provide a way to save material and human resources.

Second, it allows to create new types of computational studies in a simpler, consistent way. Testing if the di-ONJ compound or the rotaxane can reach both binding sites of a dimeric protein would be normally done with a steered molecular dynamics simulation. However, parameterization would be needed first. For the rotaxane alone, this would take more than a day. The actual simulation would take around a week. With GaudiMM, this can be obtained in hours. Then, if the results are positive, a MD simulation would be in

order. However, if the results did not show anything promising, those expensive computational and time resources could be invested in testing a different hypothesis. In the same fashion, testing if two divalent ONJ ligands can interact at the interface would usually be studied with molecular docking, but there is no software suite that can perform a multi-ligand, restrained study like the one herein presented.

Third, even if the researchers prefer to go straight to the MD stage without confirming the feasibility of the hypothesis first, they would still need to build the initial structure. The researcher usually constructs those manually, with the aid of an interactive 3D viewer and related tools. This is normally doable with small ligands, but it starts getting disturbingly complex when bigger structures are involved. Setting up a rotaxane model suitable for MD assessment would have involved hours of finetuning and trial-and-error attempts. With GaudiMM, these can be obtained automatically when the correct recipe is used.

Of course, GaudiMM is not the answer to every question. It only helps guide the creation of new hypothesis at the initial steps of the brainstorming. For more accurate results, higher levels of theory must be applied through more advanced protocols. Even molecular dynamics might not be enough if quantitative magnitudes are demanded, such as binding energy or free energy. To obtain those, one would have to employ broad sampling methods like metadynamics or free energy perturbation, hybrid schemes like QM/MM, or even QM calculations of reduced cluster models. Those are out the scope of this dissertation and could not be performed within the available timeframe.

### 6.3 FINAL CONCLUSIONS

Throughout this chapter, it has been shown that, while computational studies can be strictly theoretical, there is no point in denying that molecular modeling is a helpful tool for experimental works. *In silico* can go hand in hand with *in vitro*, and some research groups would argue that they must. It is common to see how experimental groups maintain strong alliances with theoretical groups. Fruitful joint efforts like this bring different points of view and ways of thinking to the discussion table, which can only enhance the brainstorming sessions, especially when counterintuitive phenomena like the aforementioned happen.



## General conclusions

**I**N this dissertation several computational tools have been presented and several applications have been benchmarked and showcased. Globally, the list of achievements could be summarized in six points:

1. GAUDIMM has been presented as a versatile molecular optimization framework with high modularity. Its uncoupled plurigenetic, multi-objective implementation provides researchers an unprecedented flexibility in molecular modeling. Instead of conforming to the requirements of a sequential multistep protocol, the same methods can work synergistically in the same modeling exercise. The concept of *recipe* paves the way towards performing hypothesis-driven modeling as well as other simulations like dockings or restrained conformational exploration.
2. TANGRAM is a collection of more than 15 tools for UCSF Chimera that will help in the generation of input files for 3<sup>rd</sup> party software and diverse interactive structural analysis within a single graphical interface and user experience.
3. OMMPROTOCOL provides a user-friendly, single-file interface to the powerful, GPU-accelerated OpenMM molecular dynamics libraries. These tools have brought a 20-fold speed increase to the previously followed MD protocols in our group.
4. GARLEEK has been designed to help in those QM/MM studies that require extended molecular mechanics force fields. By seamlessly interfacing Gaussian with modern MM suites, more accurate calculations can be obtained.
5. ESIGEN can save hours of manual text manipulation in computational chemistry. Its ability to automatically generate technical reports suitable for attachment as supporting information documents or internal communication with colleagues will be hopefully appreciated by this community. Computational chemists will also welcome EASYMECP, designed to facilitate the calculation of minimum energy crossing points (MECP) with Gaussian.

These ongoing efforts have been the first steps towards developing a suite able to compete, feature-wise, with available commercial suites —which can be particularly expensive in some cases— at no cost for academics.

## ON GAUDI MM

In addition to reproducing and benchmarking known problems, this platform has been able to model orphan systems where currently available information is scarce. This is thanks to a versatile approach: creating optimization synergies between deliberate simplistic chemical and geometric descriptors. Some tasks that have benefitted from this idea are:

- **EXOTIC DOCKING PREDICTION.** GaudiMM expands the possibilities of docking calculations beyond the traditional flexible protein-ligand dockings, enabling unconventional docking studies like competitive docking or multivalent restraints (see section 6.1.1 for a benchmark on standard protein-ligand docking and the take on more exotic cases).
- **COMPLEX MOLECULAR DESIGN.** Predicting possible structures of partially characterized systems by performing hypothesis-driven modeling (see section 6.2 on multivalent enzyme inhibition). This includes designing complex ligands where only some experimental information is available, if any (see section 6.1.1.3 for the optimization of a dibiotin ligand).
- **FINDING METAL BINDING SITES IN PROTEINS.** Modeling organic systems where metal-residue interactions can be expressed with coordination geometries (see section 6.1.2 for this and other cases of coordination-driven folding).

Additionally, the conceptual separation of exploration and evaluation as implemented in GaudiMM gives a clear understanding of the different variables involved in an optimization process. This has proved to be a very valuable as a teaching tool in lower degrees of education. Students involved in GaudiMM development have contributed new modules even with a non-chemical background. Some highlights include a gene to navigate the chemical space or a coupled gene/objective pair to assess ligand binding pathways, detailed in appendix B.

Of course, there is further work to do. GaudiMM's approach has a modestly steep learning curve and configuring an input file is mostly done on the text editor. A general-purpose graphic interface would be desirable and is something to consider. In the short term, the concept of application-specific interfaces is very attractive (e.g. searching metal binding sites or optimizing the length of linkers).

Analyzing results from a multi-objective process can be daunting at first because considering the optimality of two or more criteria simultaneously is not intuitive. While GaudiView is available to perform sorting and filtering on the candidate solutions, certain applications could benefit from a unified scoring term. However, this would require constructing a weighted linear sum of the objectives by benchmarking big datasets. In that matter, machine learning approaches could be very helpful.

## ON PYTHON

Without Python and its great ecosystem (UCSF Chimera, the SciPy stack and the Omnia project have been particularly important) this dissertation would not have been possible. All the developments carried out during this Ph.D. are the consequences of its unique vision.

The *de facto* Python installation already provides a library for high-level operations, freeing the developer from dealing with technical nuances. Beyond the official distribution, the catalog of ready-to-use packages is excellent, allowing to prototype projects in very little time just by importing the needed requirements. This is particularly true in scientific software, where it shines as the perfect glue language to stick different projects together.

Moreover, the emphasis on readability and self-documented code contributes to maintaining good practices along the full development cycle, even when different people are involved. This is particularly important for long-lasting efforts in research and fruitful investment in research.

This Ph.D. hopefully illustrates how Python and its exceptional ecosystem offer molecular modelers with a versatile canvas for innovative science.





# Epilog

During the development of new software, difficulties can arise anytime, for any reason. Dependencies, installation and distribution are inherent problems to the complex landscape of libraries, operating systems and hardware architectures. Solving them efficiently requires using developer-specific tools, usually disregarded by end-users. I began my Ph.D. studies as a user and ended up as some sort of developer, and to my surprise, my most popular project is not GaudiMM itself, but a tool created as a helper for its development: PyChimera. The need for interconnected software is patent, and a big part of this dissertation has been devoted to bringing new free alternatives to the table. The Tangram suite is only an attempt at providing molecular modeling tools accessible for users that do not want to mess up with complex installations and input files: the workflow has been designed to be intuitive and consistent.

Still, most complex tasks would require some sort of scripting for an efficient solution. Programming skills are essential in all fields of science that can be enhanced by computational support. Learning how to program can have a huge impact on the researcher workflow. Three main advantages can be identified:

- It can accelerate repetitive tasks, freeing time for other problems.
- It gives access to understanding existing code. This way, unexpected errors can be investigated thoroughly before having to ask for external help. It also allows to extend the original code with new functionality, if desired.
- It enables new problem-solving strategies and can help plan studies in a different way.

In other words, programming skills streamline creative thinking.

Initially, the *novel software platform* in the title of this dissertation only referred to GaudiMM, but during its development some aspects of the workflow of lab-mates and colleagues caught my attention. What started

as innocent efforts to help automate some manual steps or provide easier access to new technology ended up as big projects on their own. These opportunities were not anticipated, but perfectly fitted the intentions of this thesis.

Yet we are failing to convince students why coding is important. Most still prefer to go manually to *make sure* results will be correct, and not a false positive product of a non-obvious bug. Is this due to the uninformative error messages, or does it go deeper in our educational system roots? Students are still taught memorization and mechanization, but that should not be the point in this era. Computers are much better at that than us, and will surpass us in other areas too. Solving problems is not copying algorithms and following instructions. It should be more about the reasons behind each of those steps. Designing algorithms, protocols, tools and frameworks: that should be the goal. Otherwise, the inability to write code will become the illiteracy of the XXI<sup>st</sup> century.



# Perspectives for molecular modeling

**I**N an ideal future, there would be no need for multiscale protocols because accuracy compromises will not be needed in exchange for performance. However, to get there a large series of milestones must be conquered first. That path can only be pursued if there is a global interest.

## A.1 THE IMPACT OF MOLECULAR MODELING

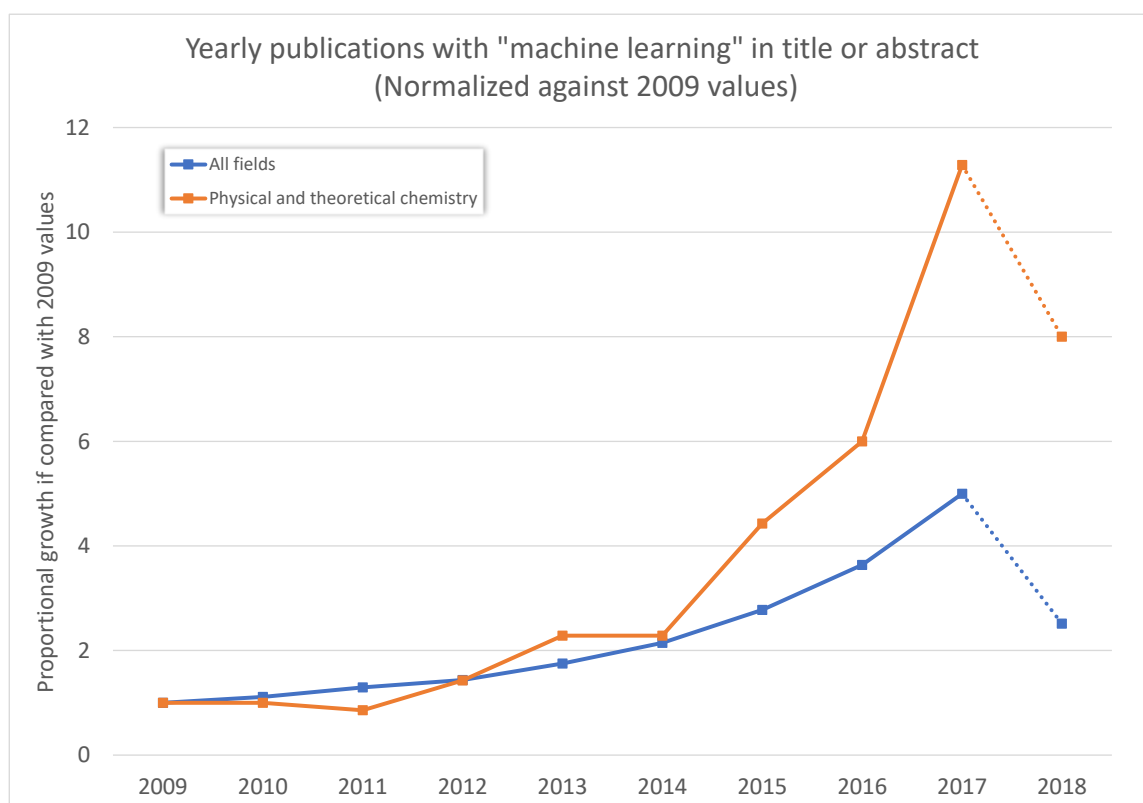
Such a vast array of tools and resources can only be product of thousands of researchers, both in the public and private sector, and such devotion can only come if the field is attractive enough. Computational modeling is widely regarded as one of the fastest growing sectors in science, as perceived by researches and engineers themselves. According to a recent survey from the European FP7 project MULT-EU-SIM22, which measures the impact of general modeling in science and engineering, 75 % of researchers see a high impact of modeling and simulation in their fields, and 70 % foresee a strong growth of these methods, with an impact far beyond the one currently achieved.<sup>208</sup> International institutions also believe in the trend and, in fact, there are several ongoing projects working on standardizing basic concepts such as the terminology to be employed.<sup>209</sup> The very existence of reports covering the topic<sup>210–212</sup> also serves as support for this general idea.

More concrete examples of this perception include the aerospace industry, which uses computational chemistry to better understand the effect of high temperatures and combustion on the stability of the coating present in the materials employed, thus increasing flight safety. Additionally, the longevity of nuclear reactors is affected by the impact of neutrons on the walls, which result in atomic displacement evaluable with computational chemistry. Better determining the life expectancy of the reactor and can potentially

save millions by preventing an early shutdown of the plant.<sup>213</sup> In pharma, measuring the heat of formation experimentally is 50 times more expensive than a comparable DFT study.<sup>2</sup>

These anecdotal examples can be quantified by analyzing some metrics on each of the three levels of the knowledge transmission model:<sup>214</sup> (1) Authors, (2) Users, (3) Society. The authors of theories and models (1), usually belonging to the academia, publish their findings to scientific journals, which end up in software products that can be used by professional modelers (2), leading to process improvements. This directly benefits society with lower prices in value products (3).

The increased popularity in basic research can be measured by the number of published manuscripts mentioning the topic, as well as their specific proportion within the field and impact factor. Observing the increasing presence of techniques such as DFT or molecular dynamics is a good proxy to the trend.<sup>2</sup> An updated example can be seen in fig. A.1.



