THESE DE DOCTORAT

NNT : 2020UPASA001

Computational Fragment-Based Design of Chemically Modified Oligonucleotides for Selective Protein Inhibition: BACE1 as a Case Study

Doctoral thesis from the University of Havana and the University of Paris-Saclay

École doctorale n° 577: Structure et Dynamique des Systèmes Vivants
Spécialité de doctorat: Sciences de la vie et de la santé
Graduate School : Life Sciences and Health, Référent : —

Thesis prepared at the Laboratorio de Química Computacional y Teórica (LQCT) and the Laboratoire Séquence Structure et Fontion des ARN (SSFA), under the joint supervision of Luis MONTERO CABRERA and Fabrice LECLERC

Thesis defended at La Havana, the 18th December of 2023, by

Roy GONZÁLEZ-ALEMÁN

Jury composition

Maykel Marquez-Mijares Dr. C., University of Havana	President
Ricardo Bringas-Pérez Dr. C., Center for Genetic Engineering and Biotechnology	Rapporteur
Matthieu Montès PhD, Conservatoire National des Arts et Métiers	Rapporteur
Karina García-Martínez Dra. C., Center of Molecular Immunology	Examinatrice
Tâp Ha-Duong PhD, Université Paris Saclay	Examinateur
Luis Montero-Cabrera PhD, University of Havana	Directeur de thèse
Fabrice Leclerc PhD, Université Paris Saclay	Directeur de thèse

université PARIS-SACLAY | ÉCOLE DOCTORALE Structure et dynamique des systèmes vivants (SDSV)

Title: Computational fragment-based design of chemically modified oligonucleotides for selective protein inhibition: BACE1 as a case study

Keywords: fragment-based drug design, BACE1, modified oligonucleotides, conformational clustering

Abstract: Fragment-based drug design (FBDD) has become an increasingly popular approach in ligand design, boasting numerous success stories within the drug discovery process. Despite some challenges relating to synthetic accessibility and ligand-design strategies, FBDD remains a promising method for addressing chemical space, molecular complexity, binding probability, and ligand efficiency. RNA therapeutics are rapidly expanding, undergoing a resurgence due to the several advantages of these molecules over traditional drugs and antibodies, including their small size, ease of synthesis, stability, and lack of immunogenicity. However, like many other drugs, off-target effects (where inhibitors designed for a specific molecule inadvertently inhibit others unintended) can hinder their usage. In this study, we introduce an in silico strategy for the fragment-based designing of a promising class of ligands: chemically modified oligonucleotides that exhibit potential selectivity for their intended targets. As a proof of concept, we employed the BACE1 enzyme, a well-established therapeutic target for Alzheimer's disease. The fragments library exploration was conducted through extensive docking simulations of mono-nucleotides using the Multiple Copy Simultaneous Search method, whose docking and screening power were rigorously assessed through a comprehensive benchmark of 121 nucleotide-protein complexes for the first time. An efficient nucleotide assembler was developed to link the best hits obtained in docking stages. Differential analysis of the best-scored oligonucleotides allowed us to find specific binding modes to BACE1 over BACE2. At a methodological level, we also propose substantial memory optimization of four widely employed clustering algorithms, which allow the identification of essential structural features for ligand-receptor binding, an integral part of any FBDD campaign.

List of Tables

List of Figures

Contents

List of Abbreviatures and Acronyms

Aβ Amyloid-beta

ABNR Adopted Basis Newton-Raphson

AD Alzheimer's Disease

AMBER Assisted Model Building and Energy Refinement

AMP Adenosine monophosphate

APP Amyloid Precursor Protein

ARI Adjusted Rand Index

ASR Active Site Region

BACE-X BACE1 and BACE2

BACE1 β-site Amyloid Precursor Protein Cleaving Enzyme 1

BACE2 β-site Amyloid Precursor Protein Cleaving Enzyme 2

BINANA BINding ANAlyzer

CADRO Common Alzheimer's Disease Research Ontology

CASF Comparative Assessment of Scoring Functions

CHARMM Chemistry at Harvard Molecular Mechanics

cLogP Octanol-Water Partition Coefficient

CLoNe Clustering based on Local density Neighborhoods

CMO Chemically Modified Oligonucleotide

CMP Cytidine monophosphate

CphMD Constant pH Molecular Dynamics

CRISPR Clustered Regularly Interspaced Short Palindromic Repeats

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DECODE DiscovEring Clusters Of Different dEnsities

DENCLUE DENsity-based CLUstEring

DFS Depth-First Search

DMT Disease-modifying Therapy

DNA Deoxyribonucleic Acid

DP Density Peaks

FBD Fragment-Based Design

FBDD Fragment-Based Drug Design

FDA United States Food and Drug Administration

GB Generalized Born

GMP Guanosine monophosphate

gRNA guide Ribonucleic Acid

GROMACS GROningen MAchine for Chemical Simulation

gSkeletonClu graph-skeleton based clustering

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise

HTS High Throughput Screening

HTVS High Throughput Virtual Screening

kd-tree k-dimensional tree

LE Ligand Efficiency

LNA Locked Nucleic Acid

MCP Maximum Clique Problem

MCSS Multiple-Copy Simultaneous Search

MD Molecular Dynamics

miRNA micro Ribonucleic Acid

ML Machine Learning

MM Molecular Mechanics

MRD Mutual Reachability Distance

mRNA messenger Ribonucleic Acid

MSF Minimum Spanning Forest

MST Minimum Spanning Tree

NMR Nuclear Magnetic Resonance

NUCLEAR NUCLEotide AssembleR

PCR Polymerase Chain Reaction

PDB Protein Data Bank

QM Quantum Mechanics

QSAR Quantitative Structure-Activity Relationships

QSPR Quantitative Structure-Property Relationships

QT Quality Threshold

RAM Random Acces Memory

RBP Ribonucleic Acid Binding Protein

RCDPeaks Refined-Core Density Peaks

REMD Replica Exchange Molecular Dynamics

RI Rand Index

RMSD Root Mean Square Deviation

RNA Ribonucleic Acid

RRM RNA recognition motif

SD Steepest Descent

SELEX Systematic Evolution of Ligands by Exponential Enrichment

SIGKDD Special Interest Group on Knowledge Discovery and Data Mining

siRNA small interfering Ribonucleic Acid

SOMAmer Slow Off-rate Modified Aptamer

SPR Surface Plasmon Resonance

ssRNA Single stranded RNA

TI Tanimoto Index

UMP Uridine monophosphate

VMD Visual Molecular Dynamics

vp Vantage point

vp-tree Vantage point tree

WHO World Health Organization

XRC X-Ray Cristallography

List of Publications

The research comprising this thesis has resulted in several peer-reviewed publications, which are presented below. These works represent novel scientific contributions made throughout the doctoral program studies and have helped to advance knowledge in the field. The publications have already gained visibility and recognition from peers, totaling 65 citations. The symbol * denotes corresponding authorship.

1. González-Alemán, R.*, Hernández-Castillo, D., Caballero, J., and Montero-Cabrera, L. A. ***Quality Threshold Clustering of Molecular Dynamics: A Word of Caution.*** Journal of Chemical Information and Modeling, 60(2):467–472, 2020. doi: 10.1021/acs.jcim.9b00558.

2. González-Alemán, R.*, Hernández-Castillo, D., Rodríguez-Serradet, A., Caballero, J., Hernández-Rodríguez, E. W., and Montero-Cabrera, L. ***BitClust: Fast Geometrical Clustering of Long Molecular Dynamics Simulations.*** Journal of Chemical Information and Modeling, 60(2):444–448,2020. doi: 10.1021/acs.jcim.9b00828.

3. González-Alemán, R., Chevrollier, N., Simoes, M., Montero-Cabrera, L., and Leclerc, F. ***MCSS-Based Predictions of Binding Mode and Selectivity of Nucleotide Ligands***. Journal of Chemical Theory and Computation, 17(4):2599–2618, 2021. doi: 10.1021/acs.jctc.0c01339.

4. González-Alemán, R.*, Platero-Rochart, D., Hernández-Castillo, D., Hernández-Rodríguez, E. W., Caballero, J., Leclerc, F., and Montero-Cabrera, L. ***BitQT: a graph-based approach to the quality threshold clustering of molecular dynamics.*** Bioinformatics, 38(1):73–79, 2022. doi: 10.1093/bioinformatics/btab595.

5. Platero-Rochart, D., González-Alemán, R.*, Hernández-Rodriguez, E. W., Leclerc, F., Caballero, J., and Montero-Cabrera, L. ***RCDPeaks: Memory-efficient density peaks clustering of long molecular dynamics***. Bioinformatics, 38(7):1863–1869, 2022. doi: 10.1093/bioinformatics/btac021.

6. González-Alemán, R.*, Platero-Rochart, D., Rodríguez-Serradet, A., Hernández-Rodríguez, E. W., Caballero, J., Leclerc, F., and Montero-Cabrera, L. ***MDSCAN: RMSD-based HDBSCAN clustering of long molecular dynamics.*** Bioinformatics (Oxford, England), 38(23):5191–5198, 2022. doi: 10.1093/bioinformatics/btac666.

INTRODUCTION

Drug discovery is a complex and lengthy process that involves several stages, from identifying a biological target to approving a new drug by regulatory agencies. The task can take several years and billions of dollars, with a high failure rate at each stage[1]. The first of these stages involves identifying a biological target that plays a role in a disease process. Once identified, researchers use various approaches to discover potential drug candidates that can interact with the target and modulate its activity[2].

From the family of drug targets known so far, proteins are the most common members[3]. Their inhibition plays an essential role in drug discovery, as it provides a means to suppress their participation in a particular disease. However, blocking a specific protein is challenging and frequently encounters the so-called *off-target effect*. This term describes the events that can occur when a drug binds to targets (proteins or other molecules in the body) other than those for which it was meant to bind, causing unexpected and potentially harmful side effects[4].

Fragment-Based Drug Design (FBDD) is a rational way to conceive the design of protein inhibitors. It begins with a small collection of low-molecular-mass, low-affinity molecules called fragments and then scales them into drug leads[5]. There are several success stories when FBDD has been applied to drug design and discovery, with more than 30 fragment-based drug candidates entering the clinic since the mid-1990s[5–9].

Ribonucleic Acid (RNA) and derived molecules have emerged as promising tools for selective protein inhibition due to their high specificity, low immunogenicity, and tunable physicochemical properties[10]. For example, the development of aptamers (short single-stranded Deoxyribonucleic Acid (DNA) or RNA oligonucleotides) as therapeutic agents has been a subject of intense research, with numerous studies reporting their successful application in the treatment of various diseases[11]. Aptamers' advantages include their ease of generation, low manufacturing cost, and low immunogenicity. However, these molecules must undergo chemical modifications to avoid their inherent susceptibility to nuclease hydrolysis and rapid clearance through glomerular filtration[12].

Proven allies of experimental drug discovery campaigns are the computational or *in silico* methods, which mitigate time and resource costs through virtual simulations. When looking for new drugs, a neuralgic step is screening immense databases representative of the drug-like chemical space to find a suitable molecular cure for a disease. Computational docking methodologies play a vital role, and numerous alternatives are available to researchers[13]. In the FBDD arena, the Multiple-Copy Simultaneous Search (MCSS) software[14] stands out as a pioneering virtual methodology for docking that has been previously coupled with other software to join fragments into lead compounds[15–19].

Clustering algorithms (devoted to grouping similar entities into sets called clusters)[20] are employed in various stages of the FBDD pipeline (though often in a transparent manner to users), primarily to group similar fragments or compounds based on their structural or physicochemical properties. Conceiving new efficient clustering algorithms

or optimizing those currently used is mandatory to face the increasing size of molecular ensembles generated by computational techniques.

The present thesis focuses on the computational fragment-based design of chemically modified oligonucleotides (inspired by the aptamers' development and success) for selective protein inhibition, using β-site Amyloid Precursor Protein Cleaving Enzyme 1 (BACE1) as a case study. BACE1 is a well-established therapeutic target for the Alzheimer's Disease (AD) due to its essential role in the production of amyloid-beta peptides, which are the primary constituents of amyloid plaques in the brains of Alzheimer's patients. One of the main disadvantages of targeting BACE1 resides in the off-target inhibition evinced by a related protease, β-site Amyloid Precursor Protein Cleaving Enzyme 2 (BACE2)[21].

Although RNA therapies (including aptamers alternatives) are becoming increasingly popular, there is no available methodology for the rational design of selective chemically modified oligonucleotides for medical applications, which constitutes the **scientific problem** here addressed.

The following work was conceived on the global **hypothesis** that FBDD principles can be effectively applied to the rational *in silico* design of chemically modified oligonucleotides with high affinity and selectivity for protein targets.

The **main objective** of this thesis is to develop an integrated computational framework for the fragment-based design of chemically modified oligonucleotides with high affinity and selectivity for protein targets, using BACE1 as a relevant proof of concept. The **specific objectives** that guided our efforts are the following:

1. To asses the MCSS's docking and screening powers on a representative benchmark of protein-nucleotide complexes.
2. To optimize popular clustering algorithms that intervene at distinct phases of the FBDD.
3. To implement an efficient computational linker for assembling chemically modified nucleotides (fragments) onto oligochains (lead compounds).
4. To validate a computational workflow for proposing chemically modified oligonucleotides as selective proteins inhibitors using the BACE1 as a case study.

This manuscript contains several contributions that can be highlighted in terms of **novelty**:

1. For the first time, the docking and screening power of the MCSS software was evaluated on a set of 121 representative protein-(mono)nucleotide complexes and compared to other established scoring functions.
2. For each of the four clustering algorithms treated in our work, a novel idea was implemented that significantly impacted their spatial complexity.
   (a) A binary encoding of the pairwise molecular similarity (as well as the ability to translate clustering steps into bitwise operations) was applied to the Daura and the Quality Threshold clustering.

   (b) Also, a methodological correction was raised for these two algorithms that were incorrectly and systematically used interchangeably.

   (c) The spatial complexity of the Density Peaks and the Hierarchical Density-Based Spatial Clustering of Applications with Noise algorithms was reduced from quadratic to linear.

3. No other published tool is available to efficiently link MCSS-generated poses of modified nucleotides onto oligonucleotides, a task we addressed with our proposed software NUCLEotide AssembleR (NUCLEAR).

4. To our knowledge, an attempt to provide a computational workflow yielding potentially selective oligonucleotides to inhibit protein conformations has not been reported.

1 - BIBLIOGRAPHIC REVIEW

This chapter explains core concepts, definitions, antecedents, and motivations of the thesis work later presented in results Chapters 3 to 6 through a deliberate organization that covers, in the first place, the principles and rationale behind the Fragment-Based Drug Design (FBDD) methodology, emphasizing its *in silico* component in Section 1.1.

As it is an integral part of the computational FBDD, the basics of docking simulations and scoring functions are reviewed in Section 1.2, highlighting the Multiple-Copy Simultaneous Search (MCSS) alternative given the significant importance this methodology plays in our approach to conceive selective oligonucleotides.

The main motivation to our study of CMOs as potential drugs, comes from the relative success of aptamers and aptamers-like molecules. So these antecedents are described in Section 1.3 where we also briefly touch some important aspects of Ribonucleic Acid (RNA) therapeutics.

In Section 1.4, we detail the role of the β-site Amyloid Precursor Protein Cleaving Enzyme 1 (BACE1) protein in the Alzheimer's Disease (AD). In this section, we make clear the importance of selective inhibition of this target and the usefulness of avoiding off-target effects with the related β-site Amyloid Precursor Protein Cleaving Enzyme 2 (BACE2) protein.

As this manuscript proposes significant contributions to the field of clustering molecular ensembles, we dedicated Section 1.6 to describe the machinery of several (often confused even) algorithms widely employed in the molecular field.

However, as a mathematical background will be necessary to comprehend this and another essential contribution we made to link nucleotide fragments in Chapter 5, we first gather some basic mathematical background on asymptotic analysis of function growth, graph theory, similarity measures, essential data-structures, and bitwise operations in Section 1.5.

1.1 . Fragment-based drug design

The drug discovery process is a complex and lengthy process that involves several stages, starting from identifying a biological target and ending with approving a new drug by regulatory agencies. The process can take several years and billions of dollars to complete, with a high failure rate at each stage[1].

The first of these stages involves identifying a biological target that plays a role in a disease process. Once a target is identified, researchers use various approaches to identify potential drug candidates that can interact with the target and modulate its activity[2].

High Throughput Screening (HTS) is a drug discovery approach that involves rapidly testing many compounds for their ability to bind to a specific target or produce a desired biological effect. It is a widely used approach in the pharmaceutical industry and academic research to identify potential drug candidates from vast chemical libraries. HTS typically

involves using automated technologies, such as robotics and liquid handling systems, to perform assays on hundreds of thousands or even millions of compounds in a relatively short time[22].

Although HTS has revolutionized the drug discovery process and led to the discovery of many essential drugs (including some that have become blockbuster drugs in the market), when screened against newer or more difficult targets, huge compound libraries sometimes yielded few hits or, in more problematic cases, yielded hits that were false positives[23]. Besides, there is a growing awareness of the enormity of chemical space[24], from which an early estimate put the number of possible small drug-like molecules at 10^{60}, an immensity compared to the fragment library sizes available up to date (in the scale of the millions).

Unlike conventional drug discovery methods that involve screening millions of compounds to find drug-sized starting points, FBDD takes a different approach (see Figure 1.1). It begins with smaller collections of low-molecular-mass, low-affinity molecules called fragments and then scales them up into drug leads[5].

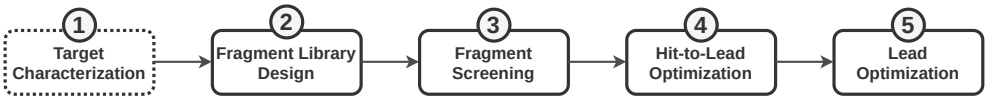


Figure 1.1: *Typical steps involved in an FBDD campaign.*

Despite some challenges related to ligand-design strategies and synthetic accessibility, fragment-based approaches remain highly attractive as they deal more efficiently with chemical space, molecular complexity, probability of binding, and Ligand Efficiency (LE)[23]. After the HTS, FBDD approaches represent one of the most critical lead-generation strategies for clinical candidates[25]. There are several examples of success stories when FBDD has been applied to drug design and discovery, with more than 30 fragment-based drug candidates entering the clinic since the mid-1990s[5–9].

In spite of its promising future, the current FBDD approaches may face limitations and challenges that hinder its widespread application: (i) large-scale pure, high-quality, and stable target proteins are required for the screening but the crystallization of certain kinds of proteins can be expensive and time-consuming, (ii) given the relatively small sizes of fragments, limited receptor-ligand interactions can be formed with the surrounding residues and only a part of them is strong enough for detection, (iii) only fragments with relatively high solubility can be suitable for the screening step, (iv) FBDD endeavors also face the total diversity space (10^3 fragments can typically sample the chemical diversity space of 10^9 molecules) that restrains the exploration of a larger region of the drug-like space, (v) it is known that FBDD is more suitable for certain classes of targets (such as kinases) whose binding site often consists of multiple distinctive sub-sites , (vi) identifying the optimal linkers in the hit-to-lead step can be challenging and time-consuming[26], (vii) many proteins are flexible, which can make it cumbersome to design fragments that bind to a specific conformation, (viii) most of the FBDD methods do not take ligand specificity or selectivity into account.

Above-mentioned restrictions progressively drove researchers to consider chemo-informatics and computational methodologies that could help at different stages of the FBDD efficiently and cost-effectively[27,28]. However, much like the experimental FBDD, their computational counterparts have a set of restraints mainly regarding the lack of accuracy and reliability due to the methodological approximations that govern them[29–32].

In silico methods can generate false positives or compounds predicted to bind to the target protein but do not have a therapeutic effect. This can lead to wasted time and resources in the experimental validation of compounds that are not promising drug candidates. The takeaway lessons point, not surprisingly, to the complementary use of both approaches to advance the field of FBDD[33].

Next, we will give a succinct overview of the primary stages illustrated in Figure 1.1, highlighting the role of computational methods in each step.

1.1.1 . Target characterization

Drugs are compounds that interact with a biological system to produce a biological response. Those molecular structures with a binding site into which the drug fits and binds are termed *drug targets*[34]. Although proteins are the most common drug targets, species-specific genes, Deoxyribonucleic Acid (DNA), RNA, and membranes have also been recognized as such[3].

Usually considered a pre-requisite more than a first step for FBDD, drug target identification and characterization is the first step in drug discovery[35]. The primary goal is to identify a target macromolecule or biological pathway responsible for causing or contributing to a specific disease. This macromolecule or pathway may serve as a point of intervention for a potential drug to treat or manage the disease.

The target identification starts selecting a suitable disease to study. Researchers may choose based on prevalence, disease burden, and unmet medical needs. Next, they search for the molecular basis of the disease, often by studying the genes, proteins, and cellular pathways involved. Several approaches can be employed in this process like genomics[36], transcriptomics[37], proteomics[38] or system biology[39] (integrating multi-omic data to create a holistic understanding of biological systems). Besides, well-established databases and biological and Machine Learning (ML) methods exist to identify drug targets[40].

Once a target has been identified, it is essential to characterize its biological function, molecular interactions, and role in disease pathology. This can be achieved through various experimental and computational methods such as structural biology, biochemical or cell-based assays, and animal models.

1.1.2 . Fragment library design

After the target against which a drug will be designed is known, the next step in the FBDD process that is critical to its success is to design a fragment library. It is imperative to have a diverse fragment library encompassing a wide range of chemical space and comprising compounds with the potential to bind to the target of interest.

Traditionally, the FBDD campaigns have been applied to design ligands assembled using small chemical groups derived from such fragment libraries. It is common that

these libraries contain molecules that conform to the rule of three[41], which specifies that compounds should possess (i) a molecular weight under 300 Da, (ii) less than three hydrogen-bond donors and acceptors, (iii) fewer than three rotatable bonds, and (iv) an Octanol-Water Partition Coefficient (cLogP) of three or less. This rule is merely a guideline that should not be over-emphasized, and some arguments challenge its validity[42]. Nevertheless, it remains the preferred model for fragment selection within libraries.

As the composition of the library used in an FBDD project has a direct impact on the outcome[43,44], it is important to analyze commercially available fragments and fragment libraries to make a choice that meets the primary criteria based on the profile of the target being studied[45,46].

Commercially available fragment libraries are typically selected based on chemical and size diversity and different well-balanced properties to cover essential features. Natural products or natural-product-inspired fragments can be included as they are often helpful[47]. Additionally, identifying a series of non-commercially available fragments from synthetic chemistry efforts, such as *in-house* libraries or collaborating groups, is important for future medicinal chemistry optimization strategies[48].

Researchers usually have their customized libraries in FBDD, and the molecular weight of a fragment can be above 300 Da. These libraries are based on their respective experience and usually do not contain molecules that are reactive to targets, bind to proteins unspecifically, form aggregate, or form covalent bonds with proteins[49]. Recently, Carbery *et. al.*[50] showed that fragment libraries designed to be functionally diverse (ranking fragments by the number of novel interactions they introduce to the library) recover protein critical information more efficiently than standard structurally diverse libraries.

Computational methods can be employed to quickly obtain physicochemical properties, solubility, synthetic accessibility, *etc.* and use them as filters for commercially available fragment databases. Additionally, these *in silico* methods can be utilized to remove fragments with unwanted chemical groups and incorporate the most frequently occurring fragments from known drugs, thus ensuring good diversity to represent drug-like chemical space. Approaches such as Quantitative Structure-Activity Relationships (QSAR) and Quantitative Structure-Property Relationships (QSPR) modeling can be used to predict aqueous solubility as the fragments must be highly soluble (they are screened at high concentrations)[51,52].

1.1.3 . Fragment screening

Once the fragment library is conceived, screening it against the target of interest is necessary. As fragments from the library are expected to bind weakly to the target protein, the screening assay must be sensitive enough to detect such interactions and robust to prevent false identification of hits, which can arise due to interference with the assay readout. Classical biophysical assays, including Nuclear Magnetic Resonance (NMR) spectroscopy, Surface Plasmon Resonance (SPR), and X-Ray Cristallography (XRC), have been found to meet the requirements for sensitivity and robustness[53].

Molecular *docking* is a computational method to predict small molecules' binding mode and affinity (ligands or fragments) to a target active site[54]. It simulates the binding

process between the ligand and the target to predict the most favorable binding pose (sampling) and the corresponding binding energy (scoring). During the fragment screening stage, docking has been used as a pre-screen tool to reduce experimental efforts as a vital component of High Throughput Virtual Screening (HTVS).

HTVS is a valuable tool in the early stages of drug discovery, which complements HTS by attempting to identify potential hits. The primary distinction between HTS and HTVS is that HTS is an experimental approach, while HTVS is a theoretical one. In HTS, many compounds are screened to determine their ability to interact with target molecules by assessing whether a compound reacts biochemically with the target.

While most investigators rely on a familiar or laboratory-available docking program, this attitude may not be optimal. Instead, suppose the structure of the receptor-small molecule complex is known (*e.g.*, from the PDB ID of the target). In that case, multiple docking algorithms should be employed to determine which one places the small molecule in the same orientation as observed in the crystal structure[13].

After identifying the docking program that replicates the experimental pose observed in the crystal structure, the researcher should evaluate how the pose is ranked by the native scoring function, which is included with the docking program. If the pose is ranked at the top, this docking and scoring method should be employed in the HTVS experiment. Otherwise, the researcher should re-score the poses using alternative scoring functions.

For the results discussed in this manuscript, the significance of the selected docking method is prominent. Therefore, we will delve into this methodology's fundamental principles and the rationale behind our research choice (namely MCSS) in the subsequent Sections 1.2 and 1.2.1, respectively.

1.1.4 . Hit-to-lead optimization

Next to hit identification, the FBDD process moves to the lead-generation stage[55,56]. The prioritization of fragment hits involves considering multiple parameters, including biological activity, LE[57], solubility, ease of synthesis, availability of commercial analogs, and structural information regarding the binding mode.

Either by linking, merging or growing, prioritized fragments must be joined together[23]. The linking strategy offers theoretically better perspectives for gaining binding energy[58], as it connects two non-competitive fragments by fusing some chemical bonds or creating some additional through a spacer molecule[59]. Computational methods (*e.g. de novo* drug design algorithms[60]) can iteratively assist the buildup of the fragment hits into a new lead compound by virtually screening linker libraries[61].

In bio-polymers, the chemical connectivity is well-defined, and the linking strategy does not require a spacer to be part of the fragments. Thus, the linking solves a distance-constraint problem in joining the connecting atoms of successive residues and guarantees a straightforward chemical synthesis. On the other hand, the chemical diversity is reduced to that of the residues (20 for unmodified amino acids, 4 or 5 in the case of unmodified nucleotides).

1.1.5 . Lead optimization

Lead optimization is a crucial process in drug discovery that involves the identification of a pre-clinical candidate with optimal biological activity and drug-like properties[2]. Following hit-to-lead efforts, the most promising hit series are advanced to the lead optimization stage, where extensive optimization of both biological activity and physicochemical properties is carried out[62]. This is achieved through a dedicated screening funnel of both *in vitro* and *in vivo* assays that are designed to evaluate the physio-chemical properties of lead compounds and identify the best ones for formulation and dosing. Lead optimization is a highly iterative process, which requires robust and efficient screening assays to prioritize compounds with optimal drug-like properties.

1.2 . Docking simulations

The molecular docking process begins by posing small molecules in the active site. This is challenging due to the conformational degrees of freedom that even a simple molecule can exhibit. Sampling these degrees of freedom must be performed with sufficient accuracy to identify the conformation that best matches the receptor structure and must be fast enough to permit the evaluation of thousands of compounds in a given docking run.

Posing algorithms are complemented by scoring functions that are designed to predict the biological activity through the evaluation of interactions between compounds and potential targets. Relatively simple scoring functions continue to be heavily used, at least during the early stages of docking simulations. Pre-selected conformers are often further evaluated using more complex scoring schemes with more detailed treatment of electrostatic and van der Waals interactions, and inclusion of at least some solvation or entropic effects[63].

It is worth mentioning that the binding of ligands is influenced by both enthalpic and entropic factors, and in some cases, one may dominate over the other. This can pose a challenge for current scoring functions, as they tend to prioritize capturing energetic effects rather than entropic ones.

In addition to problems associated with scoring of compound conformations, other complications exist that make it challenging to accurately predict binding conformations and compound activity. These include, among others, limited resolution of crystallographic targets, inherent flexibility, induced fit or other conformational changes that occur on binding, and the participation of water molecules in protein–ligand interactions[64].

An objective test of docking methods performed in 1997 confirmed the assumption that recognizing near-native geometries and predicting their affinities could be achieved only with limited success, whereas the problem of generating reasonable ligand orientations is considered to be virtually resolved[65], at least for proteins with rather rigid binding pockets, not involving any water molecules in binding and without any change in protonation state of either ligand or protein upon binding[66].

As a variety of scoring functions have been developed so far, some objective benchmarks are desired for assessing their strengths and weaknesses. The Comparative Assess-

ment of Scoring Functions (CASF) benchmark serves this purpose as it has been designed as a "scoring benchmark", where the scoring process is decoupled from the docking process to depict the performance of scoring function more precisely.

Developers of CASF proposed four metrics in 2013 that were later improved in 2016[67]. They refer to the ability of a scoring function to: (i) produce binding scores in a linear correlation with experimental binding data (scoring power), (ii) correctly rank the known ligands of a certain target protein by their binding affinities when the precise binding poses of those ligands are given (ranking power), (iii) identify the native ligand binding pose among computer-generated decoys (docking power), and (iv) identify the true binders to a given target protein among a pool of random molecules (screening power).

There are various types of scoring functions available. Below, we will provide a succinct overview of the most frequently used ones.

Forcefield- or physics-based scoring functions are extensively used due to their ability to estimate the potential energy of a protein-ligand complex based on classical mechanics. These scoring functions use empirical parameters to describe the interactions between atoms and molecules, including van der Waals forces, electrostatics, and hydrogen bonding, which makes them computationally efficient and suitable for molecular dynamics simulations and other computational modeling studies.

Despite their advantages, forcefield-based scoring functions often neglect the contributions of entropy and solvent effects, which can limit their accuracy in predicting the behavior of molecules in complex biological systems[59]. To improve their accuracy, researchers have incorporated additional factors, such as the torsional entropy of ligands to account for the flexibility of the ligand molecule[68], and solvation/desolvation effects by using explicit solvent models[69], implicit solvent models[70], or a combination of both.

Scoring functions based on Quantum Mechanics (QM) have been developed to address the challenges of covalent interactions, polarization, and charge transfer in docking[71,72]. However, with greater accuracy comes the prohibitively computational cost. As a result, hybrid QM/Molecular Mechanics (MM) approaches have been developed as a compromise solution[73].

Pason & Sotriffer defined **empirical scoring functions** as all functions derived from experimental structures and affinity data by means of descriptors (of any kind) and a statistical regression model[74]. Classical empirical scoring functions try to estimate the affinity as a (linear) sum of individual contributions deemed to be important for the binding free energy such as hydrogen bonds, hydrophobic effects, and steric clashes. The recent literature makes a distinction between these functions and (usually non-linear) descriptor-based[75] or ML[76] scoring functions, although such a taxonomy is rather arbitrary.

The development of empirical scoring functions is based on three components: (i) a training set of experimental protein-ligand complex structures along with their affinities, (ii) descriptors that capture numerically the structural features of the protein-ligand interaction, and (iii) a regression method to establish a quantitative relationship between the descriptor-encoded structural information and the experimental affinity[74].

Although the empirical scoring functions decompose protein–ligand binding affinities into several individual energy terms, similar to physic-based scoring functions, they usually employ a flexible and intuitive functional form other than using the well- established models that physics-based scoring functions use.

Because of their simple energy terms, these scoring functions are good at predicting binding affinity, ligand pose, and virtual screening with low computing cost[77], but they are poorly suited for describing the relationship between binding affinity and the crystal structures and they encounter double-counting problems. Autodock Vina[32] and its derivative Vinardo[78] are examples of this kind of scoring functions.

In Vina, the binding energy is predicted as the sum of distance-dependent atom pair interactions (see Equation 1.1).

$$E = \sum e_{pair}(d) \tag{1.1}$$

Here d is the surface distance calculated with Equation 1.2, where r is the inter-atomic distance and R_i and R_j are the radii of the atoms in the pair.

$$d = r - R_i - R_j \tag{1.2}$$

Every atom pair interacts through a steric interaction given by the first three terms of Equation 1.3. Also, depending on the atom type, there could be hydrophobic and non-directional H-bonding interactions, given by the last two terms of Equation 1.3.

$$e_{pair}(d) = \begin{cases} w_1 * Gauss_1(d) + \\ w_2 * Gauss_2(d) + \\ w_3 * Repulsion(d) + \\ w_4 * Hydrophobic(d) + \\ w_5 * HBond(d) + \end{cases} \tag{1.3}$$

Steric interaction in Vina is assessed using three terms (Equations 1.4 to 1.6). If both atoms in the pair are hydrophobic the linear function in Equation 1.7 is included. Also, if the pair consists of an H-bond donor and an H-bond acceptor, Equation 1.8 is added. These last two equations are simple piecewise linear and their effect can be thought of as modifying the steric interaction in order to produce an increased attraction when these types of interaction are present.

$$Gauss_1 = e^{-((d-o_1)/s_1)^2} \tag{1.4}$$

$$Gauss_2 = e^{-((d-o_2)/s_2)^2} \tag{1.5}$$

$$Repulsion(d) = \begin{cases} d^2 \ for \ d \ \leq 0 \\ 0 \ for \ d = 0 \end{cases} \tag{1.6}$$

$$Hydrophobic(d) = \begin{cases} 1 \; for \; d \leq p_1 \\ p_2 - d \; for \; p_1 > d < p_2 \\ 0 \; for \; d \geq p_2 \end{cases} \tag{1.7}$$

$$Hbond(d) = \begin{cases} 1 \; for \; d \leq h_1 \\ \frac{d}{-h_1} \; for \; h_1 < d < 0 \\ 0 \; for \; d \geq \; 0 \end{cases} \tag{1.8}$$

The mechanism by which these terms were selected for the Vina scoring function, the parameters used therein, and the weight of each term are unclear, although some kind of non-linear regression on the PDBBIND 2007 database was used[78].

On the other hand, Vinardo was generated by a variety of scoring functions which consisted of the inclusion/exclusion of several interaction terms to the Vina scoring function. The terms considered for addition or exclusion were Gaussian steric attractions, quadratic steric repulsions, Lennard-Jones potentials, electrostatic interactions, hydrophobic interactions, non-hydrophobic interactions, and non-directional hydrogen bonds. These terms are all pairwise additive and are part of the 26 currently available in the Smina program following a procedure detailed in reference 78.

Knowledge-based scoring functions (also known as **statistical potentials**) derive pairwise potentials from three-dimensional structures of a large set of protein-ligand complexes based on the inverse Boltzmann statistic principle without requiring the fitting of empirical parameters[79,80]. The frequency of different atom pairs at different distances is assumed to be related to the interaction of two atoms. It is converted into the distance-dependent potential of mean force.

Because they are based on the physical atomic interactions, knowledge-based scoring functions are more interpretable and can provide insights into the underlying molecular mechanisms. The most significant advantage of knowledge-based scoring functions is their compromise between computing cost and predictive accuracy compared to physics-based and empirical scoring functions.

Training sets for knowledge-based scoring functions consist only of structural information. They are independent of experimental binding affinity data, avoiding possible binding affinity ambiguities caused by experimental conditions and making them suitable for binding pose prediction rather than binding affinities[81]. Researchers have focused on extending pairwise potentials to many-body potentials by introducing several new parameters to increase predictive accuracy. Examples of these scoring functions include DrugScore[82], DSX[83], M-Score[84], and ITscorePR[85], the latter being specifically developed for protein-RNA interactions.

As detailed in the original publication, the basic idea behind the ITscorePR method is to improve the inter-atomic pair potentials step by step though iterations by comparing the predicted pair distribution functions of the protein-RNA complexes and the experimentally observed pair distribution functions of the crystal structures in the training set. The method can be represented mathematically by the following iterative formula:

$$u_{ij}^{(n+1)}(r) = u_{ij}^{(n)}(r) + \Delta u_{ij}^{(n)}(r), \Delta u_{ij}^{(n)}(r) = \frac{1}{2} k_B T [g_{ij}^{(n)}(r) - g_{ij}^{(obs)}(r)] \quad (1.9)$$

where n stands for the iterative step, i and j represent the types of a pair of atoms in the protein and the RNA, respectively. $g_{ij}^{(obs)}(r)$ stands for the pair distribution function for atom pair ij calculated from the experimentally observed protein-RNA complex structures in the training set:

$$g_{ij}^{(obs)}(r) = \frac{1}{K} \sum_{k=1}^{K} g_{ij}^{k*}(r) \quad (1.10)$$

where K is the total number of the protein-RNA complexes in the training data set and $g_{ij}^{k*}(r)$ is the pair distribution function of the k^{th} native complex structure.

The $g_{ij}^{(n)}(r)$ is the pair distribution function calculated from the ensemble of the binding modes according to the binding score-dependent Boltzmann probabilities P_k^l that are predicted with the trial potentials $u_{ij}^{(n)}(r)$ at the n^{th} step.

$$g_{ij}^{(n)}(r) = \frac{1}{K} \sum_{k=1}^{K} \sum_{l=1}^{L} P_k^l g_{ij}^{kl}(r) \quad (1.11)$$

where $g_{ij}^{kl}(r)$ is the pair distribution function for atom pair ij observed in the l^{th} binding state of the k^{th} protein-RNA complex.

ML-based scoring functions are a type of empirical scoring function that estimate the binding affinity or other properties of protein-ligand complexes. These scoring functions use ML algorithms, such as support vector machine, random forest, neural network, and deep-learning, to learn the relationship between the features of a complex and its binding affinity or other properties.

In contrast to forcefield-based scoring functions, which rely on mathematical models based on classical mechanics to calculate the potential energy of a protein-ligand complex, ML-based scoring functions do not require explicit modeling of the physical interactions between atoms and molecules. Instead, they use statistical models to learn the relationship between the features of a complex and its binding affinity, which can include the shape and electrostatic properties of the ligand and the surface properties of the protein [86].

ML-based scoring functions can predict the properties of novel or unconventional molecules that have not been seen before, as they are not limited by the training set used to derive the statistical potential. In addition, they can be more flexible and adaptable than forcefield-based scoring functions, as they can be trained to recognize complex non-linear relationships between features and binding affinity.

Although ML-based scoring functions may outperform their classical counterpart [87], they are seldom directly incorporated into docking software but are usually used for re-scoring [88]. The reason is that ML-based scoring functions rely on the training dataset [89,90]. If the protein and ligand are docked by classical docking software, and then the docked structure is re-scored by ML scoring functions, the accuracy tends to be improved. Despite being less concrete on the physicochemical basis (it may be challenging to understand

which features of a complex are most important for predicting its binding affinity), they often demonstrated a superior or at least comparable performance to that of classic scoring functions in binding affinity estimation[89].

$\Delta_{vina}RF_{20}$[91] was derived from Vina under the main idea of employing random forest to parameterize corrections to the AutoDock Vina scoring function, and thus to take advantage of both the excellent docking power of the Vina docking function and the strength of random forest in improving scoring accuracy. The original score calculated by the AutoDock Vina program is in the unit of kcal/mol, and can be converted into pK_d unit with the following formula: $pK_d(Vina) = -0.73349E(Vina)$. Thus, the overall $\Delta_{vina}RF$ scoring function can be cast into the following form:

$$pK_d(\Delta_{vina}RF) = pK_d(Vina) + \Delta pK_d(RF) \tag{1.12}$$

where $pK_d(RF)$ is the correction term trained by the random forest (RF) algorithm using $pK_d(train)$, i.e., $pK_d(train) - pK_d(Vina)$.