**Figure A.1:** Publication trends of manuscripts in molecular modeling in the title or abstract against all publications in chemistry related fields. Values are normalized against records in 2009. While publications in all chemistry failed exhibit a slower growth, articles in the field of molecular modeling (as represented with two popular methods, MD and DFT) have grown faster in the past decade. (Data obtained through <https://app.dimensions.ai>).

The direct application of methods is measurable by looking at the number of patents on the topic,\* the return rate in the corresponding industries (between 3:1 and 10:1 in pharma<sup>215</sup>), or the specific job postings demanding such experience.†

In addition to the jobs themselves, which can be regarded as direct benefit for society, some other numbers could be thrown, such as the estimated contribution to the GDP by chemistry research (1.4% in UK as of 2010<sup>213</sup>), or the attributed spending on high performance computing by the modeling sectors, featuring bio-sciences, chemical engineering or computer-aided engineering as top contributors.<sup>216</sup>

## A.2 WHAT THE NEXT GENERATION WILL BRING TO THE TABLE

Published almost twenty years ago, the chapter *Vision 2020: Computational Needs of the Chemical Industry in Impact of Advances in Computing and Communications Technologies on Chemical Science and Technology: Report of a Workshop*<sup>217</sup> cited five main computational challenges for the chemical industry: Predicting (1) biological activity and (2) toxicity of a chemical structure, and designing (3) catalysts, (4) chemical processes and (5) materials. This englobes two intertwined areas: prediction and design. For this to happen, the report points that intense research must be carried out in, amongst others, the potential functions of MM-based methods, long MD simulations for large ensembles (in the millisecond scale), quantum effects, solvent effects, solid state structure, multiscale protocols (atomistic, micro-, meso-, and macroscopic). Most of the challenges are accuracy or scale related, so huge efforts must be invested to reach errors within 0.1-0.2 kcal/mol in thermochemistry or to design universal and polarizable force fields, to cite two examples. This is not only a matter of scientific software development, but also responsibility of computer architecture, operating systems and networks. Since a single processor can only go so fast, tera-, peta- and exascales can only be achieved with parallel scaling, both within the processor itself (multicore architectures) and across symmetric machines (nodes within a cluster). For this to work reliably, operating systems and networks must be designed with fault tolerance in mind: if a core or node fails, the whole ensemble might fail as well.

We are almost in 2020 now, and part of the predictions and demands have been fulfilled. Massively parallel architectures are now inevitably present and software has been slowly adapting to the new design paradigms. Any research group or company can get access to these resources thanks to the ubiquitous *cloud*, which offer hardware solutions on demand. The so-called *as a Service* products (Software as a Service, Platform as a Service...) allow per-usage payments without having to worry about maintenance or resource constraints.

---

\*For example, at <http://www.wipo.int/patentscope>.

†Custom searches can be performed in websites such as <http://chemjobber.blogspot.com/>, <http://www.linkedin.com>, <http://glassdoor.com> or <http://www.stackoverflow.com>.

If a given simulation needs more storage, memory or calculation speed, more nodes can be added to the ensemble with a click. Platforms such as Amazon's AWS, Google's Cloud or Microsoft's Azure<sup>‡</sup> provide the raw infrastructure, which can be configured by the researchers or employees themselves, but is more commonly setup by specialized companies devoted to this newly found market niche.

What this report did not anticipate was the advent of GPGPU (General Purpose Graphical Processor Unit) computing: the advances in 3D acceleration and desktop graphics cards proved to be a massively parallel architecture that could be exploited by software not related to games and visualization. Molecular Dynamics simulation have seen a drastic performance increase thanks to this new paradigm, implemented in major MD software (Amber, Charmm, Gromacs, NAMD, HTMD, AceMD, OpenMM...) and is now possible to simulate hundreds of nanoseconds a day with a sub-1000\$ personal desktop, thus getting closer to the millisecond-scale proposed that, while not routinely common, is starting to hit journals more often.<sup>218,219</sup> Quantum Mechanics could certainly benefit from GPU acceleration, but the offer is still reduced (BigDFT,<sup>220</sup> TeraChem<sup>221</sup>). The following years would certainly see a mainstream presence of GPU-implemented QM methods.

This would be in agreement with the 2017 Grimme's computational chemistry wish list for the upcoming 25 years: (1) Development of robust and fast electronic structure methods with chemical accuracy for all conceivable chemical processes, all states (gas, liquid, solid), and all (even exotic) types of spectroscopies, (2) Seamless and automated multilevel modeling, including error estimates, (3) Routine treatments for many nuclear degrees of freedom and entropy, (4) Inclusion of solvation effects, (5) Prediction of molecular as well as macroscopic (bulk) properties, and (6) Automated approaches for finding new reactions. Warshel, in his 2014 Nobel Lecture, pointed to broader future directions, like using molecular modeling to fight drug resistance, grasp a deeper understanding of protein-protein interactions, truly rational enzyme design or developing molecular machines; for all of them, multiscale strategies will be necessary, he concluded.<sup>222</sup>

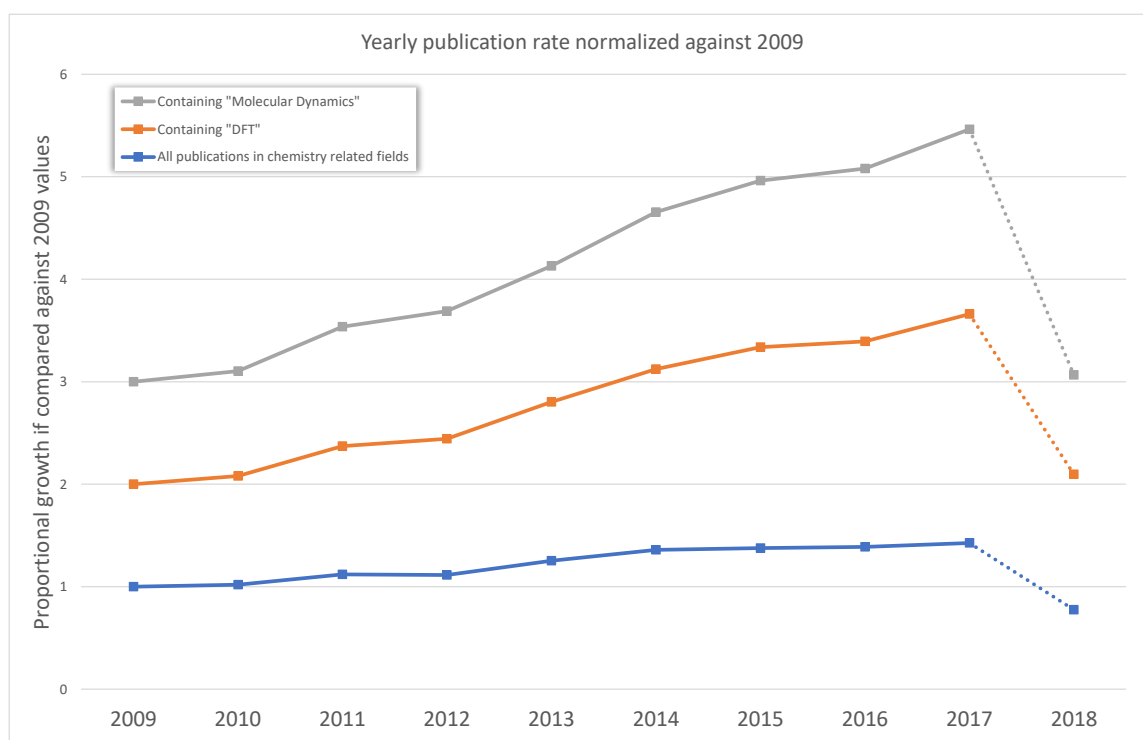
These predictions can be further extended with more specific wishes, like universal reactive force fields or cheap, large-scale QM methods, two trends that will inevitably close the gap between these traditionally divergent approaches. Faster architecture and software will possibly allow for more robust *ab initio* protein structure and folding studies,<sup>223</sup> and cheminformatics tools like (3D)QSAR will also see advances.<sup>224</sup> Also, thanks again to hardware advances originally intended for gaming, Virtual and Augmented Realities will become mainstream and that should also influence molecular modeling software, whose graphical interfaces, built around an interactive 3D viewer, will be certainly enriched. As a matter of fact, several suites already include preliminary support.<sup>190</sup> Together with mobile and web platforms, desktop software will surely evolve

---

<sup>‡</sup><https://aws.amazon.com>, <https://cloud.google.com>, <https://azure.microsoft.com>, respectively

to new interface paradigms.

Finally, a very hyped topic lately is the incursion of machine learning and neural networks in scientific software. After gaining a huge popularity for successfully solving problems traditionally understood as *easy for humans but hard for computers* (i.e. facial recognition, natural language interfaces, speech synthesis), it is now overflowing to fields like computational chemistry. Being such a hot topic, a lot of publications have arisen in the last years (see fig. A.2). While some see these proposals as the definite solution to some chemistry problems like QSAR<sup>225–231</sup> or even DFT-trained electronic predictions,<sup>232</sup> a certain skepticism is also held by others,<sup>233,234</sup> especially when it comes to the pharma industry and drug design.



**Figure A.2:** Number of publications containing *machine learning* in title or abstract, per year. Data is normalized against the values recorded for 2009. Physical and theoretical chemistry exhibit a steeper growth in the recent years.

Besides traditional computers, nascent quantum computing will be able to implement some algorithms with unprecedented efficiency. Computational chemistry will be one of the most benefitted fields in that regard,<sup>235–237</sup> as prototyped in several recent attempts.<sup>238–241</sup>





# B

## GaudiMM as an educational tool: undergoing developments

### B.1 NAVIGATING THE CHEMICAL SPACE

GaudiMM already allowed to navigate the chemical space via the dynamic building capabilities of the Molecule gene, but it presented two limitations: (1) it is restricted to the provided fragments library, and (2) it only allows to construct linear concatenations of those fragments (i.e. no ramifications or rings).

A new approach based on graph theory and pharmacophore matching is being developed in our group as part of the Ph.D. thesis of J. E. Sánchez-Aparicio. This method, which interprets molecules as non-directed graphs that can grow and shrink arbitrarily, does not require any preexisting libraries and naturally considers ramifications. It has been successfully applied to propose designs of small molecule inhibitors for *K. pneumoniae* NDM-1  $\beta$ -lactamase.

### B.2 FINDING LIGAND BINDING PATHWAYS

Docking studies provides insight on how a small molecule can interact with a bigger host molecule by assessing feasible binding poses. However, those are just static snapshots of a dynamic behavior. To study how the ligand reaches its binding sites, long Molecular Dynamics with steering restraints are needed and do not always guarantee a successful ligand pathway.

An alternative approach was considered for one of the MSc dissertations supervised during this Ph.D. The

protein space was flooded with small probes placed in a tight grid and queried for steric impediments, resulting in points with higher or lower pseudo-energy scores. Then, lower-energy points were traversed from the outer regions of the protein in hopes of finding a continuous path that reached the ligand binding site. To consider the ligand size, shape or volume, a second step was proposed. The calculated paths were segmented in 5 Å pieces and each of the resulting pieces was then submitted to a docking simulation with reduced search radius. The resulting structures were low-energy conformations of the ligand along the proposed pathway. All these poses were finally concatenated together to emulate a smooth trajectory ideal for depiction purposes.

This proof of concept proves how the versatility present in GaudiMM can be used as part of bigger protocols, and is being reimplemented as a gene able to guide the exploration of docking studies along feasible pathways in the Ph.D. studies of J. E. Sánchez-Aparicio.



# Living with metal ions in molecular modeling

One of the most exciting areas of molecular modeling sits at the frontier between organometallics and biochemistry, two fields that have been studied separately in computational chemistry for decades now. Globally, chemists exploit their features differently and, as such, present different computational challenges. Traditionally, organometallic systems feature a reduced number of atoms and accommodate transition metal centers within their structure, whose exotic electronic behavior can only be accurately computed with quantum chemistry approaches. Studies on biological problems such as the early work on folding of peptides and proteins had to face a larger number of atoms (hundreds or thousands) from the beginning, forcing the authors to use classical mechanics approaches to deal with the added dimensionality after realizing that the electronics of the system were not very important in that process.

However, metals do take part in biological processes as mainstream as oxygen transportation and muscle contraction. As such, the existence of metalloproteins cannot be neglected by the modeling community, who should bring these two areas together in a more seamless experience. Given the diverging efforts accumulated for decades, the gap is not easily overcome, but some solutions exist. Depending on the properties to study, one can resort to different approaches, as detailed below.

## C.1 QUANTUM MECHANICS

Since quantum mechanics deal explicitly with the electronic shells of atoms, the immense diversity of electronic configurations of metal ions does not represent a problem. If such, the only challenge this might

present is choosing the adequate functionals, basis sets or starting-point structures.

The challenge is more technical than scientific. While advances in DFT theories and hardware architectures allow us to deal with up to 500-atom systems in feasible timescales, this is still far from the number of atoms usually present in protein structures. For this, hybrid QM/MM studies are more adequate: the QM layer is responsible for dealing with the metal and its surroundings (at least, the first coordination sphere), while the comparatively cheap MM layer governs the rest of the structure. Even with this approach, time-dependent schemes still represent a huge computational effort, not to mention the difficulties in setting up the system adequately. One must still deal with layer boundaries effects or the parameterization needed for the MM calculations.

## C.2 MOLECULAR MECHANICS

Sometimes, QM is not necessary for a modeling study, since the metal might only play a structural role without exhibiting reactivity. In these cases, it is more interesting to gather an insight into the structural behavior of the system along time. Nowadays, for macromolecular systems, this is only feasible with molecular dynamics approaches, which require accurately parameterized force fields. Traditionally, force fields were developed to solve problems existing with proteins, nucleic acids and organic compounds,<sup>63,66,68</sup> so historically transition metals have not been considered in force field development. Additionally, they present complexities not present in the reduced set of organic elements: several coordination geometries, different charge states, exotic polarizable behavior... As a result, dealing with metals in molecular mechanics is usually challenging. One must choose between (1) not considering them at all, (2) using a low-accuracy general-purpose force field, or (3) facing the tedious process of parameterization.