From the numerous scoring functions that have been developed, none is universally applicable. While some perform well on proteins related to those used to calibrate their parameters, they may be less effective for proteins with different physicochemical properties. Additionally, each scoring function has advantages and disadvantages concerning the model formulated to describe the process of ligand-receptor association. Nevertheless, these characteristics suggest that multiple scoring functions must capture different information.

Based on this idea, **consensus scoring** was introduced by combining the predictions of multiple scoring functions. In the words of their authors: *"The only potential disadvantage we can envision regarding consensus scoring is that it may not perform as well as any specific function in a specific instance, since the intersection of two nonidentical lists is, by definition, smaller than the individual lists. However, since one never knows which function might be optimal upfront, we believe the consistency and efficiency gained by consensus scoring outweighs any potential limitation."* [92]

Several strategies varying in combining each score have been attempted and have shown improvement in predicting binding mode, affinity, or identifying ligands that can effectively bind to a receptor in virtual screening[93–95]. Examples of software using consensus scoring are MultiScore[96], GFscore[97], SeleX-CS[98], and VoteDock[99].

1.2.1 . Multiple Copy Simultaneous Search (MCSS)

MCSS is a computational method used within the framework of FBDD approaches, although it does not include any fragment-assembly strategy[14]. MCSS mainly performs local and iterative docking calculations based on an efficient sampling method[100] which is implemented in the Chemistry at Harvard Molecular Mechanics (CHARMM) program[101]. MCSS is used as a first step in the FBDD process as it generates distributions of functional groups or fragments at the surface of a protein target composed of clustered docking poses[14]. Thus, it is possible to perform virtual screening using pre-defined[14,102,103] or customized fragment libraries[104].

MCSS-based FBDD approaches were applied repetitively to the design of peptides or peptidomimetics[18,19,102,105–107] or to other bio-molecules such as aminoglycosides[108]. MCSS has gained popularity in FBDD approaches in conjunction with fragment-linking/merging methods such as HOOK[15], DLD[16], or CAVEAT[17] for chemical groups, and OLIGO[18] for oligopeptides or SiteMap for peptidomimetics[19].

The MCSS scoring function is based on the CHARMM energy function; different strategies have been applied to improve its performance using more accurate methods and implicit solvent models. The first strategy includes post-processing of the MCSS fragment poses recalculating the score function by adding solvation terms[109], or by re-scoring (single-point energy) using a Generalized Born (GB) model[110,111].

The second less time-consuming strategy is to include solvent effects in the energy function during the MCSS calculations using, for example, a distance-dependent dielectric model[109] or an alternative charge model[108]. Although implicit solvent models have become very popular, their accuracy remains limited for calculating solvation-free energies[112].

The MCSS score is defined by the electrostatic and van der Waals contributions to the interaction energy plus a penalty term corresponding to the deviation of the fragment's conformation from its energy minimum:

$$\Delta E_{\mathrm{MCSS}}^{binding} = \Delta E_{conf}^{fragment} + \Delta E_{vdw}^{inter} + \Delta E_{el}^{inter} \tag{1.13}$$

The van der Waals contribution to the score is calculated in the same way for all solvent models:

$$E_{vdw} = \sum_{excl(i,j)=1} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right) sw(r_{ij}^2, r_{on}^2, r_{off}^2) \tag{1.14}$$

while the electrostatic contribution depends on the solvent model used. In the case of the "FULL" model, it is calculated using the standard charges as follows:

$$E_{el} = \sum_{excl(i,j)=1}^{\epsilon=1} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \tag{1.15}$$

In the case of the other models using either scaled charges (*i.e.*, "SCAL") or standard charges (*i.e.*, "STD") (Figure 1.2), it is calculated this way:

$$E_{el} = \sum_{excl(i,j)=1}^{\epsilon=3} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}^2} sw(r_{ij}^2, r_{on}^2, r_{off}^2) \tag{1.16}$$

where the dielectric constant is set up according to some previous work[108].

1.3 . Oligonucleotide therapeutics

In the past, RNA was not widely regarded as a viable option for therapeutic use due to its short half-life *in vivo*. However, with advancements in stabilization chemistry and

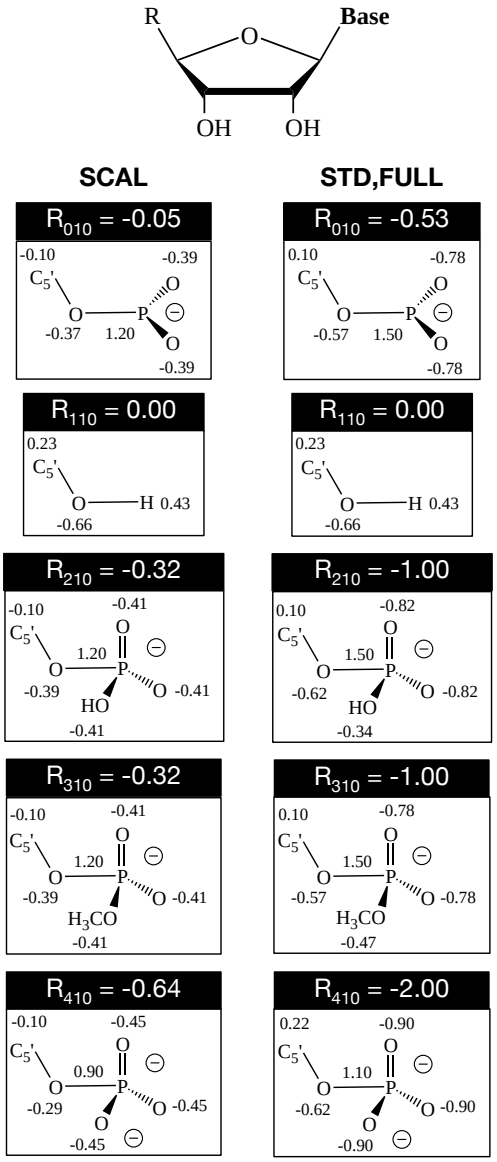


Figure 1.2: *Non-bonded models used in the MCSS calculations. The R group corresponding to the 5' end of the nucleotide includes five flavors: R_{010} (standard nucleotide residue), R_{110} (5'OH patch), R_{210} (5'PO_4H^-), R_{310} (5'$PO_4CH_3^-$), and R_{410} (5'PO_4^{2-}). Three solvent models were used: the SCAL model was based on reduced charges on the phosphate group according to Manning's Theory[113] and applied to nucleic acids[108]; the STD (standard) or FULL models were based on standard charges. The electrostatic contribution to the interaction energy was calculated based on a constant dielectric formulation for the FULL model. The SCAL and STD models were based on a distance-dependent dielectric model. The van der Waals contribution was calculated using the standard CHARMM27 potential energy function[114].*

a better understanding of the clinical value of short-lived molecules, this skepticism has primarily been dispelled[115]. As a result, RNAs have become increasingly recognized as valuable tools and targets for therapeutic interventions[10].

RNAs can adopt complex conformations that enable them to specifically bind to proteins, small molecules, or other nucleic acids. There are four main classes of therapeutic RNAs based on their modes of action[115]: (i) encoding therapeutic proteins or vaccine antigens (mRNAs), (ii) inhibiting pathogenic RNA activity (siRNAs, miRNAs, and antisense RNAs), (iii) modulating protein activity (RNA aptamers), and (iv) reprogramming genetic information (trans-splicing ribozymes and CRISPR gRNAs). This wide range of options for therapeutic targeting of RNA has garnered significant interest from academic research institutions and pharmaceutical companies, with a growing number of approved RNA therapeutics now generating significant profits[116].

In 2019, twelve molecules were approved by the United States Food and Drug Administration (FDA) for treating various pathologies. This includes nine RNA drugs in the form of siRNA or antisense oligonucleotides, two small molecules targeting RNA targets, and one aptamer[117–119]. In addition, research in vaccinology has led to the development of two mRNA-based vaccines[120,121]. With the discovery of new activities performed by RNAs, coupled with the growing recognition that transient induction of a therapeutic effect can lead to long-lasting health benefits, it is anticipated that the clinical development pipeline will continue to be enriched with therapeutic RNAs in the future[115].

1.3.1 . Aptamers and related molecules

Aptamers are short single-stranded DNA or RNA oligonucleotides, typically 15 to 100 bases in length that can specifically recognize targets with high affinity and selectivity. Often called "chemical antibodies" because of this ability, they adopt stable three-dimensional shapes both *in vitro* and *in vivo*[11]. As proteins are the most common drug targets in drug discovery, RNA aptamers that modulate protein activity are increasingly attractive.

Aptamers have a broad range of potential therapeutic applications[122]. *Macugen*[123] is an example used to treat age-related macular degeneration by targeting vascular endothelial growth factor. *AS1411*[124] has shown promise in oncology by targeting nucleolin, an over-expressed protein in cancer cells. *NOX-A12*[125] is another aptamer that has shown potential in oncology by targeting the chemokine $CXCL12$, which is involved in tumor growth and metastasis.

They also have potential applications in other disease areas. For example, *NOX-E36*[126] targets the inflammatory cytokine interleukin-6 and has shown promise in treating type 2 diabetes and *NOX-H94*[127] the pro-inflammatory protein HMGB1 (so having potential applications in treating inflammatory disorders). *ARC1779*[128] is an aptamer that interacts with the von Willebrand factor and has potential applications in the treatment of thrombotic disorders. Furthermore, aptamers have been developed to diagnose and treat tuberculosis[129].

These molecules are selected from large libraries of random oligonucleotides containing up to 10^{16} unique sequences. The process of aptamer selection, termed Systematic

Evolution of Ligands by Exponential Enrichment (SELEX), was first developed in 1990 by Tuerk and Gold[130], and Ellington and Szostak[131]. The selection cycle, whether for DNA or RNA sequences, on proteins, on cellular levels, or in living animals, requires three main steps: (i) incubating a target with a library containing randomized sequences, (ii) partitioning bound sequences from non-bound ones, and (iii) recovering and Polymerase Chain Reaction (PCR) amplifying the bound sequences[132]. The selection cycle is repeated until the sequence is enriched with the desired affinity.

While still considered a gold standard for aptamers generation, SELEX techniques have traditionally been laborious and time-consuming. Recent research efforts have been made to enhance and streamline the screening process. Innovative techniques that preserve the benefits of SELEX while simplifying the screening process for improved efficiency have been introduced and reviewed extensively (refer to reference 133).

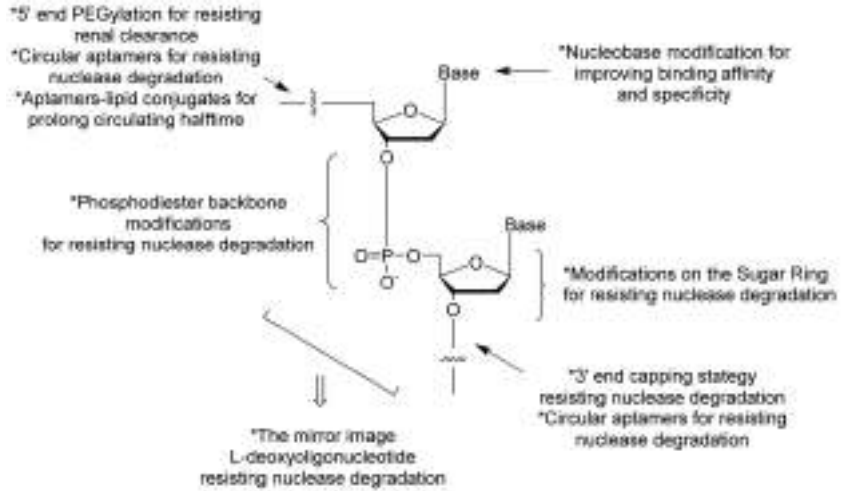


Figure 1.3: *Common strategies in the chemical modifications of nucleic acid aptamers and their purposes. Taken from reference 134.*

Considerable progress on aptamers modifications has been made by the company *SomaLogic*, which uses chemical modifications on bases to give aptamers more structural diversity and more robust target binding ability. Their Slow Off-rate Modified Aptamers (SOMAmers) demonstrate enhanced binding affinities and kinetics compared to traditional aptamers.

By substituting dT bases in oligonucleotide libraries with a modified dU base at the 5-position of the heterocyclic base, this approach has led to the discovery of many new aptamers for targets that were previously unselectable. Also, by incorporating various hydrophobic groups (benzyl, naphthyl, and indole) at the 5-position, researchers have expanded the range of potential targets that can be effectively bound and identified[134].

Additionally, modifications on the sugar ring, including 2'-F-ribose, 2'-NH2-ribose, 2'-OMe-ribose, or Locked Nucleic Acids (LNAs) that bridge the 2' and 4'-ribose positions covalently, have been introduced by incorporating unnatural nucleotides into oligonucleotides using the mutational T7 RNA polymerase, enzymatically. All these techniques

effectively improve the stability against nucleases and prolong serum half-life[135–137].

Phosphate linkage modifications can also be introduced into aptamers for stabilizing the chains of nucleic acids by replacing conventional phosphate backbones with sulfur-containing phosphate ester bonds[138–140], including phosphorothioate and phosphorodithioate bonds[141,142].

Kimoto *et al.* reported the incorporation of up to three unnatural nucleotides with the 7-(2-thienyl)imidazo- [4,5-b]pyridine (Ds base) nucleotides into an oligonucleotide library. The resulting DNA aptamers against vascular endothelial cell growth factor-165 and interferon-γ had K_d values of 0.65 pM and 0.038 nM, respectively, with more than a 100-fold improvement in affinities compared to the aptamers containing natural bases[143].

The invention of mirror image aptamers or *spiegelmers* was a creative approach to bypass the degradation of nucleases in the body. To identify spiegelmers, conventional aptamers with a D-configuration are generated to target a mirror image of the desired biological target. The identified aptamers' sequence is then produced in its respective mirror-image configuration using non-natural L-nucleotides. Guided by symmetry principles, the resulting L-aptamers bind to the natural target with the same affinity as the D-aptamers bind to the mirror-image target[144].

Other modification strategies to protect aptamers against exonucleases use the 3'-3' and 5'-5' capping methods with an inverted nucleotide in the terminus[145] or even the cyclization of nucleic acids by linking 5'- and 3'-termini[146].

Despite all synthetic efforts and achievements to make aptamers more stable, there is an inherently limited chemical diversity compared to antibodies, which makes it more challenging to find high-affinity binders for some targets, especially those lacking well-defined binding pockets. Furthermore, the SELEX methodology cannot guarantee that the best binders have been selected (due to the sequence space's combinatorial complexity) and can result in batch-to-batch variability, where each round of selection yields slightly different sequences and affinities. This can complicate manufacturing and hinder regulatory approval.

While modified nucleotides can improve nuclease resistance, aptamers and SOMAMERs still generally have more limited serum stability than antibodies and while less immunogenic, they can still elicit immune responses, especially for therapeutic applications. Modifying nucleotides and pegylation helps reduce immunogenicity but does not eliminate it.

1.4 . BACE1 as molecular target in the Alzheimer's Disease

Presently, more than 50 million individuals globally have dementia. This number is predicted to nearly double every 20 years, reaching 82 and 152 million in 2030 and 2050, respectively. Most of this rise is expected to occur in developing nations, where 60% of dementia patients currently reside. However, by 2050, this percentage is projected to increase to 71%[147]. Roughly 60 to 70% of dementia cases worldwide are attributed to AD, affecting approximately 35 million individuals.

This neurodegenerative condition is linked to aging and is experienced by 15% of those

aged 80 years or above. This situation escalates in developed and emerging nations and is among France and Cuba's six primary causes of death. The global cost of dementia, as estimated by the World Health Organization (WHO), is $604 billion annually, surpassing the combined expenses of cancer, cardiovascular disease, and stroke[148,149].

AD dementia is preceded by a pre-clinical phase that may last for 15 to 20 years and a prodromal period that persists for 3 to 6 years before the onset of dementia. Primary symptoms include difficulty remembering information that interferes with everyday activities, struggles with problem-solving and planning ahead, finding it hard to complete previously familiar tasks (whether at home, work, or in leisure activities), feeling confused about the time or place, having difficulty recognizing visual images and spatial relationships, new difficulties with expressing verbally or in writing, demonstrating poor judgment or decision-making skills, withdrawing from social or work-related activities, and changes in mood and personality[150].

In a comprehensive report of 2022, 143 agents were reported in 172 trials assessing new therapies for AD: 31 agents in Phase 3 trials, 82 in Phase 2, and 30 in Phase 1. Figure 1.4 shows all pharmacologic compounds currently in clinical trials for AD.

The most common agents being studied are Disease-modifying Therapy (DMT) (119 agents; 83.2% of the total number of agents in trials); 24 (16.8%) are symptomatic agents, including 14 (9.8% of all agents in trials) targeting cognitive enhancement and 10 (6.9% of all agents in trials) intending to treat neuropsychiatric and behavioral symptoms. Of the DMTs, 40 (33.6% of DMTs) are biologics and 79 (66.4% of DMTs) are small molecules. Twenty (16.8%) DMTs have amyloid, 13 (10.9%) have tau, 23 (19.3%) have inflammation, and 19 (16%) have synaptic plasticity/neuroprotection as their primary mechanistic targets. Considering DMTs only, 21 (67.8%) of Phase 3 agents are DMTs; 71 (86.6%) Phase 2 drugs are DMTs; and 27 (90%) Phase 1 agents are DMTs. There are 53 repurposed agents in the pipeline comprising 37.1% of the candidate therapies (all phases combined)[151].

Some molecular and neuropathological manifestations of AD include impairment of N-methyl-D-aspartate receptor-related signaling pathways, intracellular accumulation of hyperphosphorylated tau protein, extracellular deposition of amyloid-beta, oxidative stress, metal ion metabolism disorders, abnormalities of lipid metabolism, and disturbances[152]. AD 's two essential pathological aspects are related to forming plaques of Amyloid-beta (Aβ) peptides and tangles of tau proteins.

Although the physio-pathology of AD is not fully known, the currently most accepted model and the one investigated with a therapeutic objective is based on the amyloid hypothesis[153]. The accumulation of Aβ peptides in the brain tissue constitutes the starting point for the characteristic pathological changes of the disease. From the physiological point of view, the formation of amyloid plaques would give rise to synaptic dysfunction that leads to dementia.

At the molecular level, the joint action of BACE1 and γ secretases on Amyloid Precursor Protein (APP) produces Aβ peptides of variable lengths (see Figure 1.5), but mostly between 39 and 42 amino acids (Aβ-39, Aβ-40, Aβ-42). These peptides

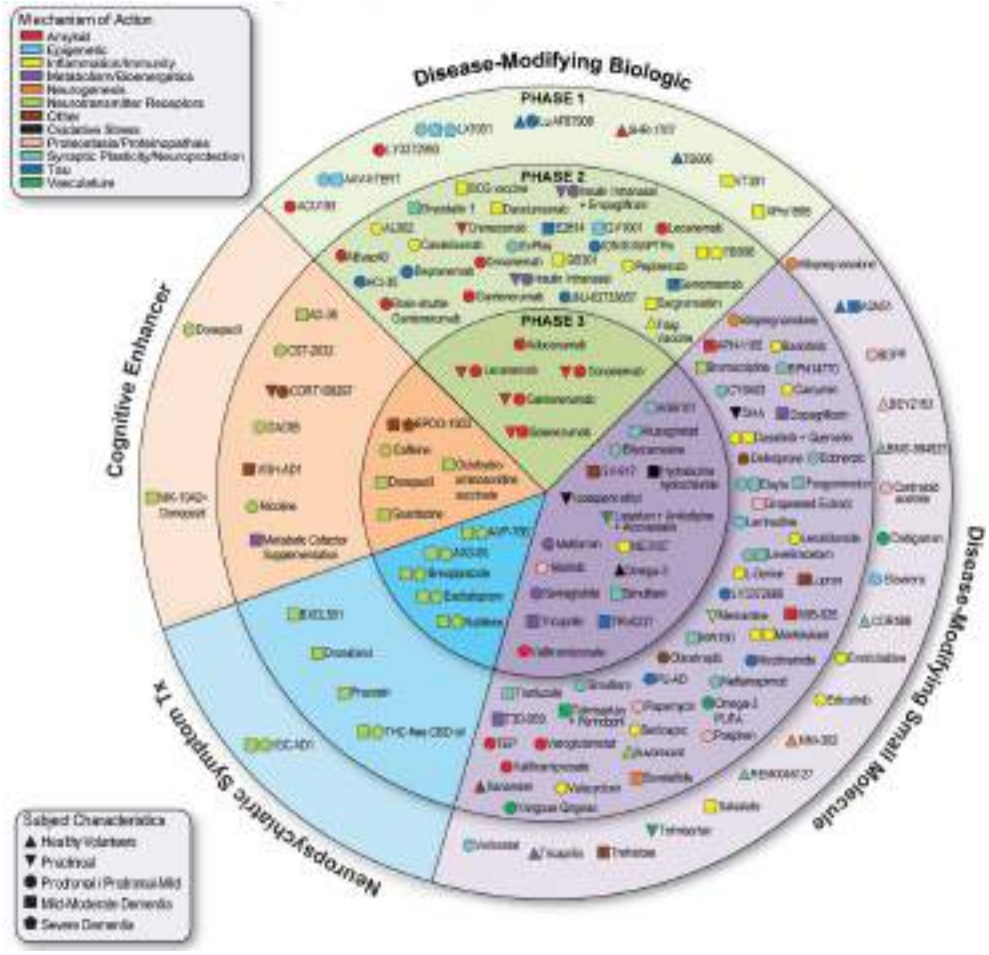


Figure 1.4: *The agents are categorized by phase of the clinical trial (Phase 1, 2, or 3) and by type of pharmacological compound (biologics, disease-modifying small molecules, or symptomatic agents). The shape of the icon represents the population of the trial, and the color represents the Common Alzheimer's Disease Research Ontology (CADRO)-based class of the agent. The "Other" category includes CADRO classes with three or fewer agents in trials. Agents that are new to the pipeline since 2020 are underlined. Taken from reference 151.*

can aggregate to form oligomers called amyloid fibrils, among which certain aggregated forms of Aβ42 have been structurally characterized[154]. Therefore, many drug candidates targeted at β- or γ-secretase have been developed to treat AD.

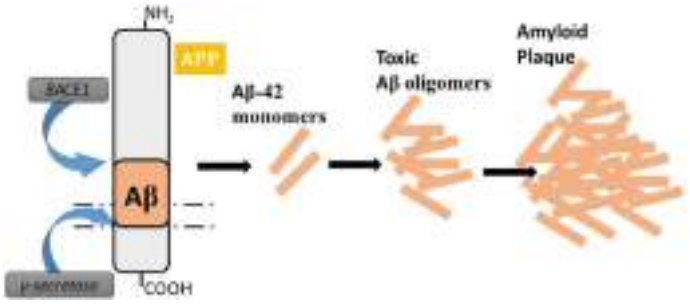


Figure 1.5: *The amyloidogenic pathway of Amyloid Precursor Protein (APP) involves the sequential cleavage of APP by β-secretase.The remaining fragment is then cleaved by γ-secretase, resulting in the formation of the Aβ peptide. Due to its high propensity to aggregate, Aβ peptide oligomerizes, accumulates, and forms amyloid senile plaques leading to the documented alterations in Alzheimer's disease.Taken from reference 155.*

γ-secretase inhibitors and modulators were the first to be tested in clinical trials. However, all the trials have had to be halted due to lack of efficacy or sometimes severe side effects[156]. γ-secretase has many other biological substrates that could explain the adverse effects, and therefore, the focus turned to β-secretase inhibitors[157].

There is considerable evidence regarding the involvement of BACE1 in AD pathogenesis. Increased protein levels and activity of BACE1 have been reported in the normal aging brain and to an even more considerable extent in the AD brain[158–160]. In addition, a mutation in APP resulting in increased BACE1 cleavage (called the Swedish mutation) results in increased Aβ production and a familial form of AD (FAD)[161]. On the contrary, the so-called Icelandic mutation in APP[162] alters one amino acid at the BACE1 cleavage site of APP, reducing the ability of BACE1 to cleave APP by about 30%[163], is strongly protective against AD. Therefore, various small molecules inhibiting BACE1 were developed and brought to clinical trials.

Clinical trials of BACE1 inhibitors for treating AD patients have yielded disappointing results. Despite their ability to reduce Aβ plaque load[164,165] in both animals and humans[166,167], they have not demonstrated improvements in cognitive function[164,165], so they have been discontinued during phase II or III trials due to a lack of efficacy, side effects, or both, including cognitive decline[167].

Despite the recurrence of failure observed in clinical trials of BACE1 inhibitors, recent studies have shed light on potential explanations for their negative impact on synaptic transmission. It has been suggested that the adverse effects could be attributed to the high concentrations of inhibitors used in those studies. Conversely, low-dose BACE1 inhibition, which results in a moderate reduction (30-50%) of Aβ peptides, has shown no significant impact on synaptic transmission[168].

In the pursuit of improving cognition in AD, various strategies have been explored,

including BACE1 inhibition and immunotherapy aimed at clearing brain amyloid plaques or neurofibrillary tangles[157,169]. Recent research has highlighted previously overlooked benefits associated with targeted BACE1 inhibition in microglia. These benefits include enhanced amyloid clearance and improved cognitive performance[170,171].

1.4.1 . BACE1 and BACE2 structural similarities

Similar to BACE1 (52% sequence identity and 68% sequence similarity[172]), BACE2 is a type I transmembrane protein that belongs to the peptidase A1 family (also called the pepsin family) of aspartyl proteases. Unlike BACE1, which is highly expressed in the brain, BACE2 is more prominently found in peripheral tissues (colon, kidney, and pancreas)[173]. There are hints, however, indicating that side effects caused by BACE2 cross-inhibition in the brain may become apparent upon inflammation events characteristic of the AD[174].

BACE1 and BACE2 (referred as BACE-X from now on) are close homologs that share sequence similarity and highly similar 3D structures. Mirsafian *et. al.* conducted an evolutionary trace study for the structural comparison of BACE-X[21]. They performed the superposition of the crystal structures of BACE1 (PDB ID: 1FKN) and BACE2 (PDB ID: 2EWY) yielded an RMSD of 1.46 Å over 373 C-alpha atoms, indicating that these structures are remarkably similar to each other. The researchers also identified 123 group-specific residues (present only in one of the two proteins) accounting for 24.5% of human BACE1 and 23.7% of human BACE2).

The BACE-X active site comprises the catalytic aspartic acid dyad residues, the flap region, and 10s loop (see Figure 1.6). The amino acids within the active sites of BACE-X are very well-conserved at greater than 80% identity, making the design of selective BACE1 or BACE2 inhibitors exceptionally challenging[175]. Interestingly, there are group-specific residues in this region also; Pro70, Ile110, Ile126, and Asn233 of BACE1 substituting Lys86, Leu126, Leu142, and Leu246 of BACE2, respectively. These four residues are expected to play a pivotal role in determining the selectivity of enzymatic activity, especially Pro70, since the proline's cyclic property may affect the flap region's flexibility.

The structural similarity of these two enzymes is at the root of the off-target effect they exhibit. *Off-target inhibition* is a term that describes the effects that can occur when a drug binds to targets (proteins or other molecules in the body) other than those for which the drug was meant to bind. This can lead to unexpected potentially harmful side effects[4]. The majority of BACE1 inhibitors in past and current clinical trials do not show significant selectivity for BACE1 over BACE2. To prevent such potential undesired physiologic effects, selectivity for BACE1 over BACE2 has been targeted by several companies[172,176].

The activation and inhibition of BACE1 exhibit a distinctive feature characterized by its sensitivity to pH levels. Results from fluorescence experiments reveal that the peptide cleavage activity of BACE1 is highly specific to a narrow pH range, with optimal activity occurring at pH 4.5. However, this activity sharply decreases when the pH falls below 4 or rises above 5.9, as demonstrated by the same experiments[177,178].

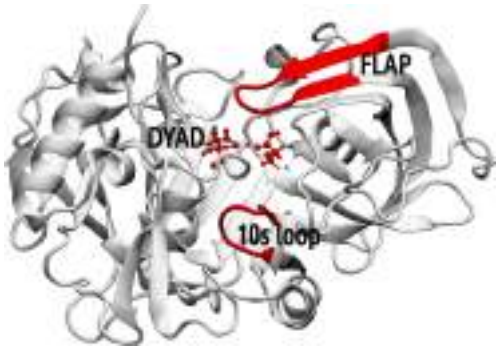


Figure 1.6: *The binding pocket of BACE1. Three important parts are highlighted in red; the flap region (the most flexible part of the binding site controlling the access of substrates), the catalytic aspartic acid dyad (crucial for the proteolytic activity of BACE1), and the 10 seconds loop (10s loop).*

The enzyme's catalytic action involves the following steps (see Figure 1.7): (i) the substrate binds to the enzyme's active site, and the catalytic dyad activates a water molecule by forming a hydrogen bond, (ii) the activated water molecule makes a nucleophilic attack on the scissile carbonyl in the peptide, leading to the formation of a diol intermediate. This intermediate is stabilized by forming hydrogen bonds with the carboxyl group of aspartates in the catalytic dyad. (iii) Ultimately, a proton is transferred from Asp to the leaving amino group, resulting in the peptide bond cleavage[179].

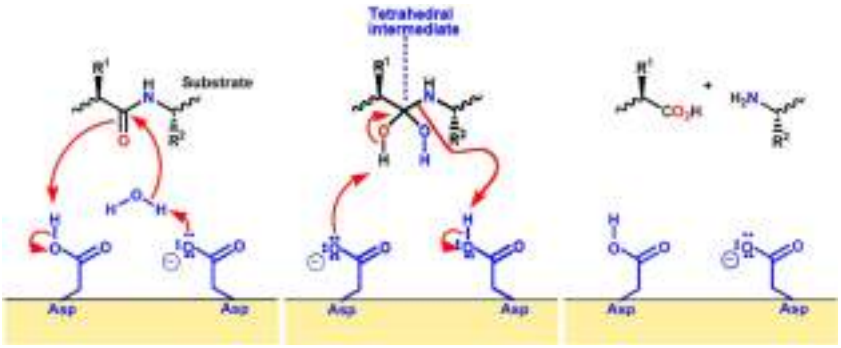


Figure 1.7: *Mechanism of action of the catalytic aspartic acid residues of BACE1. Taken from reference 180.*

The intricate mechanism behind BACE1's pH-regulated enzymatic activity and peptide-inhibitor binding has been explored through computational studies. Contrary to previous assumptions that dehydration at low pH was responsible for inactivation, Constant pH Molecular Dynamics (CpHMD) simulations demonstrate that the active site maintains hydration but assumes a self-inhibited state involving Tyr71 flap residue[181]. This finding is consistent with an earlier structural report that identified eight conserved water molecules in the BACE1 active site, five of which were conserved in 90% of 153 studied structures and the remaining three in 70-80% of the structures[182].

1.5 . Mathematical background

It is natural for methodological contributions in computational chemistry, to integrate diverse mathematical concepts. This is because mathematics offers a rigorous framework for formulating models, analyzing data, and making predictions. In this section, we aim to provide a clear understanding of those key definitions necessary to comprehend the algorithms we propose in Chapters 4 and 5.

1.5.1 . Big-O notation

During the software development process, it is essential to evaluate how well a computer program performs in various potential scenarios. The number of fundamental operations an algorithm executes as a function of the length of its input will typically be used to assess its computational efficiency.

A function T from the set $\mathbb{N}$ of natural numbers to itself can be used to determine an algorithm's efficiency if $T(n)$ is equal to the number of most basic operations the algorithm can perform on inputs of length n. However, the low-level specifics of what a fundamental operation is, can sometimes make this function T overly dependent on them. The well-known Big-O notation helps to ignore these low-level details and concentrate on the big picture[183], allowing to classify algorithms according to how their run time or space requirements grow as the input size grows (a.k.a asymptotic analysis of function growth).

A formal definition of Big-O can be stated as follows: If f, g are two functions from $\mathbb{N}$ to $\mathbb{N}$, then we say that $f = O(g)$ if there exists a constant c such that $f(n) \leq c * g(n)$ for every sufficiently large n. By using the Big-O notation, we ignore (i) behavior on small inputs (because programs typically run fast enough on small test cases), and (ii) multiplicative constants (because they are extremely sensitive to details of the implementation, hardware platform, *etc.*)[183].

Assume that $f(n) = n$ and $g(n) = n^2$. n^2 is smaller for low positive input values. They have the same value for input 1, but then g grows and rapidly diverges to become significantly larger than f. We will only be interested in the faster-growing term when a function is the sum of faster and slower-growing terms. For instance, n^2 will be equal to $n^2 + 7n + 105$. The term with the fastest growth controls the behavior of the function as the input n increases (the first term in this case)[184].

Some familiar order of growth in computer science are illustrated in Figure 1.8. Constant growth is represented by $O(1)$; linear growth is $O(n)$; logarithmic growth is $O(logn)$; log-linear growth is $O(nlogn)$; polynomial growth is $O(n^k)$; exponential growth is $O(k^n)$; factorial growth is $O(n!)$. These growths can be compared from best to worst as follows:

$$O(1) < O(logn) < O(n) < O(nlogn) < O(n^k) < O(k^n) < O(n!)$$

1.5.2 . Basics of graph theory

Graph Theory is a branch of mathematics devoted to studying abstract objects called graphs, which represent numerous situations in which several elements are mutually related. Applications of graphs are diverse and widespread. Much of this area's success is

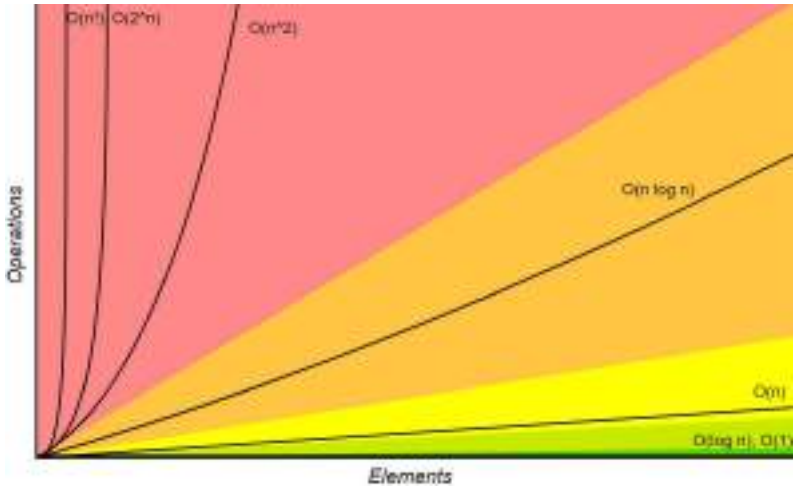


Figure 1.8: *Some familiar algorithms' order of growth. Taken from reference 185.*

due to the ease at which ideas and proofs may be communicated pictorially in place of, or in conjunction with, the use of purely formal symbolism [20].

A **graph** $G = (V, E)$ is a pair of a set of **vertices** (*a.k.a* **nodes**) V and a set of edges E. Each **edge** is a two-element subset of V and denotes the adjacency between the nodes it connects. In Figure 1.9A, $V = \{1, 2, 3, 4, 5, 6, 7\}$ and $E = \{\{1, 3\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{3, 5\}, \{3, 6\}, \{4, 5\}, \{4, 6\},$

Two connected nodes are called **neighbors**, and the number of neighbors of a given node constitutes its **degree**. In Figure 1.9A, $\{2, 3, 5, 6\}$ are neighbors of 4, whose degree is 4.

A **path** is a non-empty graph $P = (V, E)$ of the form $V = x_0, x_1, ..., x_k$, $E = x_0x_1, x_1x_2, ..., x_{k-1}x_k$, where the x_i are all distinct. The vertices x_0 and x_k are linked by P and are called its **ends**; the vertices $x_1, ..., x_{k-1}$ are the **inner vertices** of P. The number of edges of a path constitutes its **length**. In Figure 1.9A, nodes $\{1, 3, 2, 4, 6\}$ form a path of length 3 with ends $\{1, 6\}$ and inner vertices $\{3, 2, 4\}$.

An edge with identical ends is called a **loop**, and an edge with different ends is a **link**. Two or more links with the same pair of ends are said to be **parallel edges**. In Figure 1.9A, node 6 has a loop and nodes $\{3, 5\}$ are the ends of two parallel edges.

A graph is **simple** if it has no loops or parallel edges. A **complete graph** is a simple graph in which any two vertices are adjacent. Figure 1.9B represents a simple graph which is also complete.

A graph G is **connected** if for any two vertices a and b there is a path from a to b. Connectivity of simple graphs can be represented using its **adjacency matrix**, a symmetric matrix M in which $M_{ij} = 1$ if nodes i and j are connected and $M_{ij} = 0$ otherwise. If there is no directionality in the definition of the edges and no data associated with them, it is said that the graph is **undirected** and **unweighted**, respectively. Figure 1.9C exemplify the adjacency matrix of the connected, undirected, and unweighted graph of Figure 1.9B.

If $P = x_0...x_{k-1}$ is a path and $k \geq 3$, then the graph $C := P + x_{k-1}x_0$ is called a **cycle**.The **length of a cycle** is its number of edges (or vertices). Notice that if a

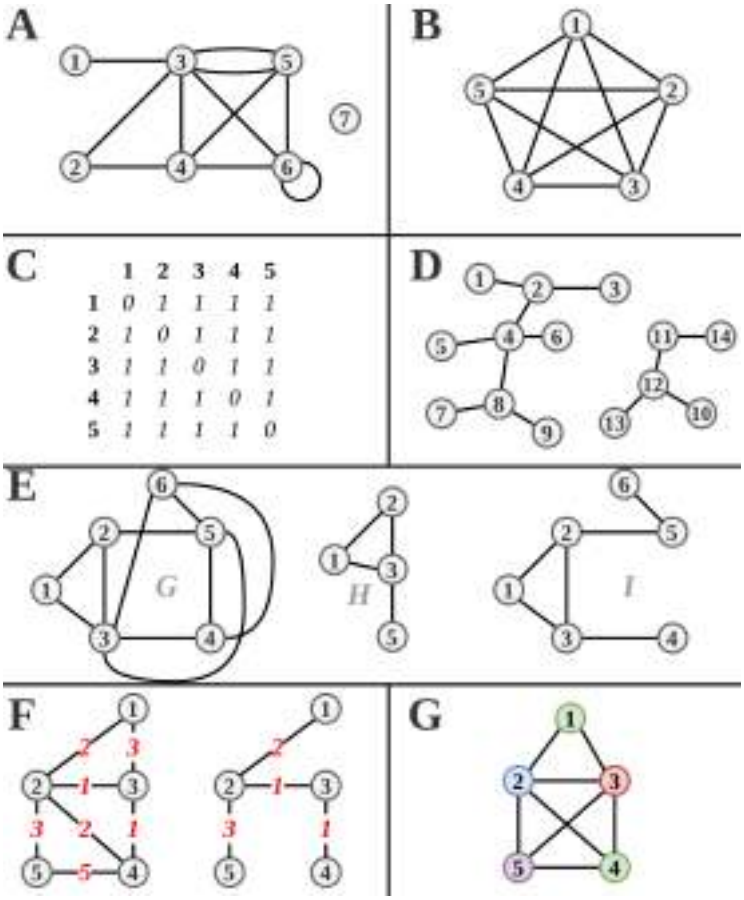


Figure 1.9: *Illustration of basic concepts of graph theory.*

connected graph contains a cycle, removing an edge from the cycle will not disconnect the graph. Nodes $\{2, 3, 4\}$ of graph in Figure 1.9A make a cycle of length 3.

An **acyclic graph**, one not containing any cycles, is called a **forest**. A connected forest is called a **tree**. Thus, a forest is a graph whose components are trees. The vertices of degree 1 in a tree are its **leaves**, the others are its **inner vertices**. Figure 1.9D depicts a forest formed by two trees.

A **sub-graph** of a graph G is a graph H such that every vertex of H is a vertex of G, and every edge of H is an edge of G also. In other words, $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. In Figure 1.9E, the graph H is a sub-graph of the graph G.