Ignoring or removing the metal ions can be acceptable in certain cases where they do not play a crucial role in the structure or dynamics of the system, but that is rarely the case. While general purpose force fields are numerous and heavily used, they mostly target organic compounds (such as CGenFF,<sup>242</sup> GAFF,<sup>243</sup> Tripos 5.2 force field<sup>244</sup>). Only some include parameters for metal ions: UFF (for Universal Force Field,<sup>178</sup> MMFF<sup>245</sup>) covers the full periodic table, but Dreiding<sup>179</sup> only contains parameters for Na, Ca, Zn and Fe. While useful for organic chemistry, they are not as used in simulations including biological systems, since they tend to rely on the Lennard-Jones based *nonbonded model*.

A feasible alternative for bio-containing systems is the so-called *bonded model*, which treats metal ion interactions with both bonded and non-bonded parameters; i.e. the metal is assumed to bond to some residues.

Some protein-oriented force fields like AMBER<sup>12</sup> or CHARMM<sup>70</sup> distribute force field extensions for some of the most common metal ions in proteins, such as hemo-coordinated iron, but mainly as examples on how custom parameters can be added in the software. These types of force field extensions are only valid for the context where the parameters were obtained; i.e. the iron parameters for the heme groups will not reproduce the behavior of iron in other organic contexts such as ferrocenes. While the file format is easily understood, the values of the parameters are not easy to obtain: one has to resort to experimental data or *ab initio* calculations to get adequate constants for bonded (distances, angles, dihedrals) and nonbonded (electrostatic, Van der Waals) interactions. While an expert user can decide to obtain those values manually, the process is not trivial and some protocols and tools have appeared to assist. They are mostly based on the Seminario's method and his FUERZA software,<sup>246</sup> such as MCPB, MCPB.py,<sup>247</sup> VFDFIT.<sup>248</sup> Recently, alternative approaches based on machine and statistical learning,<sup>249,250</sup> and non-Seminario strategies<sup>251,252</sup> have also appeared, but the principle remains the same: extract the information from *ab initio* calculations. Given the complexity of the task, some specific force fields have arisen lately to provide parameters for certain metals.<sup>253–259</sup>

A radically different strategy consists of mimicking the interactions of the metal site with positively-charged pseudoatoms strategically placed at around 0.9 Å from the metal nucleus following the vertices of the adequate coordination geometry. The Cationic Dummy Atom Model (CDAM) was introduced for Mn<sup>2+</sup> ions by Aqvist & Warshel in 1990<sup>260</sup> and has been successfully implemented in further studies for Zn, Mg, Ca, Fe, Co, Ni, Cu and more.<sup>159,261–266</sup> Among its advantages, once parameterized the CDAM approach is context-independent, but it forces a fixed coordination number and geometry on the modeled metal site.

The application of polarizable force fields (Fluctuating Charge methods, ABEEM, Drude oscillators and rods, induced dipoles, AMOEBA, PFF) or more exotic models based on Angular Overlap and Valence Bond Theory are also promising approaches, but the additional calculations incur in a big performance penalty when compared to other strategies and still require additional parameterization. Further details on the topic can be found in the extensive review published by Li and Merz Jr. in 2017.<sup>267</sup>

### C.3 LOWER LEVELS OF THEORY

If the study at hand does not require a molecular mechanics treatment, such as docking studies of virtual screening approaches, the parameterization problem is usually not present or, at least, not that complicated. Docking studies, which try to accommodate small compounds within macromolecules, have not considered metals for years, since they were originally designed to find drug-like, organic compounds suitable for the pharmaceutical industry. Fortunately, over time some of the most popular docking packages have included

strategies to deal with metals,<sup>268</sup> albeit sometimes they could only be part of the host (usually a protein), and not part of the probe (the ligand).<sup>269</sup> To overcome the problem, approaches inspired in the Cationic Dummy Atom Model implemented in MM studies have been designed (*H-bond trick*): in this case, the dummy atoms are hydrogen atoms that behave as a hydrogen bond donor, a chemical feature commonly implemented in docking software.

Other approaches involve considering the metal problem as a geometric optimization problem, restraining their position with distances, angles and dihedrals measurements. This strategy is partially implemented in homology modeling software like MODELLER,<sup>270</sup> and is one of the main features of the developments presented in this thesis (detailed in chapter 3).

In cheminformatics, explicit consideration of atoms is not as important and strategies like the pharmacophoric studies only have to consider metals as a custom type of interaction hotspot.<sup>271-274</sup> In QSAR, a catalogue of metal empirical properties is enough to build the dataset.<sup>275</sup>

# D

## Tangram extensions for analysis

### D.1 INTERACTION ANALYSIS

#### D.1.1 GAUDIVIEW

GaudiMM, described in chapter 4, can generate tens of solutions including several *good-enough* answers to the problem posed due to its multi-objective nature. Seeing them all in UCSF often meant waiting for all the files to load beforehand, even the ones you might not be interested in seeing. Additionally, hiding the current one to show the following one required more than one action. As a result, the GaudiView graphical interface was designed to overcome those difficulties by providing the following features:

- Provide a tabular view of the results listing all the solutions in rows, and objective scores in columns. Rows can be sorted by one or more columns and filtered out by providing one or more cutoffs depending on the value of one column.
- Since the result index (\* .gaudi-output file) already contains the list of filenames and their scores, this is enough to display the initial table. Actual molecule objects are only loaded when its row is selected. This allows for fast browsing of only the requested solutions, without initial loading times.
- Every time one or more new rows are selected (with a mouse click or with keyboard arrows), the previously selected rows are hidden and the new ones are displayed.
- New selections can run any Chimera command specified in the command-line field below the table. This can be really useful to update the displayed residues around a ligand in protein-ligand docking, for example.
- Some clustering and rescoring utilities are also included for deeper analysis.



The architecture behind GaudiView does not depend on the initial data structure: a preprocessing step is performed to build the tabular data view, that ultimately servers molecules to the interactive canvas. Thanks to that, it's easy to integrate other file formats that can benefit from this interface. Currently, GaudiView accepts solutions from GOLD and arbitrary lists of Mol2 files. In the future, more docking programs could be integrated, like AutoDock Vina or DOCK.

### D.1.2 NCIPlotGUI

NCIPlot is a widely used visualization method<sup>160</sup> that uses non-covalent interaction indices derived from electronic density and its derivatives to help distinguish attractive interactions like Van der Waals, London dispersion forces or hydrogen bonds from repulsive ones like bad steric impediments. The original implementation is a FORTRAN program that requires specific input file with atomic coordinates and special keywords. While not difficult to write, it is still a small entry barrier.

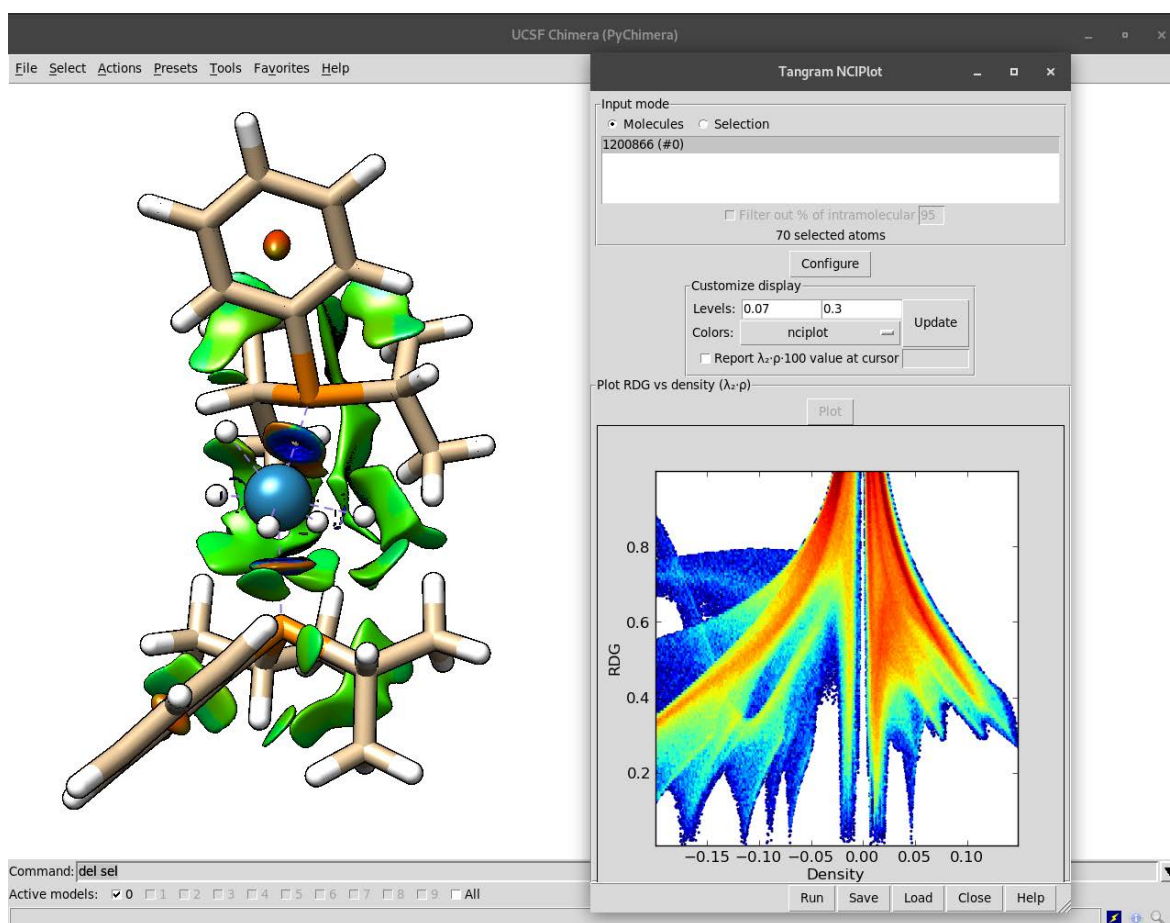
With NCIPlotGUI, the input file is automatically generated from any opened molecule in UCSF Chimera and the calculation is run in the background. When the program is done, the results are loaded in the same UCSF Chimera instance and plotted as colored volume maps (see fig. D.1). For large numbers of atoms, an alternative, 40-times faster CUDA implementation of the NCIPlot method<sup>276</sup> is also supported and recommended for GPU-enabled computers.

### D.1.3 PLIPGUI

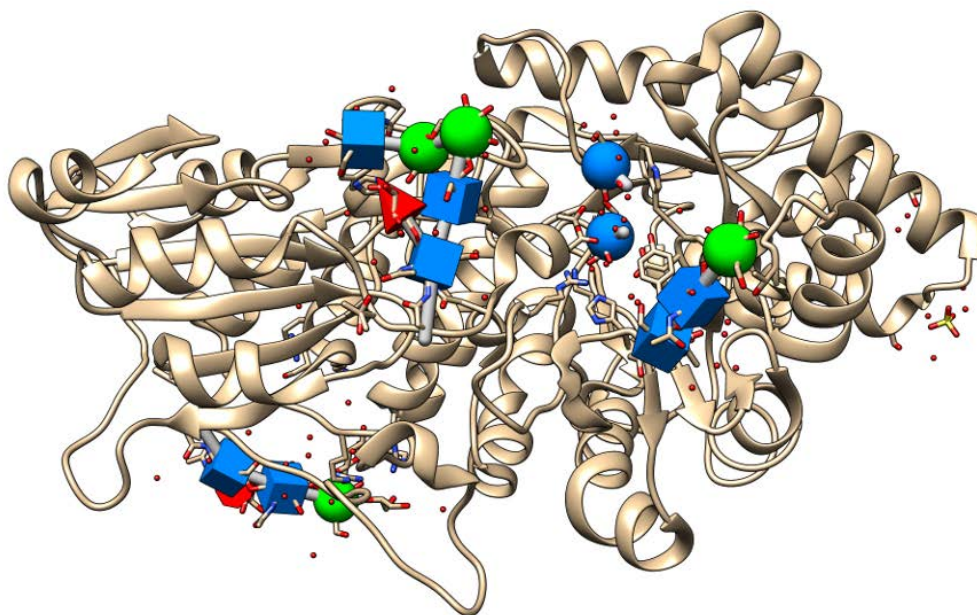
Protein-Ligand Interaction Profiler (PLIP)<sup>161</sup> is a Python utility to identify, list and represent non-covalent interactions between protein-ligand complexes. It depends on OpenBabel and VMD to work, but some UCSF Chimera integration is available. PLIPGUI is a Chimera extension that wraps PLIP in a graphical interface so all the tasks can be performed in a single program. The resulting will list all the identified interactions with a dynamic table that is updated depending on the binding site selected (if multiple are present). This can be coupled with docking studies to identify additional features implicitly described in the docking score.\*

---

\*For example, in GaudiView, the included `plip` command can be run for each solution, illustrating the possible cooperative tasks enabled with Tangram.



**Figure D.1:** Non-Covalent Interaction analysis of the partial structure of KUJLIK CSD structure,<sup>277</sup> with 70 atoms. The interface shows the input and configuration forms, as well as the Reduced Density Gradient (RDG) versus Density plot.



**Figure D.2:** 3D-SNFG representation of glycoside exohydrolase from *Hordeum vulgare*.<sup>278</sup>

## D.2 STRUCTURE ANALYSIS

### D.2.1 3D-SNFG

Glycoproteins are proteins that feature oligosaccharidic cofactors and are actively researched for its involvement in recognition processes, metabolism and allergies. However, since oligosaccharides are usually chains of different variations of 6 or 5-member carbon heterocycles with hydroxyl-containing substitutions, it is difficult to differentiate them visually when using classic 2D or 3D depictions. For that reason, the GLYCAM committee decided on a standardized 2D representation using colored geometric shapes called Symbol Nomenclature for Glycans (SNFG).<sup>162</sup> A 3D implementation for VMD was developed by Thieker<sup>163</sup> in TCL language, and is the original 3D-SNFG project. This is a reimplementaion of the same idea, but using Python and UCSF Chimera. It provides three alternative depictions, and the possibility to customize sizes and scales without modifying the source code (as it was expected in the original TCL implementation). The representation (see fig. D.2) can be switched on with the `snfg` command and switched off with `~ snfg`.