A **spanning sub-graph** of a graph G is a sub-graph obtained by edge deletions only, in other words, a subgraph whose vertex set is the entire vertex set of G. If S is the set of deleted edges, this sub-graph of G is denoted $G \setminus S$. Observe that every simple graph is a spanning sub-graph of a complete graph. In Figure 1.9E, the graph I is a spanning sub-graph of G.

A **sub-tree** of a graph is a sub-graph that is a tree. If this tree is a spanning sub-graph, it is called a **spanning tree** of the graph. A connected spanning subgraph of minimum

weight is called **minimum spanning tree**. The Figure 1.9F represents a weighted graph (left) and its minimum spanning tree (right).

Algorithms for **graph search** (or **graph traversal**) examine a graph for broad discovery or explicit search and are typically used as a foundation for additional procedures. They will attempt to visit as much of the graph as possible, but there is no assumption that the pathways they take will be computationally optimal.

Depth-first search (DFS) is a tree-search in which the vertex added to the tree T at each stage is a neighbor of the most recent addition to T as possible. In other words, we first scan the adjacency list of the most recently added vertex x for a neighbor not in T. If there is such a neighbor, we add it to T. If not, we backtrack to the vertex added to T just before x, examine its neighbors, and so on. The resulting spanning tree is called a **depth-first search tree** or **DFS-tree** (see Figure 1.10)).

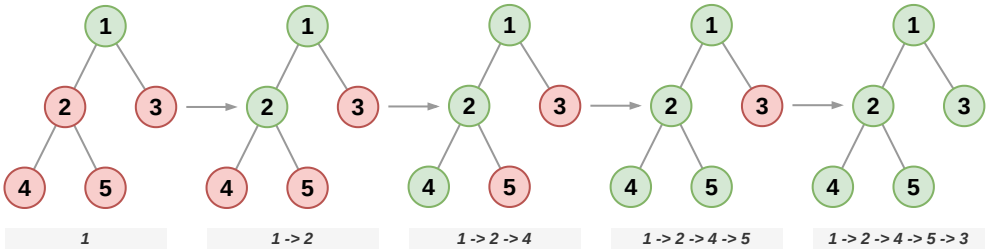


Figure 1.10: *Graphical description of the depth-first search order.*

A **clique** is a sub-graph in which vertices are all pairwise adjacent. If a clique is not contained in any other clique, it is said to be **maximal**, while the term **maximum clique** denotes the maximal clique with a maximum number of nodes (maximum **cardinality**). The Maximum Clique Problem (MCP) deals with the challenge of finding the maximum clique inside a given graph. In Figure 1.9G, nodes $\{1, 2, 3\}$ denote a maximal clique. Similarly, nodes $\{2, 3, 4, 5\}$ also denote a maximal clique that turns out to be the maximum clique of the graph. Graph in Figure 1.9B is also a clique.

A central idea of MCP algorithms is the notion of **vertex coloring**. A **proper vertex coloring** refers to assigning a particular color (or any other unique label) to each vertex of a graph so that adjacent vertices do not share the same color. The **Vertex Coloring Problem** consists of finding a proper coloring that uses the fewest number of colors, known as the graph's **chromatic number** (χ). It is common to employ coloring techniques because χ is an upper bound to a graph's maximum clique size. This property is exploited to discard impossible solutions and guide the search for cliques. Graph in Figure 1.9G has $\chi = 4$.

In computer science, there exists a category of problems known as **NP-complete**. These problems are characterized by the property that no known algorithm can solve them exactly in polynomial time. The MCP and the Vertex Coloring Problem are examples of NP-complete problems.

While finding an exact solution to these problems is computationally infeasible for large instances, there are approximate algorithms and heuristics that can provide rea-

sonable solutions within a reasonable amount of time. **Heuristic** methods utilize logical assumptions and techniques to satisfactorily solve complex problems.

Despite this limitation, heuristics are extensively utilized in practical applications where a slight deviation from the optimal solution is acceptable and does not significantly impact the overall outcome. These heuristics offer a practical trade-off between computational efficiency and solution quality, enabling us to effectively address real-world instances of NP-complete problems.

1.5.3 . Similarity measures

The term similarity measure refers to a function used for comparing objects of any type. Thus, the input of a similarity measure is two objects, and the output is generally a number between 0 (entirely dissimilar) and 1 (identical). Similarity is related to distance, which is its inverse. A similarity of 1 implies a distance of 0 between two objects. The selection of the proper similarity function is a critical parameter in many applications. For example, in molecular clustering (see Section 1.6), among the most common analyses that explicitly rely on molecular similarity, one is interested in grouping molecules conformations based on their geometrical similarity.

There are two major classes of similarity functions: metric functions and non-metric functions. In order for a function d to be a metric it has to satisfy all the following three properties for any objects X, Y, Z:

1. $d(X,Y) = 0 \; if \; X = Y$ (identity axiom)
2. $d(X,Y) = d(Y,X)$ (symmetry axiom)
3. $d(X,Y) + d(Y,Z) \geq d(X,Z)$ (triangle inequality)

Metric similarity functions are very widely used in search operations because of their support of the triangle inequality. The triangle inequality can help prune a lot of the search space, by eliminating objects from examination that are guaranteed to be distant to the given query. The most frequently used metric similarity function is the Euclidean distance.

For two objects X and Y that are characterized by set of n features $X = (x_1, x_2, \ldots, x_n)$ and similarly $Y = (y_1, y_2, \ldots, yn)$ the Euclidean distance is defined as

$$D = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{1.17}$$

When comparing molecular structures, however, the Root Mean Square Deviation (RMSD) is the most universally employed similarity function. It is computed as the average distance between every pair of equivalent positions (Equation 1.18). When molecules are aligned first (to minimize the RMSD value by ignoring translation and rotations), this function is referred to as optimal RMSD ($\mathrm{RMSD}_{\mathrm{opt}}$).

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} {d_i}^2}{N}} \tag{1.18}$$

Despite its practical utility, the RMSD has punctual drawbacks already described[186], being the most remarkable its inherent difficulty in characterizing very flexible zones of a molecular structure[187]. Although some authors have proposed normalized and weighted schemes to work with RMSD[188,189] that could alleviate these inconveniences, they are more computationally demanding alternatives less frequently employed.

Euclidean or Euclidean-like metrics are exploited in molecular similarity comparison applications under the assumption that they are faster to compute than the optimal RMSD (as fewer unitary operations are involved and also because no alignment is performed between every pair of structures). An example of this philosophy is the software qtcluster of the ORAC suite (see Section 1.6) that employs the maximum difference between corresponding pairs of atoms (Equation 1.19) to compare molecules. Under this metric, the similarity of two elements S_m and S_n is assessed by the absolute maximum value of the difference between their inter-atomic distances.

$$d_{S_m,S_n} = max_{i,j}|d_{ij}(S_m) - d_{ij}(S_n)| \tag{1.19}$$

Clustering results of two different procedures can be also compared using several indices or scores from which the Adjusted Rand Index (ARI) stands out. Let's consider an MD trajectory T as a set of N elements (frames) $T = \{t_1, t_2, ..., t_N\}$. The outcome of applying a given clustering algorithm on T is a partition P of the N objects into C clusters, $P = \{p_1, p_2, ..., p_C\}$, such that the union of all the subsets in P is equal to T and the intersection of any two subsets in P is empty.

Considering $\binom{N}{2} = N(N-1)/2$ as the total number of element pairs (t_i, t_j) in T, there exist four classifications of pairs when comparing Q and B; (a) elements in a pair are placed in the same group in Q, and the same group in B (true positives), (b) elements in a pair are placed in the same group in Q, and different groups in B (false negatives), (c) elements in a pair are placed in the same group in B, and different groups in Q (false positives), and (d) elements in a pair are placed in different groups in Q and B (true negatives). It is possible to assess the equivalence between Q and B based on the number of pairs of elements lying in any of these four categories.

The Rand Index (RI)[190] (Equation 1.20) expresses the fraction of pairs of elements on which two clusterings coincide (from 0 for unrelated to 1 in a perfect match). However, RI approaches its upper limit as the number of clusters increases because d tends to grow even for poorly related partitions, giving a high score.

$$\mathrm{RI} = \frac{a+d}{a+b+c+d} \tag{1.20}$$

An Adjusted Rand Index (ARI)[191,192] corrected against "agreements-by-chance" has been extensively used (Equation 1.21) to measure the correspondence between partitions created by clustering algorithms. ARI extends from non-bounded negative values (poorly related partitions) to 1 (highly similar partitions).

$$\mathrm{ARI} = \frac{\binom{N}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{N}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]} \tag{1.21}$$

The Tanimoto Index (TI), also known as the Tanimoto coefficient or the Jaccard index, is a commonly employed similarity measure for comparing the diversity of sample sets[193]. It is defined as the ratio of the size of the intersection of two sets divided by the size of their union. The TI ranges from 0 to 1, with 0 indicating no similarity between two sets and 1 indicating identical sets. Typically, a TI greater than 0.7 is considered a good indicator that two molecules or samples are similar though this thresholds can vary based on the data and application. The TI is popular due to its simple formulation, bounded range, and interpretability, providing a straightforward quantitative measure of similarity for binary sample sets.

1.5.4 . Essential data structures

Data structures can be defined as the organization of data in a logical or mathematical model. It must be simple enough that one can effectively process the data when necessary. The data present in the data structure are processed using several operations like traversing (access to each element present in the data structure exactly once), searching (finding the location of a particular element with a key attribute), inserting (adding new elements), and deleting (removing old elements)[194].

Below we briefly describe a set of data structures that were relevant in the development of the algorithms we designed in Chapters 4 and 5).

An **array** is an ordered collection of elements of the same type, the number of elements being fixed unless the array is *flexible*. Each element in an array is distinguished by a unique list of *index* values that determine its position in the array. Each index is of a discrete type. The number of indices is fixed, and the number and ordering of the indices determines the dimensionality of the array[195].

A one-dimensional array, or **vector** (v), consists of a list of elements distinguished by a single index. If v is a one-dimensional array and i is an index value, then v_i refers to the i^{th} element of v. If the index ranges from L through U then the value L is called the lower bound of v and U is the upper bound. Usually in mathematics and often in mathematical computing the index type is taken as integer and the lower bound is taken as one[195].

In a two-dimensional array, or **matrix**, the elements are ordered in the form of a table comprising a fixed number of rows and a fixed number of columns. Each element in such an array is distinguished by a pair of indexes. The first index gives the row and the second gives the column of the array in which the element is located. The element in the i^{th} row and j^{th} column is called the i, j^{th} element of the array[195]. Of specially interest for our work are bit-arrays (bit-vectors and bit-matrices) in which the type of the element are bits.

A **heap**[196] is made up of nodes that contain values. A typical heap has a root node at the top, which may have two or more child nodes directly below it. Each node can have two or more child nodes, which means the heap becomes wider with each child node. When displayed visually, a heap looks like an upside down tree and the general shape is a heap.

While each node in a heap may have two or more *children*, most heaps limit each

node to two children. These types of heaps are also called *binary heaps* and may be used for storing sorted data. For example, a *binary min heap* stores the lowest value in the root node. The second and third lowest values are stored in the child nodes of the root node. Throughout the tree, each node has a greater value than either of its child nodes. A "binary max heap" is the opposite.

Space partitioning is the process of dividing a space (usually a Euclidean space) into two or more non-overlapping regions. Any point in the space can then be identified to lie in exactly one of the regions. A k-dimensional tree (KD-TREE)[197] (see Figure 1.11) is a space-partitioning data structure for organizing points in a k-dimensional space. A KD-TREE is constructed through iterative bisections of the input data along a single coordinate. These cuts are made at points producing a maximum spread in the selected coordinate's distribution.

Every non-leaf node in the tree acts as a hyperplane, dividing the space into two partitions. This hyperplane is perpendicular to the chosen axis, which is associated with one of the k dimensions. There are different strategies for choosing an axis when dividing, but the most common one would be to cycle through each of the k dimensions repeatedly and select a midpoint along it to divide the space. For instance, in the case of 2-dimensional points with x and y axes, we first split along the x-axis, then the y-axis, and then the x-axis again, continuing in this manner until all points are accounted for.

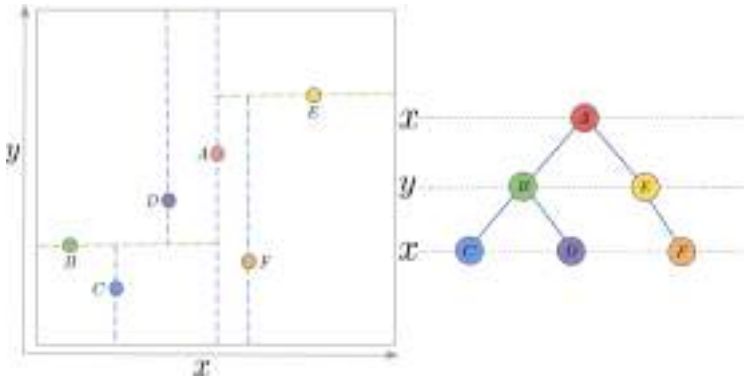


Figure 1.11: *Partition of a bi-dimensional space using a kd-tree. Taken from reference 198.*

KD-TREEs are a useful data structure for several applications, such as searches involving a multidimensional search key like nearest neighbor searches. Unfortunately, efficient usage cases of KD-TREEs are restricted to Euclidean metric spaces of low dimensionality. Such limitation prevents their utilization in the high-dimensional spaces that characterize molecular conformations.

Vantage point trees (VP-TREEs)[199] are an alternative to KD-TREEs that were conceived to work with general metrics in high-dimensional spaces. Rather than performing cuts among the coordinates values, nodes of the VP-TREE split the database into smaller subspaces employing distinctive elements known as Vantage points (VPs). By convention, near-to-vp instances constitute the left subspace, while far points are grouped into

the right subspace. The recursive partition of the input database then leads to a binary tree. In a VP-TREE, every frame has a "perspective" on the entire T via their distance to all other frames. This notion of "perspective" is a direct consequence of the triangle inequality represented in Equation 1.22 that holds for every pair of frames $(a, b) \in T$. In Equation 1.22, $d(a, b)$ denotes the distance between two points a and b, which will always be greater or equal to the absolute value of the difference between distances from p to a and b, respectively.

$$d(a, b) \geq |d(p, a) - d(p, b)| \tag{1.22}$$

Given a metric space (S, d) and some finite subset $(T \in S)$ representing a given molecular ensemble, the goal of a VP-TREE encoding is to organize T in a way that the k nearest neighbors of every query frame q, may be located faster than the naive approach of visiting all frames in T for each query. Suppose that for some frame $p \in T$, the median (μ) of the p-versus-all distances is determined. Then T can be split into two subspaces; the left subspace (or inside sphere S_{pl}), containing frames closer than μ to p, and the right subspace (or outside sphere S_{pr}), containing frames at μ or larger distance values from p (see Figure 1.12). S_{pl} and S_{pr} will have roughly the same size if there are relatively few frames that lie exactly at μ.

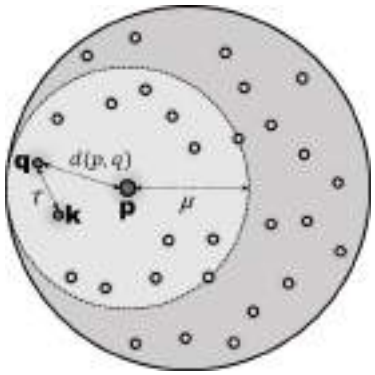


Figure 1.12: *Partition of a database via the vp p. Elements closer than μ to p form the left subspace (inside the sphere in light gray) while the rest form the right subspace (outside the sphere in dark gray). q and k denote a query point and its k^{th} nearest neighbor, respectively. τ and $d(p, q)$ represent the distances from q to k and to p respectively.*

Now suppose that for a query frame $q \in S_{pl}$ the k^{th} nearest neighbors is solicited. By only visiting frames inside S_{pl}, it is possible to define a variable τ storing the distance from q to the k^{th} neighbor found so far. The relevance of VP-TREEs comes from the fact that if $d(p, q) \geq \mu + \tau$, then the S_{pr} subspace can be safely removed from consideration as searching their elements would not lead to a $\tau \leq d(p, q)$. Similarly, if $q \in S_{pr}$ and $d(p, q) \leq \mu - \tau$, searching the S_{pl} subspace is unnecessary. In both cases, a single point's "perspective" sufficed to prune the search significantly. However, if $\mu - \tau < d(p, q) < \mu + \tau$ no such reduction is possible and the whole T must be explored. Details on the

mathematical validity of described notions is available in the fundamental publication of Yianilos on VP-TREEs[199].

1.5.5 . Bitwise operations

Bitwise operations are those that operate on a bit string, a bit array, or a binary numeral at an individual bit level. These are primitive fast actions directly supported by microprocessors that always get executed through bitwise operators. Table 1.1 contains the results of applying four binary operations on two bit arrays X and Y. As illustrated, **AND** operator only lights positions where both arrays are turned on. Inclusive **OR** light positions where at least one of two bits compared is turned on. Exclusive OR, or **XOR**, light positions where only one of two bits compared is turned on, while the complement or negation operator (**NOT**) inverts the bit values of the passed operand.

Table 1.1: *Bitwise operators logic.*

X	**Y**	X **&** Y	X | Y	X **^** Y	**~** X
		X **AND** Y	X **OR** Y	X **XOR** Y	**NOT** X
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

When properly combined, these operations may translate algorithms' steps (checking, erasing, combining, intersecting, *etc.*) into binary logic, potentially diminishing the global run times and memory consumption.

1.6 . Clustering of molecular ensembles

Formally conceptualized as an unsupervised ML technique, clustering is a ubiquitous tool across many branches of science that allows for grouping similar elements into sets called clusters. Intuitively, entities within a cluster are more similar than elements from other clusters[20,200,201]. Clustering methods play a fundamental role in extracting useful information from big datasets and are applied in numerous and diverse tasks[202–205].

In fields like computational chemistry, chemo- and bioinformatics, geometrical clustering of molecular data produced by docking, Molecular Dynamics (MD) and related simulations[206,207] is one of the most frequently found analyses. In the particular subdomain of FBDD, clustering is employed in various stages, primarily to group similar fragments or compounds based on their structural or physicochemical properties.

For example, clustering is used to create a diverse and representative fragment library by grouping similar compounds and selecting an exemplar from each cluster. This ensures a broad coverage of the chemical space and reduces redundancy in the library[208,209]. Also, after screening the fragment library against the target protein, the hits are clustered based on their structural or physicochemical properties, which helps prioritize the most promising cases for further optimization and reduces the number of redundant or similar compounds that will be progressed.

Likewise, clustering algorithms can be used to compare the binding modes of lead compounds to different proteins or targets. By identifying clusters of lead compounds that bind to multiple targets, it is possible to predict potential off-target effects and optimize the compounds to reduce their promiscuous binding. As well crucial for the FBDD is the ability to find receptor hotspots (sites having a high propensity for ligand binding[210]) that could provide insights into the most promising regions where a lead should bind. These hotspots are generally obtained as the clustering output of probe explorations on the receptor's surface[211].

1.6.1 . Types of clustering

After choosing an adequate metric that reflects the desired notion of similarity, selecting a clustering algorithm in line with the specific application is necessary. Many options are available to the user, but their results are not always analogous (as they assume different cluster definitions), and none should be taken as an all-purpose tool. Due to the inherent subjectivity associated with classification (the same set of elements can be grouped according to many different criteria), some authors consider clustering as an art[212].

Different starting criteria can give rise to diverse taxonomies of clustering algorithms. While it is challenging to define a singular, universal category that encompasses the vast array of clustering algorithms, there is a consensus within the field of bioinformatics regarding the following families: partitional clustering, hierarchical clustering, fuzzy clustering, neural network-based clustering, mixture model clustering, graph-based clustering, and consensus clustering[213].

An important distinction between types of clustering regards whether retrieved clusters are hierarchical (nested) or partitional (unnested). An algorithm is said to be **partitional** when the dataset gets divided into non-overlapping subsets (*i.e.* each element is in exactly one subset). Using this broad definition, partitional clustering could cover many clustering families.

By permitting clusters to have sub-clusters, it is possible to obtain a **hierarchical clustering**, which is a set of nested clusters that are organized as a tree. Each node (cluster) in the tree (except for the leaf nodes) is the union of its children (sub-clusters), and the root is the cluster containing all the objects. It is worth noting that a hierarchical clustering can be perceived as a sequence of partitional clusterings, which are retrievable by cutting the tree at a particular level.

Exclusive clustering algorithms assign each object to a single cluster. However, this restriction is often inadequate if elements can be reasonably placed in more than one cluster, and in such cases, an overlapping (non-exclusive) algorithm may be more appealing. Instead of performing arbitrary labeling of the object to a single cluster, overlapping clusterings place them in all the "equally good" clusters. On the other hand, a **fuzzy clustering** puts every object in every cluster and assigns them a membership score ranging from 0 (absolutely does not belong) to 1 (absolutely belongs). An additional constraint is often imposed on the fuzzy algorithm; the sum of the membership scores for each object must equal 1. Note that a fuzzy clustering result can be converted to an exclusive

clustering by assigning each object to the cluster in which its membership score is highest.

Clustering based on neural networks begins with a set of nodes, also known as neurons, that are initially similar except for some randomly initialized parameters, which cause each node to behave slightly differently. These nodes then learn from the data in a competitive manner: active nodes reinforce their proximity within certain regions while inhibiting the activities of other nodes.

Clustering based on mixture models is another significant family of clustering techniques that has gained increasing interest recently. It involves formulating a clustering kernel for each component in terms of a sampling density $p(X|\theta)$ where θ is an unknown parameter set. Compared to algorithms based on Euclidean distance, mixture model clustering often produces more meaningful results in cases where Euclidean distance-based algorithms fail, particularly for time series and categorical data sets.

Graph-based clustering arises when data is depicted as a graph, where the nodes are elements and the edges represent connections among objects. Thus a cluster is defined as a connected component; a group of nodes connected to one another, but not to nodes outside the group. Other types of graph-based clusters are also possible, for instance clique clusters.

Combining multiple clustering results, called ensemble clustering, **consensus clustering**, or cluster aggregation, has received significant attention. It has been proposed to address the inconsistency of stochastic clustering algorithms and clusterings produced with different parameters. The underlying concept of ensemble clustering is that combining various clusterings into a single consensus solution can highlight the common organization across different results. Ensemble clustering aims to produce a stable and robust final clustering by merging results that may differ due to random initialization or algorithmic variations.

1.6.2 . Molecular clustering

The upcoming sections focus on four molecular clustering algorithms that, in our view, represent some of the most widely used or promising clustering methods that have already been applied to molecular ensembles: (i) Quality Threshold (QT), covered in Section 1.6.2.1, (ii) Daura, discussed in Section 1.6.2.2, (iii) Density Peaks (DP), detailed in Section 1.6.2.3, and (iv) Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), explored in Section 1.6.2.4.

1.6.2.1 Quality Threshold

The QT clustering was initially designed for grouping gene expression patterns. First proposed by Heyer *et al.* in 1999[214], the authors intended to overcome severe limitations inherent to other available clustering algorithms like k-means, self-organizing maps, and hierarchical variants. Since then, besides its usage in clustering gene expression, QT has been employed in fields other than microbiology[215–218], and in particular to deal with molecular ensembles like MD trajectories[219,220].

This algorithm excels in cases where strongly geometrically correlated conformations

need to be returned as clusters. Several tools like VMD[221] (through its *measure cluster* command), ORAC[220] (through its *qtcluster* procedure), WORDOM[222] (through its *qt-like* option) and a standalone script contributed by Melvin *et al.* in 2016 described as *"A python implementation of the quality threshold clustering algorithm of Heyer, 1999, specialized to molecular dynamics trajectories"*[223]. These variants all assert to use clustering procedures based on QT, but the veracity of this claim is thoroughly discussed in Section 4.1.1.

If we define the diameter of a cluster C as the maximum distance between any pair of its elements (Equation 1.23), the original QT formulation can be described as follows. After the user sets a similarity threshold k (maximum diameter of clusters to be returned), one arbitrary element is selected and marked as a candidate cluster C_1. The remaining elements are iteratively added to C_1 if and only if two conditions hold: **Condition 1**- the entering frame minimizes the increase of C_1 diameter, and **Condition 2**- the diameter of C_1 does not exceed the threshold k. A second candidate cluster is formed by starting with another element and repeating the procedure. Note that all elements are made available to the second candidate cluster (*i.e.*, elements from the first candidate cluster are not discarded from consideration). This process continues for all elements n in the trajectory until C_n candidate clusters have been formed. The one with more elements is set as a cluster, removed from further consideration, and the entire process repeated until no more clusters can be discovered. In Algorithm 1 it is shown a pseudocode for this procedure.

$$diam(C_n) = max(d_{ij}) \mid \forall\ (i,j)\ \in C_n \tag{1.23}$$

Algorithm 1: Pseudocode for the QT clustering algorithm

```
qt_clustering(G, d){
for p_i ∈ G do
    flag = True
    C_i = {p_i}
    while (flag = True) AND (C_i ≠ G) do
        find p_j ∈ (G − C_i) for which diameter(C_i ∪ p_j) is minimum
        if diameter(C_i ∪ p_j) > d_c then
            flag = False
        else
            C_i = {C_i ∪ p_j}
identify set C ∈ {C_1, C_2, ..., C_|G|} with maximum cardinality
output C
call qt_clustering(G − C, d)
}
```

The crucial aspect of the above-described workflow lies in its ability to guarantee that all pairwise similarities inside a cluster will remain under the threshold k. This aspect is assured entirely by **Condition 2**. It should be stressed that **Condition 1** merely limits the size of retrieved clusters but has no impact on maintaining their collective similarity.

While being a powerful algorithm, QT is a costly solution whose naive implementation leads to a prohibitively $O(n^5)$ temporal complexity[219].

1.6.2.2 Daura

The clustering algorithm described by Daura *et al.*[224] is a fast and powerful approach to partitioning molecular datasets. Although Daura does not guarantee the collective similarity of returned clusters as QT does, it may be considered as a trade-off for systems bigger than what QT can process.

This algorithm has been implemented in the popular suites GROMACS, WORDOM, and VMD in which it may run in a slow or memory inefficient way when processing large ensembles. It is available under distinct and even confusing names. GROMACS, WORDOM and VMD packages termed this algorithm as gromos, qt-like (qt standing for quality threshold) and quality threshold respectively. In a recent contribution by Melvin and Salsbury[223], a supposedly QT clustering implementation is proposed that also turned out to be Daura (see Section 4.1.1 and 4.2.3 for a detailed discussion).

In order to find clusters in a trajectory, the Daura algorithm works as follows. The number of neighbors is determined for each element in the analyzed dataset. We denote neighbors as those pairs of frames with a distance value less than a previously specified cutoff d_c. The frame with most neighbors (and all its neighbors) gets saved as a cluster and removed from further consideration in successive steps. The process starts again from the beginning until no more clusters can be found. In Algorithm 2 it is shown a pseudocode for this clustering.

Algorithm 2: Pseudocode for the Daura clustering algorithm

```
daura_clustering(G, d){
for p_i ∈ G do
    C_i = {p_i}
    while C_i ≠ G do
        find p_j ∈ (G − C_i) for which distance(C_i ∪ p_j) ≤ d_c
        C_i = {C_i ∪ p_j}
identify set C ∈ {C_1, C_2, ..., C_|G|} with maximum cardinality
output C
call daura_clustering(G − C, d)
}
```

1.6.2.3 Density Peaks

Though a wide variety of geometrical clustering algorithms has been proposed and continuously optimized to deal with the growing size of molecular ensembles, the famous DP alternative[225] stands out for its simple yet powerful definitions. In DP, cluster centers are spotted as those elements displaying both a high density of neighbors and a relatively large distance from other high-density elements.

As it has already been pointed out[226], the previous statement fits the nature of a converged MD simulation, where relevant biological states would lie in denser regions separated by lower-density zones of transitional basins.

Despite its theoretical convenience, DP has some practical limitations that have given rise to diverse enhancement proposals (see references 227 and 228 for a review) typically addressed to one of the following aspects: (i) the robust estimation of each element's density[229,230], (ii) the selection of an adequate distance metric[231], (iii) reducing the complexity of computing the local density of each component and their distance to neighbors of higher density[232], (iv) the automatic determination of clusters centers[228,233], and (v) optimizing the process of assigning elements to clusters[227,234].

There are a few implementations of DP specifically designed to treat molecular ensembles. The *cpptraj* module of the AMBER suite[235] is equipped with an exact variant while a recent contribution has proposed Clustering based on Local density Neighborhoods (CLoNe)[226], a robust improvement of the original algorithm.

In DP formalism, cluster centers are surrounded by neighbors of lower local density and distant from any point with high local density. This simple statement rules the algorithm, described as follows when processing a molecular dataset. Two magnitudes are computed for each frame i after setting a distance cutoff (d_c); its local density (ρ_i in Equation 1.24) and its minimum distance to a neighbor of higher local density, (δ_i in Equation 1.25). Both quantities depend on the distances between data points d_{ij}. In Equation 1.24 the term $\chi(x) = 1$ if $x < 0$ or zero otherwise so this is equivalent to define ρ_i as the number of i neighbors whose distance from i is under the d_c cutoff.

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{1.24}$$

In Equation 1.25, an exception is made for the frame of maximum ρ_i, which is conventionally set to $max(d_{ij})$. Note that δ_i is significantly larger than the typical nearest neighbor distance only for those frames that are local or global maxima in ρ. Higher values of δ_i are a distinctive hallmark of cluster centers. Previous information can be condensed and visually inspected in the decision graph of the trajectory, a 2D representation of ρ versus δ in which cluster centers are spotted at higher values of these two magnitudes. After selecting cluster centers from the decision graph, each remaining frame is assigned to the same cluster as its nearest neighbor of higher ρ.

$$\delta_i = min(d_{ij}) \ : \ \rho_j > \rho_i \tag{1.25}$$

To account for the notion of noise, DP defines a boundary region for each cluster C_i consisting of frames previously assigned to C_i but being within a distance d_c from frames belonging to other clusters. The maximum density value of the boundary region is designated as ρ_b and compared to the ρ_i of every frame in C_i. If $\rho_i > \rho_b$, the frame belongs to the core region (robust assignation). Otherwise, it can be considered in the halo zone (noisy assignation). In Algorithm 3 it is shown a pseudocode for DP.

Algorithm 3: Pseudocode for the DP clustering algorithm

Require: G, d_c

▶ 1. Compute the pairwise similarity matrix
$rmsd\_matrix =$ ***calc_rmsd_matrix***(G)

▶ 2. Compute ρ values for each node
$rho\_values = \{\}$
for $p_i \in G$ **do**
    $p_i\_vector = rmsd\_matrix[p_i]$
    $rho\_values[p_i] =$ ***count_elements***$(p_i\_vector < d_c)$

▶ 3. Compute δ values for each node
$delta\_values = \{\}$
for $p_i \in G$ **do**
    $p_i\_vector = rmsd\_matrix[p_i]$
    p_i_rho = $rho\_values[p_i]$
    $p_i\_sorted =$ ***sort_elements***(p_i_vector)
    for $p_j \in p_i\_sorted$ **do**
        p_j_rho = $rho\_values[p_j]$
        if $p_j\_rho > p_i\_rho$ **then**
            $delta\_values[p_i] = rmsd\_matrix[p_i][p_j]$
    if $delta\_values[p_i] == None$ **then**
        $delta\_values[p_i] =$ ***get_max_value***$(rmsd\_matrix)$

▶ 4. Select cluster centers from the Decision Graph
$decision\_graph =$ ***plot***$(rho\_values, delta\_values)$
$rho\_cut, delta\_cut =$ ***select_cutoffs***$(decision\_graph)$
$centers =$ ***select_centers***$(decision\_graph, rho\_cut, delta\_cut)$

▶ 5. Assign remaining elements
$clusters =$ ***assign_elements***$(elements, centers)$

1.6.2.4 HDBSCAN

Density-based clustering variants represent clusters as regions of high density surrounded by noisy low-density zones. This notion, translated to the MD jargon, is the equivalent of defining a cluster as a temporary stable region of the conformational landscape. From the density-based clustering alternatives, the HDBSCAN[236] has proved one of the most robust currently accessible solutions. This method generates a complete hierarchy of the most significant and stable clusters through two intuitive parameters (easily fixable to get a pseudo-non-parametric algorithm). According to classifications discussed in Section 1.6, HDBSCAN is a hierarchical, exclusive, and partial algorithm that generates density-based clusters.

Among the HDBSCAN 's advantages described in the original paper, the following are of particular interest: (i) the ability to characterize datasets with nested clusters or

clusters of different densities (a challenging task with other variants like Density-Based Spatial Clustering of Applications with Noise (DBSCAN)[237] or DENsity-based CLUstEring (DENCLUE)[238]), (ii) the straightforward simplification of the cluster hierarchy into an easily interpretable representation of the most significant clusters, as opposed to methods like graph-skeleton based clustering (gSkeletonClu)[239], (iii) the fact of not being circumscribed to specific classes of problems, like gSkeletonClu or element sets in the real coordinate space (like DiscovEring Clusters Of Different dEnsities (DECODE)[240] or generalized Single-Linkage[241], and (iv) the non-reliance on multiple (often critical) input parameters like the mentioned algorithms and many others.

From the previous list, DBSCAN (implemented in the *cpptraj* module of the AMBER is an appealing choice for the analysis of MD trajectories. As stated by Schubert[237], it is an algorithm proven to work in practical situations that received the Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) test-of-time Award in 2014. Conceptually, HDBSCAN supersedes DBSCAN, reporting clusters over all values of the DBSCAN's distance scale parameter ϵ and finding those clusters that persist for many values of this magnitude.

Though not primarily conceived to deal with molecular ensembles, HDBSCAN has been used successfully in the conformational study of MD simulations[242,243] through a deeply optimized implementation referred to as HDBSCAN* from now on[244]. HDBSCAN*'s authors creatively addressed each major step of the original version, reducing their time complexity from $O(n^2)$ to near $O(n\ log\ n)$ in the average case. Even in the worst cases, a fast sub-quadratic time complexity of the algorithm is expected. The main steps of HDBSCAN are detailed in Algorithm 4.

Algorithm 4: Main steps for the HDBSCAN clustering algorithm

Require: G, m

1: Compute the mutual reachability distance (MRD) matrix of G (Equation 1.26)
2: Build a MST of G
3: Compute the condensed cluster hierarchy from the MST of G (Figure 1.13) using m
4: Select the most stable clusters (Equation 1.27)

HDBSCAN formally defines the density of each frame i in terms of a core distance $\kappa(i)$; the distance from i to its k^{th} nearest neighbor. Note that the chosen metric for $\kappa(i)$ can be Euclidean, RMSD, or any other selected by the user. Computing $\kappa(i)$ for every frame of the trajectory permits to effectively spread apart denser frames from noise by defining a new similarity metric, the Mutual Reachability Distance (MRD) (Equation 1.26), in which $d(i, j)$ is the distance between elements i and j in the input metric. Under MRD, dense conformations (having low $\kappa(i)$) remain at the same original distance from each other while sparser frames are "pushed" to be at least their core distance away from any other point.

$$d_{mr}(i,j) = \begin{cases} max\{\ \kappa(i), \kappa(j), d(i,j)\ \}, & i \neq j \\ 0\ , & i = j \end{cases} \tag{1.26}$$

From a graph-theoretic point of view, a molecular trajectory can be seen as a complete graph (T) in which nodes represent frames, and pairwise edges hold the MRD distance between nodes. In this scenario, creating a hierarchical divisive partition of T can proceed by setting a high threshold value ($dist$) at which to start erasing edges in a way that T would pass from a complete graph to a completely disconnected one. As this naive approach is computationally prohibitive, HDBSCAN recurs to construct a Minimum Spanning Tree (MST) whose progressive disconnection leads to the same hierarchy of components described. An MST of T is a subset of T edges that connects all T nodes (without forming cycles) with the minimum total weight.

The MST inferred from T can be progressively disconnected to produce a hierarchy of clusters. HDBSCAN introduces a parameter m that represents the minimum number of points in a component to classify it as a cluster. m allows condensing the cluster hierarchy because now, cutting an edge that produces a component with less than m points is considered a "just-loosing-elements" cluster and not an independent one.

Concretely, the MST disconnection process (Figure 1.13) proceeds in this fashion: A new magnitude λ is defined as the inverse of the MRD distance ($\lambda = 1/dist$). MST edges are sorted in increasing order of their λ value (high distances edges come first). Successive edge cutting produces two child sub-trees at each cleavage, giving rise to one of the following situations: (i) one of the child sub-trees contains m or fewer points, (ii) both child sub-trees include m or fewer points, and (iii) both child sub-trees carries more than m points. In the first situation, a component without the lost members is retained, and the split is considered spurious. No child is returned, only a shrink component. The second situation marks the MST disconnection endpoint, as no further valid components will be produced. The third case corresponds to a"true split" and effectively separates the parent component into two new independent ones.

The extraction of final clusters from the condensed hierarchy takes place according to the definition of cluster stability ($\sigma(C_i)$, Equation 1.27). First, for a given cluster C_i (see C_2 in Figure 1.13) let's define its λ_{birth} and λ_{death} as the λ values at which C_i becomes a cluster and disappears, respectively. Inside C_i, for each element e, the λ_e value denotes when e "abandons" C_i, either as a spurious or a true split (note that $\lambda_{birth} < \lambda_e < \lambda_{death}$). Then the stability of C_i is calculated through the Equation 1.27.

$$\sigma(C_i) = \Sigma_{e \in C_i}(\lambda_e - \lambda_{birth}) \tag{1.27}$$

Once $\sigma(C_i)$ of all hierarchical clusters are computed, the final step is to find a flat (non-hierarchical) set of disjoint clusters with maximum stability. To that end, the cluster tree is processed from the leaves ($C_3, C_9, C_{10}, C_8, C_5$, and C_6 in Figure 1.13) upwards. Initially, all leaves are declared as clusters. Then, the stabilities of sibling leaves i, j (sharing the same parent k) are summed, and the result is compared to the stability of their parent. If $\sigma(i) + \sigma(j) > \sigma(k)$, $\sigma(k)$ is set to $\sigma(i) + \sigma(j)$, but i and j (not k) are still

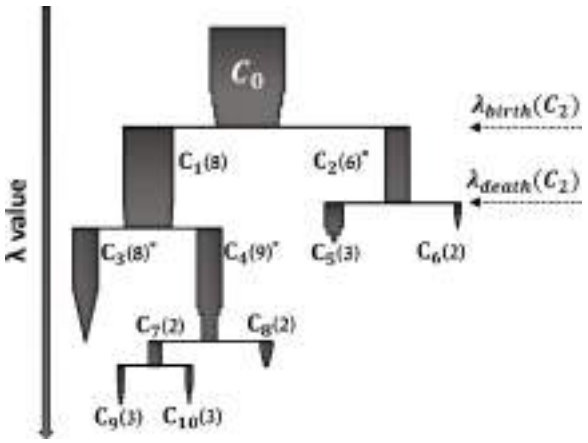


Figure 1.13: *Condensed hierarchy of clusters produced by the HDBSCAN's disconnection process. The stability of cluster C_x is inside parentheses. The final selected clusters have an asterisk. Their relative width scales components' size.*

considered the selected clusters. On the contrary, if $\sigma(i) + \sigma(j) \leq \sigma(k)$, $\sigma(k)$ conserve its value, k is marked as selected cluster, and all descendants of k are unselected.

In Figure 1.13 (cluster stability values are inside parentheses) we can start the previously described process from leaves C_9 and C_{10}. As $3 + 3 > 2$, $\sigma(C_7)$ is set to 6, but C_7 is not selected as cluster. Repeating with C_7 and C_8 results in selecting C_4 as cluster and unselecting C_9, C_{10}, and C_8 $(6 + 2 < 9)$. Continuing with C_3 and C_4 excludes the possibility to select C_1 as a final cluster because $8 + 9 > 8$. Note that no comparison is ever made against C_0. Similarly, for the right section of the tree, C_2 is selected as a cluster (excluding C_5 and C_6), giving that $3 + 2 < 6$. In this manner, final clusters with maximal stability are C_3, C_4, and C_2.