### D.2.2 BONDORDER

UCSF Chimera does not consider bond orders in the connectivity information it stores or represents. An approximate calculation is done during the parsing stage of molecule files to compute some internal atom types, but then that data is discarded. For some jobs, this information is important, though.

This extension provides a way to define an *order* parameters manually in chimera.Bond objects, so it can be used by other extensions that could rely on it. For example, QMSetup could use it to write the connectivity matrix using proper bond orders instead of the default 1.0. When the order attribute is present, this extension enables alternative representations of the bond with additional decorations in the cylinder.

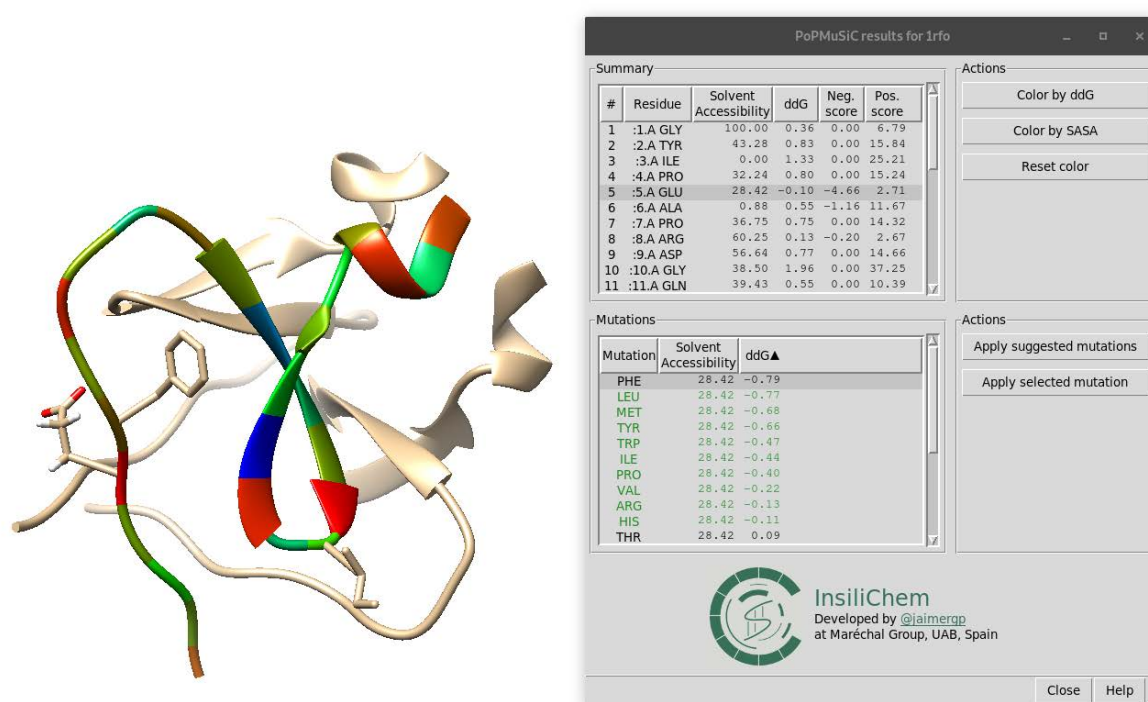
Additionally, the bond order information can be automatically computed with external libraries like RDKit, OpenBabel or AmberTools. The algorithms employed in that case are only applicable for small molecules though, so some work is needed when dealing with macromolecules. In those cases, template structures for common residues could be applied.

### D.2.3 ORBITRAJ

OrbiTraj patches the Molecular Dynamics trajectory viewer already present in Chimera and adds support for loading volume files for each frame. For example, this can be useful for QM optimization calculations where orbitals data have been generated for each frame. By using the OrbiTraj patch, the XYZ trajectory can display the orbitals volumetric isosurfaces along the way, thus representing electronic density transfer. The package also ships some independent Python scripts that can be used to convert WFN files as provided by Gaussian to CUBE files compatible with UCSF Chimera loaders.

### D.2.4 POPMUSICGUI

PoPMuSiC<sup>106</sup> is a web service that can calculate potentially stabilizing mutation sites in protein and peptide structures. Users need to register an account before submitting their files, and once the results are computed, they can be download from the user web panel. The results are plain-text files that list the different mutations associated to each residue position and their calculated score. PoPMuSiCGUI can open these files along with the submitted protein structure and depict those scores in a dynamic, two-panel tabular view. Residues can be colored according to its *mutability* score: positions that would stabilize under certain mutations will have a positive score and colored in a shade of green proportional to that score, while non-stabilizable positions



**Figure D.3:** PoPMuSiC results for the trimeric Foldon of the T4 phagehead fibrin.<sup>279</sup> One of the monomers has been colored according to the stabilizing potential of a mutation in that position (red being destabilizing, green neutral, and blue stabilizing).

would have a negative score and a red shade. Additionally, residue positions can be mutated to one of the proposed substitutions by using the Dunbrack's<sup>140</sup> and Dymameomics<sup>141</sup> rotamer libraries implemented in UCSF Chimera, which will have the changes immediately applied in the interactive 3D canvas.

### D.2.5 PropKaGUI

PropKa is a Python library developed by Jensen<sup>117</sup> that calculates pKa values of protein residues under different environment pH values. PropKaGUI wraps this package to make it usable in UCSF Chimera with a simple graphical interface. After selecting the opened molecule to be analyzed and the pH value, the PropKa routines are run and the results are shown in a new dialog listing the calculated pKa value for each residue. Adequate hydrogens can be added in situ by taking that information into account with the *addh* command in UCSF Chimera.

### D.2.6 SUBALIGN

UCSF Chimera provides several utilities for molecular superposition. The *matchmaker* command allows to efficiently superpose protein structures using sequence alignment and homology score matrices as guiding criteria. For non-protein structures, the simple *match* command is able to obtain the optimal superposition of two molecules, but only if atom pairs correspondences are manually provided. Several algorithms exist to identify the best atoms correspondences automatically,<sup>280,281</sup> but none of them are implemented in Chimera. The SubAlign extension provides a command (no graphical interface currently) to superpose small molecules by applying several alignment protocols implemented in RDKit.<sup>164</sup> The root-mean-square deviation (RMSD) of the superposed molecules is also provided as a result of the alignment, so it can be used for that kind of analysis as well. If more than two molecules are provided, all of them are aligned against the first, and the average RMSD is reported. In the future, more algorithms can be implemented, with a particular focus on those coming from the Computer Vision field, where Point Set Registration problems are common.



# Bibliography

- [1] D. M. C. Hodgkin, "Molecular model of penicillin (Science Museum Group Collection)," 1945 (accessed 2018-07-12). <https://collection.sciencemuseum.org.uk/objects/co417245/molecular-model-of-penicillin-by-dorothy-m-crowfoot-hodgkin-england-1945-penicillin>.
- [2] E. J. Maginn, "From Discovery to Data: What Must Happen for Molecular Simulation to Become a Mainstream Chemical Engineering Tool," *American Institute of Chemical Engineers*, vol. 55, no. 6, pp. 1304–1310, 2009.
- [3] N. Prize, "The nobel prize in chemistry 2013," 2014 (accessed 2018-07-14). [http://www.nobelprize.org/nobel\\_prizes/chemistry/laureates/2013/](http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2013/).
- [4] D. Robinson, "The Incredible Growth of Python," 2017 (accessed 2018-07-14). <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>.
- [5] P. S. Foundation, "Python success stories," 2018 (accessed 2018-07-14). <https://www.python.org/about/success/>.
- [6] Quora, "What is Python used for at Google?," 2014 (accessed 2018-07-14). <https://www.quora.com/What-is-Python-used-for-at-Google>.
- [7] Quora, "How is Python being used at Facebook?," 2014 (accessed 2018-07-14). <https://www.quora.com/How-is-Python-being-used-at-Facebook>.
- [8] R. Rapoport, B. Moyles, J. Cistaro, and C. Bertram, "Python at Netflix," 2013 (accessed 2018-07-14). <https://medium.com/netflix-techblog/python-at-netflix-86b6028b3b3e>.
- [9] W. L. DeLano, "Pymol: An open-source molecular graphics tool," *CCP4 Newsletter On Protein Crystallography*, vol. 40, pp. 82–92, 2002.
- [10] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin, "UCSF Chimera – a visualization system for exploratory research and analysis," *Journal of Computational Chemistry*, vol. 25, no. 13, pp. 1605–1612, 2004.
- [11] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, *et al.*, "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics," *PLoS Computational Biology*, vol. 13, no. 7, p. e1005659, 2017.
- [12] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, and P. Kollman, "AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules," *Computer Physics Communications*, vol. 91, no. 1-3, pp. 1–41, 1995.
- [13] O. Trott and A. J. Olson, "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading," *Journal of Computational Chemistry*, vol. 31, no. 2, pp. 455–461, 2010.
- [14] F. Maseras and K. Morokuma, "IMOMM: A new integrated ab initio+ molecular mechanics geometry optimization scheme of equilibrium structures and transition states," *Journal of Computational Chemistry*, vol. 16, no. 9, pp. 1170–1179, 1995.



- [15] S. Humbel, S. Sieber, and K. Morokuma, "The IMOMO method: Integration of different levels of molecular orbital approximations for geometry optimization of large systems: Test for n-butane conformation and sn 2 reaction: Rcl+ cl-," *The Journal of Chemical Physics*, vol. 105, no. 5, pp. 1959–1967, 1996.
- [16] M. Svensson, S. Humbel, R. D. Froese, T. Matsubara, S. Sieber, and K. Morokuma, "ONIOM: a multilayered integrated mo+ mm method for geometry optimizations and single point energy predictions. a test for diels-alder reactions and pt (p (t-bu) 3) 2+ h2 oxidative addition," *The Journal of Physical Chemistry*, vol. 100, no. 50, pp. 19357–19363, 1996.
- [17] V. Muñoz Robles, E. Ortega-Carrasco, L. Alonso-Cotchico, J. Rodríguez-Guerra Pedregal, A. Lledós, and J.-D. Maréchal, "Toward the computational design of artificial metalloenzymes: From protein–ligand docking to multiscale approaches," *ACS Catalysis*, vol. 5, no. 4, pp. 2469–2480, 2015.
- [18] I. Drienovská, L. Alonso-Cotchico, P. Vidossich, A. Lledós, J.-D. Maréchal, and G. Roelfes, "Design of an enantioselective artificial metallo-hydratase enzyme containing an unnatural metal-binding amino acid," *Chemical Science*, vol. 8, no. 10, pp. 7228–7235, 2017.
- [19] I. Tuñón and V. Moliner, eds., *Computational Methods in Enzyme Catalysis*, ch. 15, p. 514. Royal Society of Chemistry, 2016.
- [20] P. E. Bourne, H. M. Berman, B. McMahon, K. D. Watenpaugh, J. D. Westbrook, and P. M. Fitzgerald, "Macromolecular crystallographic information file," *Methods in Enzymology*, vol. 277, no. 1977, pp. 571–590, 1997.
- [21] H. M. Berman, "The Protein Data Bank: A historical perspective," *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 64, no. 1, pp. 88–95, 2008.
- [22] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, "Open Babel: An Open chemical toolbox," *Journal of Cheminformatics*, vol. 3, no. 1, 2011.
- [23] Schödingen LLC, "Maestro."
- [24] D. S. Biovia, "Discovery studio modeling environment," *San Diego: Dassault Systèmes*, 2017.
- [25] Chemical Computing Group Inc., "Molecular operating environment (MOE)," 2016.
- [26] OpenEye, "Lead Suite."
- [27] Industry Arc, "Computational medicine and drug discovery software market: By tools (software, databases and others); by application (drug discovery and development, disease modeling, medical imaging and others) & by geography - forecast (2018-2023)," 2018 (accessed 2018-07-13). <https://industryarc.com/Report/1252/computational-medicine-drug-discovery-software-market-analysis.html>.
- [28] Grand View Research, "Structural biology & molecular modeling techniques market analysis by tools (saas & standalone modeling, homology modeling, threading, molecular dynamics, ab initio, visualization & analysis, databases), by application, and segment forecasts, 2018 - 2025," 2017 (accessed 2018-07-13). <https://www.grandviewresearch.com/industry-analysis/structural-biology-and-molecular-modeling-technique-market>.
- [29] Accuray Research LLP, "Global computational biology market analysis & trends - industry forecast to 2025," 2017 (accessed 2018-07-13). <https://www.researchandmarkets.com/research/fxp5fw/global>.
- [30] Markets and Markets, "Biosimulation market by product (software, molecular simulation, in house, contract services), application (clinical trials, pkpd, adme), delivery (subscription, ownership), end user (biotech, pharma companies, cros, regulatory) - global forecast to 2022," 2018 (accessed 2018-07-13). <https://www.marketsandmarkets.com/Market-Reports/biosimulation-market-838.html>.
- [31] M. Frisch, G. Trucks, H. Schlegel, G. Scuseria, M. Robb, J. Cheeseman, G. Scalmani, V. Barone, G. Petersson, H. Nakatsuji, *et al.*, "Gaussian 16, revision a. 03," *Gaussian Inc., Wallingford CT*, 2016.
- [32] Schrödinger, Inc., "Schrödinger acquires pymol," 2010 (accessed 2018-07-13). <https://www.schrodinger.com/news/schr%C3%B6dinger-acquires-pymol>.
- [33] OMICtools, "Trends, benchmarks and insights in bioinformatics tools.," 2018 (accessed 2018-07-13). <https://omictools.com/bioinformatics-trends>.

- [34] S. Pirhadi, J. Sunseri, and D. R. Koes, "Open source molecular modeling," *Journal of Molecular Graphics and Modelling*, vol. 69, pp. 127–143, 2016.
- [35] GitHub, "GitHub," 2008 (accessed 2018-07-12). <https://github.com>.
- [36] UCSF Chimera Team, "Chimera release notes," 2000 (accessed 2018-07-12). <http://plato.cgl.ucsf.edu/chimera/data/downloads/1.1602/docs/relnotes.html>.
- [37] M. Hashemi, "10 Myths of Enterprise Python," 2014 (accessed 2018-07-14). <https://www.paypal-engineering.com/2014/12/10/10-myths-of-enterprise-python/>.
- [38] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.
- [39] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: A LLVM-based python JIT compiler," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, p. 7, ACM, 2015.
- [40] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, "Cython: The best of both worlds," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 31–39, 2011.
- [41] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: a system for large-scale machine learning," in *OSDI*, vol. 16, pp. 265–283, 2016.
- [42] J. Bergstra, F. Bastien, O. Breuleux, P. Lamblin, R. Pascanu, O. Delalleau, G. Desjardins, D. Warde-Farley, I. Goodfellow, A. Bergeron, *et al.*, "Theano: Deep learning on gpus with python," in *NIPS 2011, Big Learning Workshop, Granada, Spain*, vol. 3, pp. 1–48, Citeseer, 2011.
- [43] A. Paszke, S. Gross, S. Chintala, and G. Chanan, "PyTorch," 2017.
- [44] K. Lipkowitz and J. Laane, "N.l. allinger – biography," *Journal of Molecular Structure*, vol. 556, no. 1-3, pp. xi–xiv, 2000.
- [45] C. F. Kettering, L. W. Shutts, and D. H. Andrews, "A representation of the dynamic properties of molecules by mechanical models," *Physical Review*, vol. 36, no. 3, p. 531, 1930.
- [46] E. Schrödinger, "Quantisierung als eigenwertproblem," *Annalen der Physik*, vol. 384, no. 6, pp. 489–527, 1926.
- [47] Ø. Burrau, "Berechnung des Energiewertes des Wasserstoff molekellons (H<sub>2</sub><sup>+</sup>) im Normalzustand," *Die Naturwissenschaften*, vol. 15, no. 1, pp. 16–17, 1927.
- [48] W. Heitler and F. London, "Wechselwirkung neutraler atome und homöopolare bindung nach der quantenmechanik," *Zeitschrift für Physik*, vol. 44, no. 6-7, pp. 455–472, 1927.
- [49] E. Teller, "About the hydrogen molecular ion," *Zeitschrift fuer Physik*, vol. 61, pp. 458–480, 1930.
- [50] H. C. Longuet-Higgins, "Robert sanderson mulliken, 7 june 1896 - 31 october 1986," *Biographical Memoirs of Fellows of the Royal Society*, vol. 35, pp. 327–354, Mar 1990.
- [51] N. Kemmer and R. Schlapp, "Max Born, 1882-1970," *Biographical Memoirs of Fellows of the Royal Society*, vol. 17, pp. 17–52, 1971.
- [52] H. A. Bethe, "J. robert oppenheimer, 1904-1967," *Biographical Memoirs of Fellows of the Royal Society*, vol. 14, pp. 390–416, 1968.
- [53] J. D. Dunitz, "Linus carl pauling, 28 february 1901 - 19 august 1994," *Biographical Memoirs of Fellows of the Royal Society*, vol. 42, pp. 317–338, 1996.
- [54] H. Hartmann and H. C. Longuet-Higgins, "Erich Hückel, 9 august 1896 - 16 february 1980," *Biographical Memoirs of Fellows of the Royal Society*, vol. 28, pp. 153–162, 1982.
- [55] C. G. Darwin, "Douglas rayner hartree, 1897-1958," *Biographical Memoirs of Fellows of the Royal Society*, vol. 4, pp. 102–116, 1958.
- [56] V. Fock, "Näherungsmethode zur lösung des quantenmechanischen mehrkörperproblems," *Zeitschrift für Physik*, vol. 61, no. 1-2, pp. 126–148, 1930.
- [57] Computer History Museum, "Timeline of Computer History," 2018 (accessed 2018-07-14). <http://www.computerhistory.org/timeline/1940/>.

- [58] B. J. D. and H. R. B., *The Development of Computational Chemistry in the United States*, pp. 1–63. Wiley-Blackwell, 2007.
- [59] Computational Chemistry List members, “100 years of computational chemistry?,” 2018 (accessed 2018-07-13). <http://www.ccl.net/cgi-bin/ccl/message-new?2018+04+19+002>.
- [60] B. J. Alder and T. E. Wainwright, “Studies in molecular dynamics. i. general method,” *The Journal of Chemical Physics*, vol. 31, no. 2, pp. 459–466, 1959.
- [61] J. Gibson, A. N. Goland, M. Milgram, and G. Vineyard, “Dynamics of radiation damage,” *Physical Review*, vol. 120, no. 4, p. 1229, 1960.
- [62] A. Rahman, “Correlations in the motion of atoms in liquid argon,” *Physical Review*, vol. 136, no. 2A, p. A405, 1964.
- [63] S. Lifson and A. Warshel, “Consistent force field for calculations of conformations, vibrational spectra, and enthalpies of cycloalkane and n-alkane molecules,” *The Journal of Chemical Physics*, vol. 49, no. 11, pp. 5116–5129, 1968.
- [64] A. Hagler, E. Huler, and S. Lifson, “Energy functions for peptides and proteins. i. derivation of a consistent force field including the hydrogen bond from amide crystals,” *Journal of the American Chemical Society*, vol. 96, no. 17, pp. 5319–5327, 1974.
- [65] S. Niketic and K. Rasmussen, “Lecture notes in chemistry,” in *The consistent force field: a documentation*, vol. 3, Springer Berlin, 1977.
- [66] N. L. Allinger and J. T. Sprague, “Conformational analysis. xc. calculation of the structures of hydrocarbons containing delocalized electronic systems by the molecular mechanics method,” *Journal of the American Chemical Society*, vol. 95, no. 12, pp. 3893–3907, 1973.
- [67] N. L. Allinger, “Conformational Analysis. 130. MM2. A Hydrocarbon Force Field Utilizing V1 and V2 Torsional Terms,” *Journal of the American Chemical Society*, vol. 99, no. 25, pp. 8127–8134, 1977.
- [68] F. A. Momany, R. F. McGuire, A. W. Burgess, and H. A. Scheraga, “Energy parameters in polypeptides. vii. geometric parameters, partial atomic charges, nonbonded interactions, hydrogen bond interactions, and intrinsic torsional potentials for the naturally occurring amino acids,” *The Journal of Physical Chemistry*, vol. 79, no. 22, pp. 2361–2381, 1975.
- [69] G. Nemethy, M. S. Pottle, and H. A. Scheraga, “Energy parameters in polypeptides. ix. updating of geometrical parameters, nonbonded interactions, and hydrogen bond interactions for the naturally occurring amino acids,” *The Journal of Physical Chemistry*, vol. 87, no. 11, pp. 1883–1887, 1983.
- [70] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, “CHARMM: A program for macromolecular energy, minimization, and dynamics calculations,” *Journal of Computational Chemistry*, vol. 4, no. 2, pp. 187–217, 1983.
- [71] W. F. van Gunsteren and H. J. Berendsen, “Groningen molecular simulation (GROMOS) library manual,” *Biomos, Groningen*, vol. 24, no. 682704, p. 13, 1987.
- [72] Fortune Magazine, “The next industrial revolution,” oct 1981. <http://backissues.com/issue/Fortune-October-05-1981>.
- [73] K. B. Lipkowitz and D. B. Boyd, eds., *Reviews in Computational Chemistry*. 1990.
- [74] P. E. M. Siegbahn and F. Himo, “Recent developments of the quantum chemical cluster approach for modeling enzyme reactions,” *JBIC Journal of Biological Inorganic Chemistry*, vol. 14, no. 5, pp. 643–651, 2009.
- [75] S. Kmiecik, D. Gront, M. Kolinski, L. Wieteska, A. E. Dawid, and A. Kolinski, “Coarse-grained protein models and their applications,” *Chemical Reviews*, vol. 116, no. 14, pp. 7898–7936, 2016.
- [76] H. I. Ingólfsson, C. A. Lopez, J. J. Uusitalo, D. H. de Jong, S. M. Gopal, X. Periole, and S. J. Marrink, “The power of coarse graining in biomolecular simulations,” *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 4, no. 3, pp. 225–248, 2013.
- [77] M. J. Boniecki, G. Lach, W. K. Dawson, K. Tomala, P. Lukasz, T. Soltysinski, K. M. Rother, and J. M. Bujnicki, “SimRNA: a coarse-grained method for RNA folding simulations and 3d structure prediction,” *Nucleic Acids Research*, vol. 44, no. 7, pp. e63–e63, 2015.

- [78] D. A. Potoyan, A. Savelyev, and G. A. Papoian, "Recent successes in coarse-grained modeling of DNA," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 3, no. 1, pp. 69–83, 2012.
- [79] R. Baron, D. Trzesniak, A. H. de Vries, A. Elsener, S. J. Marrink, and W. F. van Gunsteren, "Comparison of thermodynamic properties of coarse-grained and atomic-level simulation models," *ChemPhysChem*, vol. 8, no. 3, pp. 452–461, 2007.
- [80] M. F. Hagan and R. Zandi, "Recent advances in coarse-grained modeling of virus assembly," *Current Opinion in Virology*, vol. 18, pp. 36–43, 2016.
- [81] I. V. Pivkin and G. E. Karniadakis, "Accurate coarse-grained modeling of red blood cells," *Physical Review Letters*, vol. 101, no. 11, p. 118105, 2008.
- [82] P. Soderhjelm, G. A. Tribello, and M. Parrinello, "Locating binding poses in protein-ligand systems using reconnaissance metadynamics," *Proceedings of the National Academy of Sciences*, vol. 109, p. 5170–5175, Mar 2012.
- [83] M. De Vivo and A. Cavalli, "Recent advances in dynamic docking for drug discovery," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 7, no. 6, p. e1320, 2017.
- [84] G. Neudert and G. Klebe, "DSX: a knowledge-based scoring function for the assessment of protein–ligand complexes," *Journal of Chemical Information and Modeling*, vol. 51, no. 10, pp. 2731–2745, 2011.
- [85] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, and A. J. Olson, "Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function," *Journal of Computational Chemistry*, vol. 19, no. 14, pp. 1639–1662, 1998.
- [86] C. A. Baxter, C. W. Murray, D. E. Clark, D. R. Westhead, and M. D. Eldridge, "Flexible docking using tabu search and an empirical estimate of binding affinity," *Proteins: Structure, Function, and Bioinformatics*, vol. 33, no. 3, pp. 367–382, 1998.
- [87] O. Korb, T. Stutzle, and T. E. Exner, "Empirical scoring functions for advanced protein-ligand docking with PLANTS," *Journal of Chemical Information and Modeling*, vol. 49, no. 1, pp. 84–96, 2009.
- [88] C. M. Venkatachalam, X. Jiang, T. Oldfield, and M. Waldman, "Ligandfit: a novel method for the shape-directed rapid docking of ligands to protein active sites," *Journal of Molecular Graphics and Modelling*, vol. 21, no. 4, pp. 289–307, 2003.
- [89] H. A. Gabb, R. M. Jackson, and M. J. Sternberg, "Modelling protein docking using shape complementarity, electrostatics and biochemical information," *Journal of Molecular Biology*, vol. 272, no. 1, pp. 106–120, 1997.
- [90] B. K. Shoichet, I. D. Kuntz, and D. L. Bodian, "Molecular docking using shape descriptors," *Journal of Computational Chemistry*, vol. 13, no. 3, pp. 380–397, 1992.
- [91] M. A. Khamis, W. Gomaa, and W. F. Ahmed, "Machine learning in computational docking," *Artificial Intelligence in Medicine*, vol. 63, no. 3, pp. 135–152, 2015.
- [92] A. M. Virshup, J. Contreras-García, P. Wipf, W. Yang, and D. N. Beratan, "Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds," *Journal of the American Chemical Society*, vol. 135, no. 19, pp. 7296–7303, 2013.
- [93] T. I. Oprea and J. Gottfries, "Chemography: the art of navigating in chemical space," *Journal of Combinatorial Chemistry*, vol. 3, no. 2, pp. 157–166, 2001.
- [94] R. S. Bon and H. Waldmann, "Bioactivity-guided navigation of chemical space," *Accounts of Chemical Research*, vol. 43, no. 8, pp. 1103–1114, 2010.
- [95] J. Larsson, J. Gottfries, S. Muresan, and A. Backlund, "ChemGPS-NP: tuned for navigation in biologically relevant chemical space," *Journal of Natural Products*, vol. 70, no. 5, pp. 789–794, 2007.
- [96] R. A. Goodnow, C. E. Dumelin, and A. D. Keefe, "DNA-encoded chemistry: enabling the deeper sampling of chemical space," *Nature Reviews Drug Discovery*, vol. 16, no. 2, pp. 131–147, 2016.
- [97] J.-L. Reymond, "The chemical space project," *Accounts of Chemical Research*, vol. 48, no. 3, pp. 722–730, 2015.