1.6.3 . Spatial complexity of reviewed algorithms

Molecular clustering algorithms can be found as standalone software or as modules integrated within molecular simulation or analysis packages, such as VMD[221], AMBER[235], ORAC[220], and GROMACS[245]. These packages have gained wide acceptance among users. However, the ever-increasing volume of molecular ensembles generated by computational techniques has outpaced the advancements made in the clustering algorithms integrated within these suites. This widening gap between data generation and clustering capabilities necessitates the development of more efficient clustering algorithms or the optimization of existing ones to avoid RAM crashes or impractical run times during post-simulation analyses.

The algorithms discussed so far, vary in temporal complexities, so some execute faster than others, regardless of the programming languages in which they are implemented. All of them, however, have proven beneficial, and even the most time-consuming ones have found a niche in the molecular simulation field. Their runtime is affordable because clustering analyses typically take significantly less time to complete than the simulations generating the data to be clustered. Thus, it is not much of a concern if running these

analyses spans days or even weeks. In most cases, the time complexity is not the bottleneck.

The substantial constraint arises from their usually quadratic spatial complexity (Table 1.2). While users can decide to run their clustering jobs longer in High-Performance Computing facilities, they usually do not possess the spatial resources (RAM or HDD) needed to execute them on a large trajectory. The simple but ineffective solution to this issue has been arbitrarily selecting a portion of the trajectory instead of addressing the spatial complexity of these procedures.

Table 1.2: *Spatial complexity of reviewed clustering algorithms*

Software (Suite)	Algorithm	m (bytes)	V_{RAM}	Spatial complexity
qtcluster (ORAC)	QT	4	$\frac{m \cdot n \cdot \frac{natoms \cdot (natoms-1)}{2}}{2^{30}}$	$O(n \cdot natoms^2)$
gromos (GROMACS)	Daura	4	$\frac{m \cdot n^2}{2^{30}}$	$O(n^2)$
qt-like (WORDOM)		4		
measure cluster (VMD)		4	$m \cdot n$	$O(n)$
pyMS		4	$\frac{m \cdot n^2}{2^{30}}$	$O(n^2)$
TTClust	Hierarchicals	8		
cpptraj (AMBER)	DP	4	$\frac{m \cdot \frac{n \cdot (n-1)}{2}}{2^{30}}$	
CLoNe		4	$\frac{m \cdot n^2}{2^{30}}$	
gen.-RMSD (HDBSCAN*)	HDBSCAN	8		
gen.-Euclidean (HDBSCAN*)		8		

Two primary components consume memory resources when clustering molecular data: the “trajectory" file, which stores snapshots of molecular ensembles generated by simulations, and the data structure that contains pairwise similarity values. By storing these components in RAM or disk, clustering algorithms enhance time performance, ensuring fast accessibility to all the required information and eliminating the need for redundant recalculations each time a cluster is processed. However, it is important to note that this approach requires a sufficient amount of spatial resources in the system.

The amount of space needed for the storage of the similarity matrix (expressed in GB) can be calculated using the equations on column V_{RAM} of Table 1.2, where n signifies the total count of elements in the trajectory, while m denotes the size (in bytes) of the numeric type used to express the similarity values. Being the RMSD a float number ranging from 0.0 to infinite, it is conventional to employ floating numeric types for the representation of inter-element similarity. Although the valuable information is contained in one of the triangles, many current clustering software preserves the whole matrix to avoid the performance penalty of working with “triangular" data structures.

Some clustering alternatives like TTClust[246] use the costly choice of double-precision float ($m = 8$). Other options like GROMACS[245] and WORDOM[222] packages use single-precision floats ($m = 4$), saving half of RAM just by adjusting the precision used to express RMSD. The minimum size of standard available floats is a half-precision value ($m = 2$), which is (from the author's perspective) enough for most molecular clustering but not much implemented. Even when lowering the value of m does not imply improving the spatial complexity of these algorithms (that remains quadratic), this simple detail does significantly decrease the amount of space they need to run.

In cases where the RMSD is not the chosen metric, other approaches are followed to hold the similarity information in RAM. An example is the *qtcluster* package, which employs the maximum difference between corresponding pairs of atoms (Equation 1.19). This means holding the square matrix of the selected inter-atomic distances for each conformation in RAM is necessary. In practice, *qtcluster* allocates the values of only one triangle of that matrix for every conformation. The RAM used by its similarity matrix (in GB) is expressed in Table 1.2, where $natoms$ is the number of selected atoms.

2 - METHODS, MODELS AND COMPUTATIONAL DETAILS

As general remark, all protein-ligand contacts were retrieved using the program BINding ANAlyzer (BINANA)[247]. Molecular visualizations were produced using VMD[221] v1.9.3. Plotting of data was addressed with matplotlib Python's library and flowcharts were depicted using draw.io v13.0.3. The following sections describe the methods, models and computational details proper to each result Chapter.

2.1 . MCSS-based predictions of binding and selectivity of nucleotides

2.1.1 . Protein-nucleotide benchmark design

The Protein Data Bank (PDB) was filtered out to select a set of protein-nucleotide complexes based on different structural criteria to evaluate the Multiple-Copy Simultaneous Search (MCSS) docking and screening powers' performance. A first query was conducted to find protein complexes with each of the four nucleotides as ligands and annotated in the PDB by the following labels: AMP, C5P, 5GP, and U5P. An additional criterion used a cutoff value of 2 Å resolution to select only high-resolution X-Ray Cristallography (XRC) structures.

The resulting complexes were then clustered according to their sequence similarities to remove redundancy. If any protein's chain of a complex had at least 30% sequence identity with a chain in the protein from another complex, the two complexes were grouped into the same cluster. Each group's crystal structure with the best resolution was selected as the cluster's representative.

The 188 complexes thus selected by pulling down the results from the four queries (AMP-bound: 123, C5P-bound: 18, 5GP-bound: 21, U5P-bound: 27) were then curated to retain those that exhibit a known binding preference for the crystallized ligand. This feature was established based on the literature and the annotation of the protein (*e.g.*, a C nucleotide for CMP-kinase, *etc.*). After curation, the dataset was reduced to 132 complexes.

An additional restriction was performed to eliminate some potential redundancy associated with the presence of similar binding sites for different types of nucleotides. The procedure followed consisted of superimposing all the protein structures using the program TM-align[248] and reviewing all of them that are similar based on the TM-score (TM-score $\geq$ 0.8). Two binding sites were considered non-redundant if they differed by only one amino acid residue in direct contact with the ligand. According to this criterion, only one complex was removed from the dataset in the case of the proteins corresponding to the PDB IDs: 3DXG (U5P ligand) and 3DJX (C5P ligand); the latter was conserved to compensate for the minor under-representation of C5P. The whole procedure ended up

with a dataset of 131 protein-nucleotide complexes.

After a review of the MCSS docking calculations (see Section 2.1.3), ten protein-nucleotide complexes resulted in non-productive and were then removed from further analyses. The resulting benchmark is thus composed of 121 protein-nucleotide complexes associated with 13 known biological functions (Annex Table 9.5). The binding features of these complexes were characterized by (i) the number of contacts between the protein and its ligand, (ii) the fraction of buried surface area, (iii) the number of H-bonds in the binding site, and (iv) the energy of interaction as calculated by the MCSS scoring function.

2.1.2 . Patches, charges, and solvent models

Several phosphate group patches were used in the MCSS calculations to determine the optimal parameters for mapping nucleotides at the protein surface (see Section 2.1.3). The five different phosphate models correspond to 5' patches (R010, R110, R210, R310, and R410) that differ by the valence and charge of the phosphate group (Figure 1.2). The R010 patched nucleotide corresponds to the standard nucleotide residue defined in CHARMM, and it was the only fragment with an unfilled valence shell at the 5' end.

All the partial charges on the phosphate groups were derived from the CHARMM parameters. They correspond to the original CHARMM charges or were derived based on Manning's theory of counter-ion condensation to account for the partial neutralization of the negative charges of poly-electrolytes solution[113]. In this latter case, the net charge on the phosphate group was scaled down according to the implicit solvent model previously used in MCSS calculations performed on nucleic acids[108].

The SCAL charges model (Figure 1.2 left) was combined with a distance-dependent dielectric (Equation 1.16) with or without water molecules: SCAL and SCALW, respectively. The default charges model STD or FULL (Figure 1.2 right) was combined with explicit solvent representation and a distance-dependent dielectric (Equation 1.16): STDW, or with a constant dielectric (Equation 1.15): FULLW.

2.1.3 . MCSS docking protocol

The 121 selected proteins were prepared using the CHARMM-GUI interface[249] to convert the PDB files into CRD and PSF formats. After removing all heteroatoms, hydrogens are added to the protein using the HBUILD command from CHARMM. Water molecules were present in all the protein-nucleotide complexes, particularly in the binding site. Depending on the solvent representation (implicit/hybrid), they were either removed or included before energy minimization.

The protein targets were then submitted to an energy minimization (tolerance gradient of 0.1 kcal/mol/Å). The average deviation between the experimental structure and the minimized was around 1.0 Å for the structures optimized without water molecules and 0.5 Å for those optimized with the crystallized water molecules (Annex Figure 9.5).

The nucleotide library of fragments includes multiple conformations, 5' and 3' patches (see MCSS documentation: `https://www.mcss.cnrs.fr/MCSSDOC`). The initial default conformation used in the calculations was a C3'-endo/anti ribonucleotide with typi-

cal values of the seven torsion angles (phosphodiester backbone and base orientation). A set of five different patches on the 5' end is used in the current study with this nucleotide conformation: R010, R110, R210, R310, and R410. The nucleotide fragments are fully flexible during the calculations and are prone to adjustments of the torsion angles to better fit in the binding site (Annex Figure 9.6). Each binding region is defined by a 17Å^3 cubic box centered on the ligand centroid where all the inorganic compounds or organic ligands were removed (Figure 2.1).

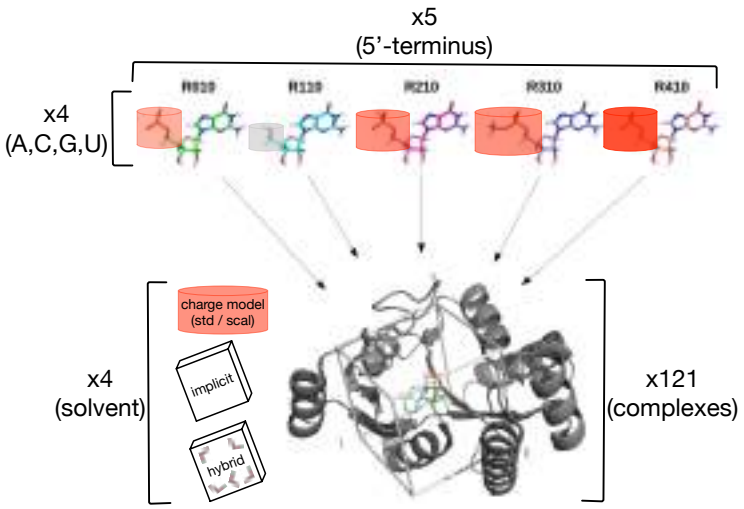


Figure 2.1: *Schematic description of the MCSS calculations performed on the protein-nucleotide benchmark. Five chemical structures of the 5'-terminus are considered (R010, R110, R210, R310, R410). For each 5'-terminus, the four standard nucleotides (A, C, G, U) are also considered. The phosphate group is enclosed into a cylinder: the bigger the cylinder, the bigger sterically, the darker red, the more negative charge (the gray color indicates a null charge). Four solvent models are evaluated depending on the charge model (std: standard, scal: scaled) and the solvent representation (implicit, or hybrid: implicit and explicit). A protein target is represented in cartoon mode with the indication of the cubic box corresponding to the explored region.*

The initial distributions of fragments were generated using 2000 groups distributed randomly and repeatedly among 25 iterations. These parameters guarantee that fragments fully saturate the binding region of all the protein-nucleotide complexes in the benchmark, *i.e.*, the atomic density of the fragments mapped into the box was at least twice that of the maximum carbon density. During the calculations, the protein targets were considered rigid. Final poses (minima) generated by MCSS were ranked by their score (Equations 1.13-1.15) in ascending order.

In the explicit solvent models (SCALW, STDW, and FULLW), the water molecules were treated independently from the fragments, which were replicated from their initial distribution during each iteration. The number of water molecules was conserved during the calculations, and they were free to move around without any constraint. However, they were not considered in the scoring as described below.

The MCSS software may be obtained after signing a license agreement upon request to Martin Karplus (marci@tammy.harvard.edu). The source code can be obtained from a Git repository on the I2BC software forge `https://forge.i2bc.paris-saclay.fr`).

2.1.4 . Clustering of MCSS distributions

A fast and straightforward clustering procedure inspired on BitClust (see Section 4.2)was performed on the MCSS distributions. The first pose (best ranked) was taken as the seed of the first cluster, and all other poses in the exploration with an RMSD less equal than 1 Å to the seed (redundant poses) were removed from the dataset. The seed was preserved, and the process resumed by taking the next best-ranked available pose as seed and performing the same comparison against the remaining poses. In the end, a set of geometrically non-redundant seeds was obtained. The MCSS results presented include the analyses of the raw (R) and clustered (C) distributions.

2.1.5 . Docking and screening power

The *docking power* was defined as the ability of the scoring functions to identify the native ligand binding pose with respect to the non-native poses generated by MCSS for the native nucleotide ligand (single nucleotide distribution). The MCSS predictions were ranked according to the success rate for the identification of at least one native pose obtained on the entire benchmark in the Top-i (Top Native in the best ranked i poses) with i taking values of 1, 5, 10, 50, and 100.

The scoring functions used were those implemented into MCSS with the four solvent models which were evaluated (see Section 2.1.2). Four alternative scoring functions used in the Comparative Assessment of Scoring Functions (CASF) challenges[32,67] have been used, as well as two Molecular Mechanics (MM)-Generalized Born (GB) models through a re-scoring scheme based on single-point calculations to assess the relative performance of MCSS in docking power. The two MM-GB models were CHARMM implementations: GBSW[250] and GBMV[251]. The other four selected scoring functions were either generic (Autodock Vina[252] and Vinardo[78]), or specialized on nucleic acids ligands (ITscorePR[85] and $\Delta_{vina}RF_{20}$[253]).

To evaluate the screening power, the MCSS distributions from the four nucleotides were merged and sorted according to their score in increasing order as in the nucleotide-specific distributions (from the more negative to the less negative or positive). In each Top-i, a prediction was considered optimal if two conditions were met: (i) it corresponds to a native pose (RMSD $\leq$ 2.0 Å), (ii) the native nucleotide was ranked ahead of the three other non-native nucleotides. For example, an optimal prediction in the Top-1 means a native pose was found with the best score from the merged distributions.

Since the scoring function was still an estimate and raw approximation of the relative binding energy, we consider as good predictions the cases where the native nucleotide was predicted within a 2 kcal/mol range from the best ranked non-native nucleotide. This threshold value corresponds to a maximum offset of 2 kcal/mol in 90% of the benchmark (STDW model), where the offset was defined as the difference between the best-ranked pose, whatever the nucleotide type, and the best-ranked pose for the nucleotide corre-

sponding to the native ligand (Annex Figure 9.7). Predictions that do not satisfy these criteria were considered poor.

The scores calculated with all the scoring functions: ITscorePR[85], $\Delta_{vina}RF_{20}$[253], Autodock Vina score[252], Vinardo[78], and the MM-GB models correspond to single-point calculations on the MCSS-generated poses.

2.1.6 . Molecular features

The molecular features were analyzed on a benchmark subset corresponding to the 17 protein-nucleotide complexes not generating any prediction in the Top-10 without any distinction from the model and patch. We consider that a given feature significantly impacted the prediction when it was found to be associated with the absence of prediction at a higher frequency than that in the benchmark (Table 9.6).

The volume calculation of the binding site was performed using the PyVOL python package[254]. PyVOL was used with the pocket corresponding to the nucleotide-binding site as input (coordinates of the nucleotide ligand of interest). The threshold value to discriminate between high or low binding volume was set to 635 Å^3, which was the average value of the distribution (30% high and 70% low). The other molecular features comprise the number of water molecules around the nucleotidic ligand and the presence of metals or other nucleotidic ligands close to the binding site. The threshold value for the number of water molecules between nwat high and low was set to 6 (nwat.low $\leq$ 6 & nwat.high $>$ 6), which was the average value of the distribution (38% high and 62% low).

The interaction features (base contacts, clashes, salt bridge, stacking) were extracted from the analysis of the binding site using BINANA[247] and *OpenEye*[255] software.

2.2 . Reinventing the wheel of molecular clustering

2.2.1 . Molecular ensembles used to benchmark clustering algorithms

The molecular ensembles used to benchmark the clustering implementations proposed in this thesis correspond exclusively to Molecular Dynamics (MD) trajectories that are generically denoted by their size (1 kF = 1000, 1 MF = 1000000 frames). The nomenclature along all the manuscript is as follows: **6 kF:** a 6001 frames Replica Exchange Molecular Dynamics (REMD) simulation of the Tau peptide[256], **30 kF:** a 30605 frames MD of villin headpiece based on PDB 2RJY[242], **50 kF:** a 50000 frames MD of serotype 18C of *Streptococcus pneumoniae*, **100A kF:** a 100000 frames MD of Cyclophilin A based on PDB 2N0T, **100B kF:** 100000 frames of an MD simulation on the bovine-rhodopsin structure embedded inside a palmitoyl-oleoyl-phosphatidylcholine hydrated membrane, **250 kF:** a 250000 frames MD of four chains of the Tau peptide that corresponds to the MD simulation of an extended Tau peptide (PDB PHF8, Álvarez-Ginarte *et al.*, unpublished work), **500 kF:** a 500000 frames MD toy trajectory constructed from randomly selected conformations of 6 kF, and **1 MF:** a one-million frames MD of ubiquitin based on PDB 1UBQ.

All trajectory and structure files above-mentioned can be found online at the following

addresses: 6, 50, 100, 250, and 500 kF at https://doi.org/10.6084/m9.figshare.c.5403930.v1, 30 kF at https://doi.org/10.6084/m9.figshare.3983526.v1, and 1 MF at https://lbqc.ucm.cl/ubiquitin_1MF/. More details about the generation of these trajectories can be found in Annex 9.2.1.

2.2.2 . Benchmarked clustering algorithms and dependencies

The novel clustering algorithms implementations developed in this thesis (QTPy, BitQT, BitClust, RCDPeaks, and MDSCAN) were programmed under the Python 3 language (https://docs.python.org) avoiding any platforms-dependent code to maximize portability between Linux, Windows and Mac operative systems. The same language was used for all scripted analyses (except the explicit mention of the contrary).

MDTraj[257] suite (v1.9.2 or higher) was among the most important dependencies of proposed clustering procedures as it allowed a high-speed parallel calculation of pairwise optimal RMSD distances. The bitarray dependency on the other hand (https://github.com/ilanschnell/bitarray), provided a bit vector data structure unavailable in pure Python and all the necessary bit operations used in the binary proposals BitQT and BitClust (v1.6.1 and v1.2.1, respectively).

In Table 2.1 it is detailed the versions, references, and source code address of the different clustering algorithms benchmarked in this work. All cutoff values in clustering jobs were set after a trial/error procedure aided by visual inspection of the generated clusters' uniformity. As qtcluster does not use the RMSD metric, we adjusted the d_c values for each trajectory analyzed with this software. We multiplied the corresponding d_c by 2.4, in analogy with a previously published report of qtcluster's authors (see Supporting information of reference 258).