- [98] A. Chari, D. Haselbach, J.-M. Kirves, J. Ohmer, E. Paknia, N. Fischer, O. Ganichkin, V. Möller, J. J. Frye, G. Petzold, M. Jarvis, M. Tietzel, C. Grimm, J.-M. Peters, B. A. Schulman, K. Tittmann, J. Markl, U. Fischer, and H. Stark, "ProteoPlex: stability optimization of macromolecular complexes by sparse-matrix screening of chemical space," *Nature Methods*, vol. 12, no. 9, pp. 859–865, 2015.
- [99] G. M. Maggiora and J. Bajorath, "Chemical space networks: a powerful new paradigm for the description of chemical space," *Journal of Computer-Aided Molecular Design*, vol. 28, no. 8, pp. 795–802, 2014.
- [100] J. J. Naveja and J. L. Medina-Franco, "ChemMaps: Towards an approach for visualizing the chemical space based on adaptive satellite compounds," *F1000Research*, vol. 6, p. 1134, 2017.
- [101] J. D. Durrant, R. E. Amaro, and J. A. McCammon, "AutoGrow: A novel algorithm for protein inhibitor design," *Chemical Biology and Drug Design*, vol. 73, no. 2, pp. 168–178, 2009.
- [102] J. L. Andersen, C. Flamm, D. Merkle, and P. F. Stadler, "Generic strategies for chemical space exploration," *International Journal of Computational Biology and Drug Design*, vol. 7, no. 2/3, p. 225, 2014.
- [103] P. Kumar, S. Henikoff, and P. C. Ng, "Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm," *Nature Protocols*, vol. 4, no. 7, pp. 1073–1081, 2009.
- [104] L. Quan, Q. Lv, and Y. Zhang, "Strum: structure-based prediction of protein stability changes upon single-point mutation," *Bioinformatics*, vol. 32, no. 19, pp. 2936–2946, 2016.
- [105] P. Fariselli, P. L. Martelli, C. Savojardo, and R. Casadio, "INPS: predicting the impact of non-synonymous variations on protein stability from sequence," *Bioinformatics*, vol. 31, no. 17, pp. 2816–2821, 2015.
- [106] Y. Dehouck, J. M. Kwasigroch, D. Gilis, and M. Rooman, "PoPMuSiC 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality," *BMC Bioinformatics*, vol. 12, no. 1, p. 151, 2011.
- [107] M. S. Kumar, K. A. Bava, M. M. Gromiha, P. Prabakaran, K. Kitajima, H. Uedaira, and A. Sarai, "ProTherm and ProNIT: thermodynamic databases for proteins and protein–nucleic acid interactions," *Nucleic Acids Research*, vol. 34, no. 9, pp. D204–D206, 2006.
- [108] J. Rodríguez-Guerra Pedregal, L. Alonso-Cotchico, G. Sciortino, A. Lledós, and J.-D. Maréchal, *Computational Studies of Artificial Metalloenzymes: From Methods and Models to Design and Optimization*, ch. 4, pp. 99–136. Wiley-Blackwell, 2018.
- [109] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *Journal of Chemical Information and Computer Sciences*, vol. 28, no. 1, pp. 31–36, 1988.
- [110] R. Dennington, T. Keith, J. Millam, K. Eppinnett, W. L. Hovell, and R. Gilliland, "Gaussview," 2009.
- [111] M. D. Hanwell, D. E. Curtis, D. C. Lonie, T. Vandermeersch, E. Zurek, and G. R. Hutchison, "Avogadro: an advanced semantic chemical editor, visualization, and analysis platform," *Journal of Cheminformatics*, vol. 4, no. 1, p. 17, 2012.
- [112] M. D. Winn, C. C. Ballard, K. D. Cowtan, E. J. Dodson, P. Emsley, P. R. Evans, R. M. Keegan, E. B. Krissinel, A. G. Leslie, A. McCoy, *et al.*, "Overview of the CCP4 suite and current developments," *Acta Crystallographica Section D*, vol. 67, no. 4, pp. 235–242, 2011.
- [113] P. D. Adams, R. W. Grosse-Kunstleve, L.-W. Hung, T. R. Ioerger, A. J. McCoy, N. W. Moriarty, R. J. Read, J. C. Sacchettini, N. K. Sauter, and T. C. Terwilliger, "PHENIX: building new software for automated crystallographic structure determination," *Acta Crystallographica Section D: Biological Crystallography*, vol. 58, no. 11, pp. 1948–1954, 2002.
- [114] C. Colovos and T. O. Yeates, "Verification of protein structures: Patterns of nonbonded atomic interactions," *Protein Science*, vol. 2, no. 9, pp. 1511–1519.
- [115] F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. F. Meyer Jr, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, "The Protein Data Bank: A computer-based archival file for macromolecular structures," *European Journal of Biochemistry*, vol. 80, no. 2, pp. 319–324, 1977.
- [116] J. M. Word, S. C. Lovell, J. S. Richardson, and D. C. Richardson, "Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation," *Journal of Molecular Biology*, vol. 285, no. 4, pp. 1735–1747, 1999.

- [117] M. Rostkowski, M. H. Olsson, C. R. Søndergaard, and J. H. Jensen, “Graphical analysis of pH-dependent properties of proteins predicted using PROPKA,” *BMC Structural Biology*, vol. 11, no. 1, p. 6, 2011.
- [118] UCSF Chimera, “Usage of sym command,” 2010 (accessed 2018-07-12).  
<https://www.cgl.ucsf.edu/chimera/current/docs/UsersGuide/midas/sym.html>.
- [119] S. Bietz and M. Rarey, “SIENA: efficient compilation of selective protein binding site ensembles,” *Journal of Chemical Information and Modeling*, vol. 56, no. 1, pp. 248–259, 2016.
- [120] UCSF Chimera, “Usage of addh command,” (accessed 2018-07-12).  
<https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/midas/addh.html>.
- [121] P. Benkert, S. C. Tosatto, and D. Schomburg, “QMEAN: A comprehensive scoring function for model quality assessment,” *Proteins: Structure, Function, and Bioinformatics*, vol. 71, no. 1, pp. 261–277, 2008.
- [122] Z. Wang, J. Eickholt, and J. Cheng, “APOLLO: a quality assessment service for single and multiple protein models,” *Bioinformatics*, vol. 27, no. 12, pp. 1715–1716, 2011.
- [123] C. G. Broyden, “The convergence of a class of double-rank minimization algorithms: 2. the new algorithm,” *IMA Journal of Applied Mathematics*, vol. 6, no. 3, pp. 222–231, 1970.
- [124] R. Fletcher, “A new approach to variable metric algorithms,” *The Computer Journal*, vol. 13, no. 3, pp. 317–322, 1970.
- [125] D. Goldfarb, “A family of variable-metric methods derived by variational means,” *Mathematics of Computation*, vol. 24, no. 109, pp. 23–26, 1970.
- [126] D. F. Shanno, “Conditioning of quasi-newton methods for function minimization,” *Mathematics of Computation*, vol. 24, no. 111, pp. 647–656, 1970.
- [127] D. C. Liu and J. Nocedal, “On the limited memory bfgs method for large scale optimization,” *Mathematical Programming*, vol. 45, no. 1-3, pp. 503–528, 1989.
- [128] S. Kirkpatrick, C. Gelatt, and M. Vecchi, “Optimization by simulated annealing,” *Readings in Computer Vision*, vol. 220, no. 4598, pp. 606–615, 1987.
- [129] A. Colnari, M. Dorigo, and V. Maniezzo, “Distributed optimization by ant colonies,” in *Toward a practice of autonomous systems: proceedings of the First European Conference on Artificial Life*, p. 134, Mit Press, 1992.
- [130] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Micro Machine and Human Science, 1995. MHS’95., Proceedings of the Sixth International Symposium on*, pp. 39–43, IEEE, 1995.
- [131] K. Sørensen, “Metaheuristics-the metaphor exposed,” *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2013.
- [132] A. Brownlee and J. R. Woodward, “Why we fell out of love with algorithms inspired by Nature,” 2015 (accessed 2018-07-14). <http://theconversation.com/why-we-fell-out-of-love-with-algorithms-inspired-by-nature-42718>.
- [133] J. Swan, S. Adriaensen, M. Bishr, E. K. Burke, J. A. Clark, P. De Causmaecker, J. Durillo, K. Hammond, E. Hart, C. G. Johnson, *et al.*, “A research agenda for metaheuristic standardization,” in *Proceedings of the XI metaheuristics international conference*, 2015.
- [134] F. Glover and K. Sorensen, “Metaheuristics,” *Scholarpedia*, vol. 10, no. 4, p. 6532, 2015.
- [135] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “Statistical and machine learning forecasting methods: Concerns and ways forward,” *PLoS One*, vol. 13, no. 3, p. e0194889, 2018.
- [136] I. Das and J. E. Dennis, “A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems,” *Structural Optimization*, vol. 14, no. 1, pp. 63–69, 1997.
- [137] K. Deb, “Multi-objective genetic algorithms: Problem difficulties and construction of test problems,” *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.
- [138] C. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer Science & Business Media, 2007.
- [139] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.



- [140] R. L. Dunbrack Jr and M. Karplus, "Backbone-dependent rotamer library for proteins application to side-chain prediction," *Journal of Molecular Biology*, vol. 230, no. 2, pp. 543–574, 1993.
- [141] A. D. Scouras and V. Daggett, "The Dymeomics rotamer library: amino acid side chain conformations and dynamics from comprehensive molecular dynamics simulations in water," *Protein Science*, vol. 20, no. 2, pp. 341–352, 2011.
- [142] A. Bakan, L. M. Meireles, and I. Bahar, "ProDy: protein dynamics inferred from theory and experiments," *Bioinformatics*, vol. 27, no. 11, pp. 1575–1577, 2011.
- [143] R. T. McGibbon, K. A. Beauchamp, M. P. Harrigan, C. Klein, J. M. Swails, C. X. Hernández, C. R. Schwantes, L.-P. Wang, T. J. Lane, and V. S. Pande, "MDTraj: a modern open library for the analysis of molecular dynamics trajectories," *Biophysical Journal*, vol. 109, no. 8, pp. 1528–1532, 2015.
- [144] A. Krammer, P. D. Kirchhoff, X. Jiang, C. Venkatachalam, and M. Waldman, "LigScore: a novel scoring function for predicting binding affinities," *Journal of Molecular Graphics and Modelling*, vol. 23, no. 5, pp. 395–407, 2005.
- [145] D. Russel, K. Lasker, B. Webb, J. Velázquez-Muriel, E. Tjioe, D. Schneidman-Duhovny, B. Peterson, and A. Sali, "Putting the pieces together: integrative modeling platform software for structure determination of macromolecular assemblies," *PLoS Biology*, vol. 10, no. 1, p. e1001244, 2012.
- [146] G. Jones, P. Willett, R. C. Glen, A. R. Leach, and R. Taylor, "Development and validation of a genetic algorithm for flexible docking," *Journal of Molecular Biology*, vol. 267, no. 3, pp. 727–748, 1997.
- [147] M. Valiev, E. J. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, H. J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T. L. Windus, *et al.*, "NWChem: a comprehensive and scalable open-source solution for large scale molecular simulations," *Computer Physics Communications*, vol. 181, no. 9, pp. 1477–1489, 2010.
- [148] J. Rodríguez-Guerra Pedregal, G. Sciortino, E. Garriba, and J.-D. Maréchal, "Simple coordination geometry descriptors allow to accurately predict metal binding sites in proteins," (*submitted*), 2018.
- [149] F.-A. Fortin, F.-M. D. Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, no. Jul, pp. 2171–2175, 2012.
- [150] J. Rodríguez-Guerra Pedregal and J.-D. Maréchal, "Tangram: Molecular modeling tools for UCSF Chimera," 2018. <https://github.com/insilichem/tangram>.
- [151] J. Rodríguez-Guerra Pedregal, P. Gómez-Orellana, and J.-D. Maréchal, "ESIgen: Electronic supporting information generator for computational chemistry publications," *Journal of Chemical Information and Modeling*, vol. 58, no. 3, pp. 561–564, 2018.
- [152] J. Rodríguez-Guerra Pedregal, G. Sciortino, J. Guasp, M. Municoy, and J.-D. Maréchal, "GaudiMM: A modular multi-objective platform for molecular modeling," *Journal of Computational Chemistry*, vol. 38, no. 24, pp. 2118–2126, 2017.
- [153] J. Rodríguez-Guerra Pedregal and J.-D. Maréchal, "PyChimera: use UCSF Chimera modules in any Python 2.7 project," *Bioinformatics*, vol. 34, no. 10, pp. 1784–1785, 2018.
- [154] J. Rodríguez-Guerra Pedregal, L. Alonso-Cotchico, L. Velasco-Carneros, and J.-D. Maréchal, "OMMProtocol: A command line application to launch molecular dynamics simulations with OpenMM," (*submitted*), 2018. <https://github.com/insilichem/ommprotocol>.
- [155] J. Rodríguez-Guerra Pedregal, I. Funes-Ardoiz, G. Sciortino, J.-E. Sánchez-Aparicio, G. Ujaque, A. Lledós, and F. Maréchal, Jean-Didier Maseras, "GARLEEK: Adding an Extra Flavor to ONIOM," *Journal of Computational Chemistry*, 2018. <https://github.com/insilichem/garleek>.
- [156] J. Rodríguez-Guerra Pedregal, I. Funes-Ardoiz, and F. Maseras, "EasyMECP: Quick setup of MECP calculations with Gaussian," 2018. <https://github.com/jaimergp/easymecep>.
- [157] J. N. Harvey, M. Aschi, H. Schwarz, and W. Koch, "The singlet and triplet states of phenyl cation. A hybrid approach for locating minimum energy crossing points between non-interacting potential energy surfaces," *Theoretical Chemistry Accounts*, vol. 99, no. 2, pp. 95–99, 1998.
- [158] E. S. Raymond, *The art of Unix programming*. Addison-Wesley Professional, 2003.

- [159] F. Duarte, P. Bauer, A. Barrozo, B. A. Amrein, M. Purg, J. Åqvist, and S. C. L. Kamerlin, "Force field independent metal parameters using a nonbonded dummy model," *Journal of Physical Chemistry B*, vol. 118, no. 16, pp. 4351–4362, 2014.
- [160] J. Contreras-García, E. R. Johnson, S. Keinan, R. Chaudret, J.-P. Piquemal, D. N. Beratan, and W. Yang, "NCIPLOT: a program for plotting noncovalent interaction regions," *Journal of Chemical Theory and Computation*, vol. 7, no. 3, pp. 625–632, 2011.
- [161] S. Salentin, S. Schreiber, V. J. Haupt, M. F. Adasme, and M. Schroeder, "PLIP: fully automated protein–ligand interaction profiler," *Nucleic Acids Research*, vol. 43, no. W1, pp. W443–W447, 2015.
- [162] A. Varki, R. D. Cummings, M. Aebi, N. H. Packer, P. H. Seeberger, J. D. Esko, P. Stanley, G. Hart, A. Darvill, T. Kinoshita, *et al.*, "Symbol nomenclature for graphical representations of glycans," *Glycobiology*, vol. 25, no. 12, pp. 1323–1324, 2015.
- [163] D. F. Thieker, J. A. Hadden, K. Schulten, and R. J. Woods, "3D implementation of the symbol nomenclature for graphical representation of glycans," *Glycobiology*, vol. 26, no. 8, pp. 786–787, 2016.
- [164] G. Landrum *et al.*, "RDKit: Open-source cheminformatics," 2006.
- [165] K. L. Schuchardt, B. T. Didier, T. Elsethagen, L. Sun, V. Gurumoorthi, J. Chase, J. Li, and T. L. Windus, "Basis Set Exchange: A Community Database for Computational Sciences," *Journal of Chemical Information and Modeling*, vol. 47, no. 3, pp. 1045–1052, 2007.
- [166] Eastman, Peter, "PDBFixer," 2013 (accessed 2018-07-13). <https://github.com/pandegroup/pdbfixer>.
- [167] J. Wang, W. Wang, P. A. Kollman, and D. A. Case, "Antechamber: an accessory software package for molecular mechanical calculations," *Journal of the American Chemical Society*, vol. 222, p. U403, 2001.
- [168] P. Li and K. M. Merz, "Mcpb.py: A python based metal center parameter builder," *Journal of Chemical Information and Modeling*, vol. 56, no. 4, pp. 599–604, 2016. PMID: 26913476.
- [169] V. Muñoz Robles, *Development and Applications of Molecular Modelling Techniques for the Design and Optimization of Artificial Metalloenzymes*. PhD thesis, Universitat Autònoma de Barcelona, 2014.
- [170] K. Hinsen, "The molecular modeling toolkit: a new approach to molecular simulations," *Journal of Computational Chemistry*, vol. 21, no. 2, pp. 79–85, 2000.
- [171] Continuum Analytics, "Conda," 2017 (accessed 2018-07-13). <https://conda.io/docs/>.
- [172] F. Pérez and B. E. Granger, "IPython: a system for interactive scientific computing," *Computing in Science & Engineering*, vol. 9, no. 3, 2007.
- [173] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. B. Hamrick, J. Grout, S. Corlay, *et al.*, "Jupyter notebooks—a publishing format for reproducible computational workflows," in *ELPUB*, pp. 87–90, 2016.
- [174] B. Hess, C. Kutzner, D. Van Der Spoel, and E. Lindahl, "GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation," *Journal of Chemical Theory and Computation*, vol. 4, no. 3, pp. 435–447, 2008.
- [175] J. Swails, "ParmEd," 2015 (accessed 2018-07-14). <https://github.com/ParmEd/ParmEd>.
- [176] J. Chodera, "OpenMolTools," 2015 (accessed 2018-07-14). <https://github.com/choderalab/openmoltools>.
- [177] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," *Journal of the American Chemical Society*, vol. 117, no. 19, pp. 5179–5197, 1995.
- [178] A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard, and W. M. Skiff, "UFF, a Full Periodic Table Force Field for Molecular Mechanics and Molecular Dynamics Simulations," *Journal of the American Chemical Society*, vol. 114, no. 25, pp. 10024–10035, 1992.
- [179] S. L. Mayo, B. D. Olafson, and W. A. Goddard, "DREIDING: a generic force field for molecular simulations," *The Journal of Physical Chemistry*, vol. 94, no. 26, pp. 8897–8909, 1990.



- [180] V. Hornak, R. Abel, A. Okur, B. Strockbine, A. Roitberg, and C. Simmerling, "Comparison of multiple amber force fields and development of improved protein backbone parameters," *Proteins: Structure, Function, and Bioinformatics*, vol. 65, no. 3, pp. 712–725, 2006.
- [181] A. D. MacKerell Jr, D. Bashford, M. Bellott, R. L. Dunbrack Jr, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, *et al.*, "All-atom empirical potential for molecular modeling and dynamics studies of proteins," *The journal of physical chemistry B*, vol. 102, no. 18, pp. 3586–3616, 1998.
- [182] P. Ren and J. W. Ponder, "Consistent treatment of inter-and intramolecular polarization in molecular mechanics calculations," *Journal of Computational Chemistry*, vol. 23, no. 16, pp. 1497–1506, 2002.
- [183] T. A. Halgren, "Merck molecular force field. i. basis, form, scope, parameterization, and performance of MMFF94s," *Journal of Computational Chemistry*, vol. 17, no. 5-6, pp. 490–519, 1996.
- [184] N. L. Allinger, Y. H. Yuh, and J. H. Lii, "Molecular mechanics. the MM3 force field for hydrocarbons. 1," *Journal of the American Chemical Society*, vol. 111, no. 23, pp. 8551–8566, 1989.
- [185] A. S. Rose, A. R. Bradley, Y. Valasatava, J. M. Duarte, A. Prlic, and P. W. Rose, "Ngl viewer: Web-based molecular graphics for large complexes," *Bioinformatics*, vol. 1, p. 4, 2018.
- [186] CERN, "Zenodo," 2013 (accessed 2018-07-13). <https://zenodo.org>.
- [187] Figshare LLP, "FigShare," 2011 (accessed 2018-07-13). <https://figshare.com>.
- [188] N. M. O'Boyle, A. L. Tenderholt, and K. M. Langner, "Cclib: a library for package-independent computational chemistry algorithms," *Journal of Computational Chemistry*, vol. 29, no. 5, pp. 839–845, 2008.
- [189] F. Neese, "The ORCA program system," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 2, no. 1, pp. 73–78, 2012.
- [190] T. D. Goddard, C. C. Huang, E. C. Meng, E. F. Pettersen, G. S. Couch, J. H. Morris, and T. E. Ferrin, "Ucsf chimeraX: Meeting modern challenges in visualization and analysis," *Protein Science*, vol. 27, no. 1, pp. 14–25, 2018.
- [191] J. Snyder and D. H. Ess, "Development and application of minimum energy crossing point software suite for organometallic reactions," *Journal of Undergraduate Research*, 2017.
- [192] M. Radoń, "MECPy 0.9," 2015 (accessed 2018-07-13). <http://www2.chemia.uj.edu.pl/~mradon/mecpy/>.
- [193] T. Lu, "sobMECP program," 2015 (accessed 2018-07-13). <http://sobereva.com/286>.
- [194] N. S. Pagadala, K. Syed, and J. Tuszynski, "Software for molecular docking: a review," *Biophysical reviews*, vol. 9, no. 2, pp. 91–102, 2017.
- [195] S. Zirah, S. Kozin, A. Mazur, A. Blond, M. Cheminant, I. Segalas-Milazzo, P. Debey, and S. Rebuffat, "Zinc-binding domain of Alzheimer's disease amyloid beta-peptide complexed with a zinc (II) cation," May 2005.
- [196] S. Zirah, S. Kozin, A. Mazur, A. Blond, M. Cheminant, I. Segalas-Milazzo, P. Debey, and S. Rebuffat, "Zinc-binding domain of alzheimer's disease amyloid beta-peptide in water solution at pH 6.5," May 2005.
- [197] D. Rehder, *Bioinorganic chemistry*. Oxford University Press, 2014.
- [198] J. I. Mujika, J. R.-G. Pedregal, X. Lopez, J. M. Ugalde, L. Rodríguez-Santiago, M. Sodupe, and J.-D. Maréchal, "Elucidating the 3D structures of Al (iii)–A $\beta$  complexes: a template free strategy based on the pre-organization hypothesis," *Chemical Science*, vol. 8, no. 7, pp. 5041–5049, 2017.
- [199] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.
- [200] P. C. Database, "Enterobactin cid=34231." <https://pubchem.ncbi.nlm.nih.gov/compound/34231>.
- [201] N. Li and L. Gu, "Crystal structure of holo FepB," 2013.
- [202] I. Nierengarten, M. González-Cuesta, J. Rodríguez-Guerra Pedregal, U. Hahn, S. Romero-Téllez, J.-D. Maréchal, L. Masgrau, J. M. García Fernández, J.-F. Nierengarten, and C. Ortiz Mellet, "Pillar[5]arene glyco(mimetic)rotaxanes for the functional interrogation of multivalency responsive glycosidases," (*submitted*), 2018.

- [203] D. Tielker, S. Hacker, R. Loris, M. Strathmann, J. Wingender, S. Wilhelm, F. Rosenau, and K.-E. Jaeger, "Pseudomonas aeruginosa lectin lecb is located in the outer membrane and is involved in biofilm formation," *Microbiology*, vol. 151, no. 5, pp. 1313–1323, 2005.
- [204] M. W. Turner, "Mannose-binding lectin: the pluripotent molecule of the innate immune system," *Immunology Today*, vol. 17, no. 11, pp. 532–540, 1996.
- [205] H. Adwan, H. Bayer, A. Pervaiz, M. Sagini, and M. R. Berger, "Riproximin is a recently discovered type II ribosome inactivating protein with potential for treating cancer," *Biotechnology Advances*, vol. 32, no. 6, pp. 1077–1090, 2014.
- [206] P. Compain, C. Decroocq, J. Iehl, M. Holler, D. Hazelard, T. M. Barragán, C. O. Mellet, and J.-F. Nierengarten, "Glycosidase inhibition with fullerene iminosugar balls: a dramatic multivalent effect," *Angewandte Chemie International Edition*, vol. 49, no. 33, pp. 5753–5756, 2010.
- [207] M. Aguilar, T. Gloster, J. Turkenburg, M. Garcia-Moreno, C. Ortiz Mellet, G. Davies, and J. Garcia Fernandez, "Structure of family 1 beta-glucosidase from *thermotoga maritima* in complex with 3-imino-2-oxa-(+)-castanospermine," Apr 2009.
- [208] A. Correia, "European Multiscale Simulation for the Computational Era," Tech. Rep. April, Phantoms Foundation, 2012.
- [209] European Committee for Standardization, "CEN/WS MODA - Materials modelling - terminology, classification and metadata," 2017.  
<https://www.cen.eu/news/workshops/Pages/WS-2017-012.aspx>.
- [210] G. Goldbeck, "The economic impact of molecular modelling," pp. 0–45, 2012.
- [211] G. Goldbeck and C. Court, "The Economic Impact of Materials Modelling," no. January, p. 35, 2016.
- [212] S. John, C. Road, and U. Kingdom, "The scientific software industry: a general overview," no. January, pp. 1–16, 2017.
- [213] T. J. Kemp, "Epsrc / rsc joint report 'the economic benefits of chemistry research for the uk'," *Science Progress*, vol. 94, no. 2, pp. 220–231, 2011.
- [214] P. Warry, "Increasing the Economic Impact of the Research Councils.," Tech. Rep. January, 2007.
- [215] A. S. Louie, M. S. Brown, and A. Kim, "Measuring the Return on Modeling and Simulation Tools in Pharmaceutical Development (White paper N° HI204892, Health Industry Insights)," 2007 (accessed 2018-07-16). <http://accelrys.com/resource-center/white-papers/roi-mspharma.php>.
- [216] E. C. Joseph, S. Conway, C. Ingle, G. Cattaneo, N. Martinez, and C. Meunier, "A Strategic Agenda for European Leadership in Supercomputing : HPC 2020 — IDC Final Report of the HPC Study for the DG Information Society of the European Commission," no. September, 2010.
- [217] T. H. Dunning, *Impact of Advances in Computing and Communications Technologies on Chemical Science and Technology: Report of a Workshop*. 1999.
- [218] D. E. Shaw, M. M. Deneroff, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, C. Young, B. Batson, K. J. Bowers, J. C. Chao, *et al.*, "Anton, a special-purpose machine for molecular dynamics simulation," *Communications of the ACM*, vol. 51, no. 7, pp. 91–97, 2008.
- [219] T. J. Lane, D. Shukla, K. A. Beauchamp, and V. S. Pande, "To milliseconds and beyond: Challenges in the simulation of protein folding," *Current Opinion in Structural Biology*, vol. 23, no. 1, pp. 58–65, 2013.
- [220] L. Genovese, B. Videau, M. Ospici, T. Deutsch, S. Goedecker, and J.-F. Méhaut, "Daubechies wavelets for high performance electronic structure calculations: The BigDFT project," *Comptes Rendus Mécanique*, vol. 339, no. 2-3, pp. 149–164, 2011.
- [221] N. Luehr, I. S. Ufimtsev, and T. J. Martínez, "Dynamic precision for electron repulsion integral evaluation on graphical processing units (GPUs)," *Journal of Chemical Theory and Computation*, vol. 7, no. 4, pp. 949–954, 2011.
- [222] A. Warshel, "Multiscale modeling of biological functions: From enzymes to molecular machines (nobel lecture)," *Angewandte Chemie - International Edition*, vol. 53, no. 38, pp. 10020–10031, 2014.

- [223] J. Lee, P. L. Freddolino, and Y. Zhang, *From Protein Structure to Function with Bioinformatics*. 2017.
- [224] A. Cherkasov, E. N. Muratov, D. Fourches, A. Varnek, I. I. Baskin, M. Cronin, J. Dearden, P. Gramatica, Y. C. Martin, R. Todeschini, V. Consonni, V. E. Kuz'Min, R. Cramer, R. Benigni, C. Yang, J. Rathman, L. Terfloth, J. Gasteiger, A. Richard, and A. Tropsha, "QSAR modeling: Where have you been? Where are you going to?," *Journal of Medicinal Chemistry*, vol. 57, no. 12, pp. 4977–5010, 2014.
- [225] K. Paliwal, J. Lyons, and R. Heffernan, "A short review of deep learning neural networks in protein structure prediction problems," *Advanced Techniques in Biology and Medicine*, vol. 3, no. 3, p. 1000139, 2015.
- [226] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure-activity relationships," *Journal of Chemical Information and Modeling*, vol. 55, no. 2, pp. 263–274, 2015.
- [227] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nature Communications*, vol. 8, pp. 6–13, 2017.
- [228] G. B. Goh, N. O. Hodas, and A. Vishnu, "Deep learning for computational chemistry," *Journal of Computational Chemistry*, vol. 38, no. 16, pp. 1291–1307, 2017.
- [229] G. B. Goh, C. Siegel, A. Vishnu, N. O. Hodas, and N. Baker, "Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models," *arXiv*, pp. 1–38, 2017.
- [230] A. Koutsoukas, K. J. Monaghan, X. Li, and J. Huan, "Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data," *Journal of Cheminformatics*, vol. 9, no. 1, pp. 1–13, 2017.
- [231] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter, "DeepTox: Toxicity Prediction using Deep Learning," *Frontiers in Environmental Science*, vol. 3, no. February, 2016.
- [232] F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley, and O. A. von Lilienfeld, "Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error," *Journal of Chemical Theory and Computation*, vol. 13, no. 11, pp. 5255–5264, 2017.
- [233] B. Booth, "Four Decades Of Hacking Biotech And Yet Biology Still Consumes Everything," 2017 (accessed 2018-07-17). <https://www.forbes.com/sites/brucebooth/2017/04/26/four-decades-of-hacking-biotech-and-yet-biology-still-consumes-everything/#17da84a3779e>.
- [234] M. Benhenda, "Outsourcing AI For Drug Discovery: Independent Expertise Is Key To Avoid Overhyped Claims," 2017 (accessed 2018-07-17). <https://www.biopharmatrend.com/post/49-research-in-ai-for-drug-discovery-is-overhyped-and-what-to-do-about-it/>.
- [235] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, *et al.*, "Towards quantum chemistry on a quantum computer," *Nature chemistry*, vol. 2, no. 2, p. 106, 2010.
- [236] W. Zeng, B. Johnson, R. Smith, N. Rubin, M. Reagor, C. Ryan, and C. Rigetti, "First quantum computers need smart software," *Nature*, vol. 549, pp. 149–151, 2017.
- [237] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, "Elucidating reaction mechanisms on quantum computers," *Proceedings of the National Academy of Sciences*, vol. 114, no. 29, pp. 7555–7560, 2017.
- [238] A. Hellweg and F. Eckert, "Brick by brick computation of the gibbs free energy of reaction in solution using quantum chemistry and cosmo-rs," *AICHE Journal*, vol. 63, no. 9, pp. 3944–3954, 2017.
- [239] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, no. 7671, p. 242, 2017.
- [240] S. Sim, J. Romero, P. D. Johnson, and A. Aspuru-Guzik, "Quantum computer simulates excited states of molecule," *Physics*, vol. 11, p. 14, 2018.
- [241] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, "Cloud quantum computing of an atomic nucleus," *Physical Review Letters*, vol. 120, no. 21, p. 210501, 2018.

- [242] B. R. Brooks, C. L. B. III, J. A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caffisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. I. Hodoscek, and M. Karplus, "CHARMM: The Biomolecular Simulation Program B.," *Journal of Computational Chemistry*, vol. 30, no. 10, pp. 1545–1614, 2009.
- [243] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case, "Development and testing of a general amber force field," *Journal of Computational Chemistry*, vol. 25, no. 9, pp. 1157–1174, 2004.
- [244] M. Clark, R. D. Cramer, and N. Van Opdenbosch, "Validation of the general purpose Tripos 5.2 force field," *Journal of Computational Chemistry*, vol. 10, no. 8, pp. 982–1012, 1989.
- [245] T. A. Halgren, "Merck Molecular Force Field.," *Journal of Computational Chemistry*, vol. 17, no. 5-6, pp. 490–519, 1996.
- [246] J. M. Seminario, "Calculation of intramolecular force fields from second-derivative tensors," *International Journal of Quantum Chemistry*, vol. 60, no. 7, pp. 1271–1277, 1996.
- [247] P. Li and K. M. Merz, "MCPB.py: A Python Based Metal Center Parameter Builder," *Journal of Chemical Information and Modeling*, vol. 56, no. 4, pp. 599–604, 2016.
- [248] S. Zheng, Q. Tang, J. He, S. Du, S. Xu, C. Wang, Y. Xu, and F. Lin, "VFFDT: A New Software for Preparing AMBER Force Field Parameters for Metal-Containing Molecular Systems," *Journal of Chemical Information and Modeling*, vol. 56, no. 4, pp. 811–818, 2016.
- [249] F. Fracchia, G. Del Frate, G. Mancini, W. Rocchia, and V. Barone, "Force Field Parametrization of Metal Ions from Statistical Learning Techniques," *Journal of Chemical Theory and Computation*, vol. 14, no. 1, pp. 255–273, 2018.
- [250] Y. Li, H. Li, F. C. Pickard, B. Narayanan, F. G. Sen, M. K. Chan, S. K. Sankaranarayanan, B. R. Brooks, and B. Roux, "Machine Learning Force Field Parameters from Ab Initio Data," *Journal of Chemical Theory and Computation*, vol. 13, no. 9, pp. 4492–4503, 2017.
- [251] S. K. Burger, M. Lacasse, T. Verstraelen, J. Drewry, P. Gunning, and P. W. Ayers, "Automated Parametrization of AMBER Force Field Terms from Vibrational Analysis with a Focus on Functionalizing Dinuclear Zinc(II) Scaffolds," *Journal of Chemical Theory and Computation*, vol. 8, no. 2, pp. 554–562, 2012.
- [252] A. E. A. Allen, M. C. Payne, and D. J. Cole, "Harmonic Force Constants for Molecular Mechanics Force Fields via Hessian Matrix Projection," *Journal of Chemical Theory and Computation*, vol. 14, no. 1, pp. 274–281, 2018.
- [253] R. Huey, G. M. Morris, A. J. Olson, and D. S. Goodsell, "A semiempirical free energy force field with charge-based desolvation," *Journal of Computational Chemistry*, vol. 28, no. 6, pp. 1145–1152, 2007.
- [254] S. Bureekaew, S. Amirjalayer, M. Tafipolsky, C. Spickermann, T. K. Roy, and R. Schmid, "MOF-FF - a flexible first-principles derived force field for metal-organic frameworks," *Physica Status Solidi (B)*, vol. 250, no. 6, pp. 1128–1141, 2013.
- [255] M. A. Addicoat, N. Vankova, I. F. Akter, and T. Heine, "Extension of the universal force field to metal–organic frameworks," *Journal of Chemical Theory and Computation*, vol. 10, no. 2, pp. 880–891, 2014.
- [256] C. R. Landis, D. M. Root, and T. Cleveland, "Molecular mechanics force fields for modeling inorganic and organometallic compounds," in *Reviews in Computational Chemistry*, pp. 73–148, John Wiley & Sons, Inc., 2007.
- [257] S. Shi, L. Yan, Y. Yang, J. Fisher-Shaulsky, and T. Thacher, "An extensible and systematic force field, ESFF, for molecular modeling of organic, inorganic, and organometallic systems," *Journal of Computational Chemistry*, vol. 24, no. 9, pp. 1059–1076, 2003.
- [258] A. K. Rappe, K. S. Colwell, and C. J. Casewit, "Application of a universal force field to metal complexes," *Inorganic Chemistry*, vol. 32, no. 16, pp. 3438–3450, 1993.
- [259] K. D. Nielson, A. C. T. van Duin, J. Oxgaard, W.-Q. Deng, and W. A. Goddard, "Development of the ReaxFF reactive force field for describing transition metal catalyzed reactions, with application to the initial stages of the catalytic formation of carbon nanotubes," *The Journal of Physical Chemistry A*, vol. 109, no. 3, pp. 493–499, 2005.

- [260] J. Åqvist and A. Warshel, "Free Energy Relationships in Metalloenzyme-Catalyzed Reactions. Calculations of the Effects of Metal Ion Substitutions in Staphylococcal Nuclease," *Journal of the American Chemical Society*, vol. 112, no. 8, pp. 2860–2868, 1990.
- [261] L. Shao-Yong, H. Zhi-Min, H. Wen-Kang, L. Xin-Yi, C. Ying-Yi, S. Ting, and Z. Jian, "How calcium inhibits the magnesium-dependent kinase gsk3 $\beta$ : A molecular simulation study," *Proteins: Structure, Function, and Bioinformatics*, vol. 81, no. 5, pp. 740–753, 2012.
- [262] P. Oelschlaeger, M. Klahn, W. A. Beard, S. H. Wilson, and A. Warshel, "Magnesium-cationic dummy atom molecules enhance representation of DNA polymerase  $\beta$  in molecular dynamics simulations: Improved accuracy in studies of structural features and mutational effects," *Journal of Molecular Biology*, vol. 366, no. 2, pp. 687–701, 2007.
- [263] A. Saxena and D. Sept, "Multisite ion models that improve coordination and free energy calculations in molecular dynamics simulations," *Journal of Chemical Theory and Computation*, vol. 9, no. 8, pp. 3538–3542, 2013.
- [264] A. Saxena and A. E. García, "Multisite ion model in concentrated solutions of divalent cations (MgCl<sub>2</sub> and CaCl<sub>2</sub>): Osmotic pressure calculations," *The Journal of Physical Chemistry B*, vol. 119, no. 1, pp. 219–227, 2015.
- [265] Q. Liao, S. C. L. Kamerlin, and B. Strodel, "Development and application of a nonbonded cu<sup>2+</sup> model that includes the jahn–teller effect," *The Journal of Physical Chemistry Letters*, vol. 6, no. 13, pp. 2657–2662, 2015.
- [266] Y.-P. Pang, "Novel zinc protein molecular dynamics simulations: Steps toward antiangiogenesis for cancer treatment," *Journal of Molecular Modeling*, vol. 5, no. 10, pp. 196–202, 1999.
- [267] P. Li and K. M. Merz, "Metal Ion Modeling Using Classical Mechanics," *Chemical Reviews*, vol. 117, no. 3, pp. 1564–1686, 2017.
- [268] B. Kramer, M. Rarey, and T. Lengauer, "Evaluation of the FLEXX incremental construction algorithm for protein–ligand docking," *Proteins: Structure, Function, and Bioinformatics*, vol. 37, no. 2, pp. 228–241, 1999.
- [269] M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray, and R. D. Taylor, "Improved protein–ligand docking using GOLD," *Proteins: Structure, Function, and Bioinformatics*, vol. 52, no. 4, pp. 609–623, 2003.
- [270] A. Šali and T. L. Blundell, "Comparative protein modelling by satisfaction of spatial restraints," *Journal of Molecular Biology*, vol. 234, no. 3, pp. 779–815, 1993.
- [271] B. A. Johns, J. G. Weatherhead, S. H. Allen, J. B. Thompson, E. P. Garvey, S. A. Foster, J. L. Jeffrey, and W. H. Miller, "The use of oxadiazole and triazole substituted naphthyridines as HIV-1 integrase inhibitors. Part I: Establishing the pharmacophore," *Bioorganic and Medicinal Chemistry Letters*, vol. 19, no. 6, pp. 1802–1806, 2009.
- [272] T. Kawasuji, B. A. Johns, H. Yoshida, T. Taishi, Y. Taoda, H. Murai, R. Kiyama, M. Fuji, T. Yoshinaga, T. Seki, M. Kobayashi, A. Sato, and T. Fujiwara, "Carbamoyl pyridone HIV-1 integrase inhibitors. 1. Molecular design and establishment of an advanced two-metal binding pharmacophore," *Journal of Medicinal Chemistry*, vol. 55, no. 20, pp. 8735–8744, 2012.
- [273] M. Carcelli, D. Rogolino, A. Bacchi, G. Rispoli, E. Fiscaro, C. Compari, M. Sechi, A. Stevaert, and L. Naesens, "Metal-chelating 2-hydroxyphenyl amide pharmacophore for inhibition of influenza virus endonuclease," *Molecular Pharmaceutics*, vol. 11, no. 1, pp. 304–316, 2014.
- [274] M. Yang and U. Bierbach, "Metal-Containing Pharmacophores in Molecularly Targeted Anticancer Therapies and Diagnostics," *European Journal of Inorganic Chemistry*, vol. 2017, no. 12, pp. 1561–1572, 2017.
- [275] J. D. Walker, M. Enache, and M. C. Newman, *Fundamental QSARs for Metal Ions*. CRC Press, 2012.
- [276] G. Rubez, J.-M. Etancelin, X. Vigouroux, M. Krajecki, J.-C. Boisson, and E. Hénon, "GPU accelerated implementation of NCI calculations using promolecular density," *Journal of Computational Chemistry*, vol. 38, no. 14, pp. 1071–1083, 2017.
- [277] J. A. K. Howard, P. A. Keller, T. Vogt, A. L. Taylor, N. D. Dix, and J. L. Spencer, "Low-temperature neutron diffraction study of [ReH<sub>5</sub>(PPh<sub>2</sub>)<sub>2</sub>(SiHPh<sub>2</sub>)<sub>2</sub>] and low-temperature X-ray diffraction study of [ReH<sub>5</sub>(PCyp<sub>3</sub>)<sub>2</sub>(SiH<sub>2</sub>Ph)<sub>2</sub>]," *Acta Crystallographica Section B Structural Science*, vol. 48, no. 4, pp. 438–444, 1992.

- [278] V. Streltsov and M. Hrmova, "Crystal structure analysis of plant exohydrolase," 2015.
- [279] S. Guthe, L. Kapinos, A. Moglich, S. Meier, T. Kieffhaber, and S. Grzesiek, "Trimeric foldon of the t4 phagehead fibrin," 2004.
- [280] S. J. Cho and Y. Sun, "FLAME: a program to flexibly align molecules," *Journal of Chemical Information and Modeling*, vol. 46, no. 1, pp. 298–306, 2006.
- [281] X. Gironés, D. Robert, and R. Carbó-Dorca, "TGSA: A molecular superposition program based on topo-geometrical considerations," *Journal of Computational Chemistry*, vol. 22, no. 2, pp. 255–263, 2001.



BY JAIME RODRÍGUEZ-GUERRA PEDREGAL, 2018