Table 2.1: *Benchmarked clustering algorithms.*

Software	Version	Reference	Source Code
QTPy	0.0.1	novel	https://github.com/rglez/QT.git/
BitQT	0.0.1	novel	https://github.com/LQCT/BitQT.git
BitClust	0.0.13	novel	https://pypi.org/project/bitclust/
RCDPeaks	0.0.1	novel	https://github.com/LQCT/RCDPeaks.git
MDSCAN	0.0.1	novel	https://pypi.org/project/mdscan/
measure cluster (VMD)	1.9.3	221	https://www.ks.uiuc.edu/Research/vmd/doxygen/MeasureCluster_8C-source.html
pyMS	0.0.1	223	https://github.com/melvrl13/python-quality-threshold
qt-like (WORDOM)	0.22-rc2	222	http://www.wordom.sf.net
gromos (GROMACS)	2018.1	245	https://github.com/gromacs/gromacs/blob/main/src/gromacs/gmxana/gmx_cluster.cpp
median-linkage (TTClust)	4.6.3	246	https://github.com/tubiana/TTClust
qtcluster (ORAC)	6.0.1	220	http://www1.chim.unifi.it/orac/
DP (cpptraj-AMBER)	4.25.6	235	https://github.com/Amber-MD/cpptraj/blob/master/src/Cluster/
CLoNe	0.0.1	226	https://github.com/LBM-EPFL/CLoNe
HDBSCAN*	0.8.11	244	https://github.com/scikit-learn-contrib/hdbscan

RCDPeaks and DP at AMBER's cpptraj, used the same distance cutoff value for every trajectory; 2.5 Å for 6 and 500 kF, 4 Å for 30 kF, 1 Å for 50, 100A kF and 1 MF, and 2 Å for 250 kF. These values were set after a trial/error procedure aided by visual inspection of the number of possible centers in the decision graph. The only input parameter of CLoNe is a user-defined percentage p_{dc} of all pairwise similarity distances. This parameter was set to 0.4 for 6 kF (corresponding to $dc = 2.6$) and to 4.4 for 30 kF (corresponding to $dc = 4.0$) after author 's recommendations. The remaining trajectories could not be analyzed by CLoNe due to its excessive memory consumption.

The benchmark of all clustering algorithms was performed on an AMD Ryzen5 hexa-core workstation with a processor speed of 3.6GHz and 64 GB RAM under the 64-bit Xubuntu 18.04 operating system. Run times and RAM peaks were recorded with the */usr/bin/time* Linux command.

2.3 . NUCLEAR: an efficient assembler for the FBDD of CMOs

2.3.1 . CHARMM minimization protocol

Carbohydrates, protein, nucleic acids, ions, waters, and general small molecules parameters and topologies defined in version 36 of the CHARMM force field were employed in all minimization jobs. The coordinates of unlinked mono-nucleotides constituting each chain retrieved by NUCLEAR are loaded together with every chain of the protein and the corresponding water molecules. A four-step minimization protocol is followed in which different atom selections can move, always specifying a tolerance value applied to the average gradient during minimization cycles (*TOLGRD*); if the average gradient is less than or equal to *TOLGRD*, the minimization routine exits.

In the first stage, only water molecules and atoms linking nucleotides can move during 1000 steps of the Steepest Descent (SD) algorithm ($TOLGRD = 10$) and 10000 steps of the Adopted Basis Newton-Raphson (ABNR) algorithm ($TOLGRD = 0.1$). The second stage allows the motion of the phosphodiester skeleton and water molecules in a similar combination of algorithms and tolerances mentioned above. Water molecules and RNA atoms can move in a third minimization stage (10000 steps of SD with *TOLGRD* of 10 and 10000 steps of ABNR with *TOLGRD* of 0.1). In the final minimization phase, all atoms can move with the same combination of algorithms and tolerances followed in the previous stage.

2.4 . *In-silico* design of selective CMO against BACE1

2.4.1 . BACE1 protein candidates selection

Figure 2.2 summarizes the process followed to select the BACE1 protein coordinates we employed in our work. The complete description of each stage can be found on Section 9.4.1.

2.4.2 . BACE2 protein candidates selection

The PDB database was queried using the string *beta secretase 2 OR Macromolecule Name CONTAINS PHRASE "Beta secretase 2"* but unlike for the BACE1 case, only 17 structures were retrieved and none of them had the complete set of coordinates determined. We then restricted the candidates having at least the full coordinates of the active site and the exosite, selecting 2EWY (absence of ligand at the exosite) and 3ZKM (absence of ligand at the active site).

2.4.3 . MCSS library of standard and modified nucleotides

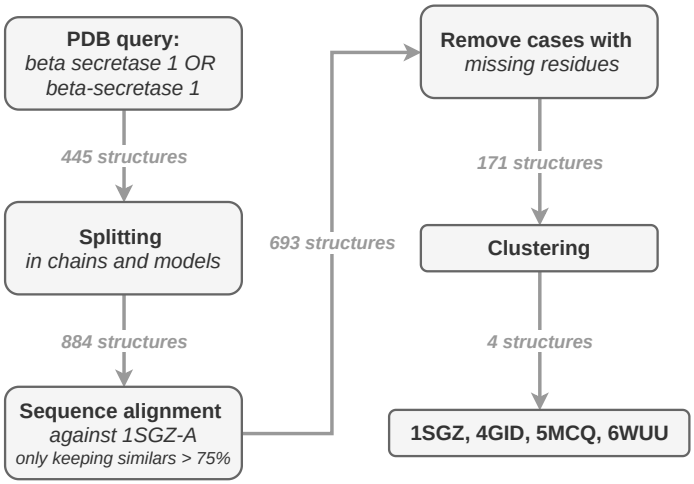


Figure 2.2: *Workflow followed for BACE1 protein candidate selection.*

Apart from the standard nucleotides (A, C, G, U/T), all the other nucleotides from the library include a modified nucleic acid base derived from the corresponding standard one; 19 A-derived (Figure 9.14), 18 C-derived (Figure 9.15), 29 G-derived (Figure 9.16), and 45 U/T derived (Figures 9.17 and 9.18).

2.4.4 . Protonation state of titratable aminoacids

For setting a specific protonation state for each of the titratable residues of proteins models, we first computed the local pK_a of amino acids using *propka*[259,260]. Then the protonation state was decided by comparing the crystal resolution pH to pK_a. For Asp and Glu, a strict comparison between the local pK_a of each residue and the pH was performed; If $pK_a < pH$, the residue was set as not protonated. If pK_a was in the range (pH - 0.1; pH + 0.1), we conducted a minimization for each possible state and set as final the less deviated in RMSD. Otherwise, residues were set as protonated.

Protocol for histidine residues protonation is depicted in Figure 2.3. Histidine residues were treated as a pair if they were at a distance less than 6Å . At acidic pH, if the pK_a > 1 pH unit, the histidine was set as protonated (HSP); otherwise, it can be protonated (HSP) or not (HSD or HSE). At neutral pH, if the $pK_a < 1$ pH unit, the histidine was set as not protonated (HSD or HSE); otherwise, it can be protonated (HSP) or not (HSD or HSE). To discriminate one state from another of a histidine (or a pair of histidines), the RMSD at 6.0 A around this histidine (or the whole pair) is calculated, and the final state is set as the less deviated in RMSD.

2.4.5 . 3D equivalence between BACE-X residues

The three-dimensional equivalence between BACE-X residues was done pairwise by setting the 4GID BACE1 conformation as the reference and 2EWY or 3ZKM BACE2 conformations as the target. After parsing structures and selecting all protein atoms of each one, we superposed the target onto reference via the *matchAlign* function of the prody package. Then we encoded the two sets of coordinates as kd-trees using the

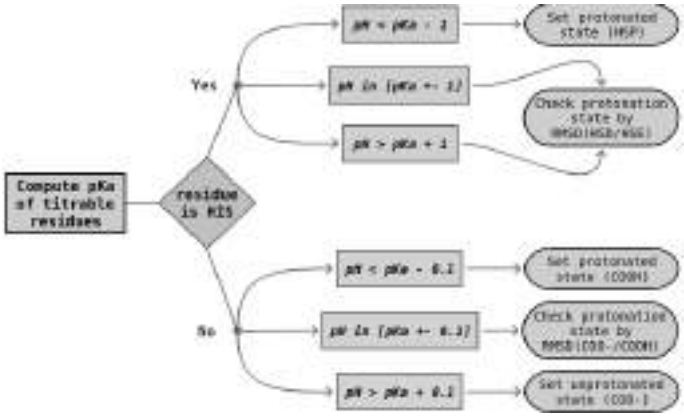


Figure 2.3: *Protonation protocol of proteins' titratable residues.*

cKDTree class in the spatial module of the scipy Python library. Later we computed for every atom in the target the nearest neighbor in the reference. Finally, as the target's equivalent residue, we assigned the one containing more neighbors to each residue in the reference.

3 - MCSS-BASED PREDICTIONS OF BINDING AND SELECTIVITY OF NUCLEOTIDES

Frequently, virtual FBD workflows present a significant limitation; the docking methods' lack of performance due to the approximate nature of their scoring functions. A fundamental assumption of this manuscript is that by employing the Multiple-Copy Simultaneous Search (MCSS) software, increased performance in the docking and the screening power (Section 2.1.5) is possible, making this tool a suitable choice for fragment-based drug design procedures involving nucleotides.

A previously published protein-nucleotide benchmark avoided the mentioned inconvenience[261]. The authors chose 62 complexes to evaluate the docking power of three methods: *AutoDock* (4.2.3), *GOLD* (5.1), and *MOLSDOCK*. However, that report is mainly outdated, with only 40% complexes with an atomic resolution less than 2.0 Å and thus not representative of the currently available structural data. Besides, the methods were tested under biased conditions: the docked region was restricted to the native ligand pose (5 $Å^3$), and the high-occupancy water molecules of the binding site were preserved within a rigid receptor.

This Chapter presents an updated and representative dataset of high-resolution protein-nucleotide complexes in which only nucleotide mono-phosphate ligands, as single-residue ligands, are included. Section 3.1 details the overall composition of the benchmark. The molecular descriptors employed to characterize the binding sites under study are presented there, and the distribution of contacts is analyzed. Section 3.2 analyzes the effect of solvent models and phosphate patches on the number of generated poses and the fraction of native-like poses obtained. After that, the docking power of MCSS (and six other scoring functions), as well as its screening power, is respectively assessed in Sections 3.3 and 3.4. Section 3.5 closes the Chapter by presenting the molecular features associated with the lack of predictions.

3.1 . Protein-nucleotide benchmark: general insights

The protein-nucleotide benchmark includes a non-redundant set of 121 complexes associated with 13 known molecular functions and a wide variety of binding modes (Figure 3.1, generated from Annex Table 9.5). The selection criteria retained to build the benchmark are detailed in Section 2.1.1. Proteins binding Adenosine monophosphate (AMP) are over-represented in PDB concerning those binding Cytidine monophosphate (CMP), Guanosine monophosphate (GMP), or Uridine monophosphate (UMP). The ligand composition in the benchmark is biased accordingly with 72% of AMP-bound complexes; the other complexes are represented in a similar proportion between 7 to 10%.

A series of molecular descriptors compose the features used to characterize the 121 nucleotide-binding sites. These features include standard contacts (closed contacts, H-

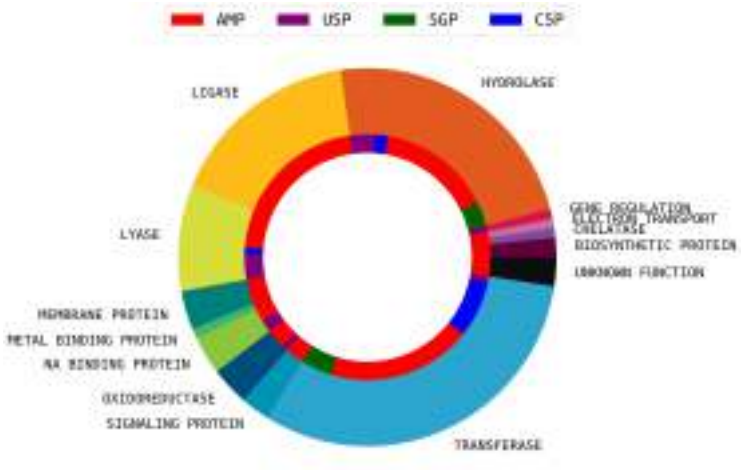


Figure 3.1: *Distribution of molecular functions and nucleotide types in the protein-nucleotide benchmark. The external wheel depicts the general distribution of molecular functions. The internal wheel shows the nucleotide-specific distribution for each function (AMP, GMP, CMP, UMP).*

bonds, and hydrophobic contacts), nucleic acid-specific contacts (stacking contacts, and salt bridges), and energy-related descriptors (buried fraction of ligand and binding energy score).

Broad distributions are observed for the standard contacts (Figure 3.2). Given that only the nucleic acid base moiety allows the chemical distinction between the four nucleotide ligands, their contacts should be represented enough in number and frequency for a reliable evaluation of the screening power (because they determine the selectivity for one specific nucleotide). The decomposition of the contacts based on the phosphate, ribose, and base moieties reveals that the base contacts are slightly more represented. They are still somewhat less frequent, especially for close contacts (Figure 3.2A-B).

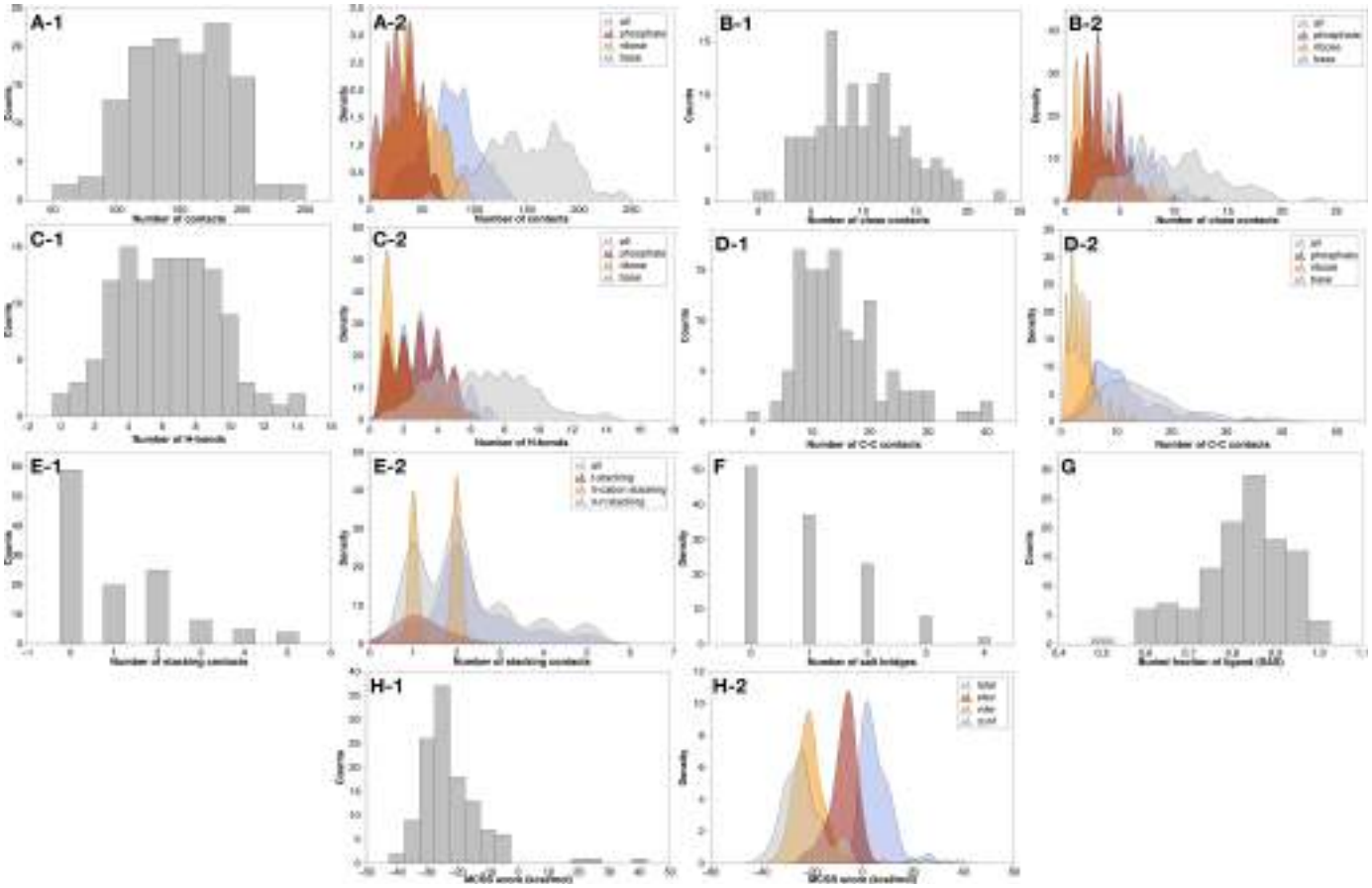


Figure 3.2: *Molecular and energy features of the nucleotide-binding sites from the benchmark of 121 complexes.* ***A-1:*** *Histogram of the number of contacts;* ***A-2:*** *Smooth histogram with decomposition per nucleotide moiety (base, ribose, phosphate);* ***B-1:*** *Histogram of the number of close contacts;* ***B-2:*** *Same as A-2 for close contacts;* ***C-1:*** *Histogram of the number of H-bonds;* ***C-2:*** *Same as A-2 for H-bonds;* ***D-1:*** *Histogram of the number of C-C contacts;* ***D-2:*** *Same as A-2 for C-C contacts;* ***E-1:*** *Histogram of the number of stacking contacts;* ***E-2:*** *Smooth histogram with decomposition per stacking types;* ***F:*** *Histogram of the number of salt-bridges;* ***G:*** *Histogram of the buried fraction of ligand (calculated from the solvent accessible surface);* ***H-1:*** *Histogram of the* MCSS *scores calculated for the ligands optimized in their binding site;* ***H-2:*** *Smooth histogram with decomposition per contribution types (electrostatics, van der Waals, conformational).*

In more than 10% of the benchmark (15 protein-nucleotide complexes), there is no direct base contact suggesting that the binding selectivity may be hard to predict in those cases and would negatively impact the screening power. Nucleic acid-specific contacts are only represented in about half of the benchmark (Figure 3.2E-F). However, the buried fraction of the ligands is more than 50% except in a single case (Figure 3.2G), indicating that the nucleotide generally binds in some well-defined cavity.

The breakdown of the contacts per nucleotide type shows a bias towards AMP, the nucleotide with almost ten times more contacts than the other nucleotides. However, the contact profile (the proportion of different kinds of contacts) is similar between the four nucleotides (Figure 3.3). Thus, we may expect AMP binding to be easier to predict, *i.e.*, to provide better performance in docking and screening powers.

The docking power, in particular, depends on both the quality of sampling and scoring. A baseline for the default MCSS scoring function (SCAL model) was established on the benchmark after minimization of the ligand by re-insertion within the optimized binding site and calculating its score (Figure 3.2H). The decomposition of the MCSS score into its different contributions (see Equation 1.13) shows that the van der Waals term dominates. Although the conformational penalty is a minor contribution (mean value of 5.5 kcal/mol), it is still significant. It stresses the importance of evaluating this term properly concerning the other contributions, given that nucleotides are very flexible (six torsion angles in nucleotide ligands). For that, good sampling is also required.

In traditional fragment-based approaches, it is recommended to use small ligands, which are easier to sample[262]. Large ligands such as nucleotides have many degrees of freedom, making computational sampling more difficult. Only a unique standard nucleotide conformation is used in MCSS while the benchmark include a large diversity of bound conformations (Annex Figure 9.6). Thus, the sampling should be efficient in identifying bound conformations that deviate from the standard (unbound) conformation, such as *syn* conformations found in 10% of the benchmark where the base orientation is opposite from the standard *anti* conformation. On the other hand, the contributions to the MCSS score should be well-balanced (the conformational penalty should not be under or over-estimated to guarantee accurate predictions).

The benchmark's high-resolution protein-nucleotide complexes include water molecules around the protein surface and the binding region. The ligand and water molecules were removed in the SCAL model, leading to some distortions of the binding sites after minimization. In the other solvent models where the crystallized water molecules were included, the original experimental coordinates were more preserved: 0.5 Å versus 1.0 Å (Annex Figure 9.5A). However, other artifacts associated with the water molecules also exist (Annex Figure 9.5B). The minimization induces displacements of water molecules in the binding region primarily due to the removal of the ligand leading to variations in their number and distribution (Annex Figure 9.5B-C). All the mentioned biases and issues will be addressed by comparing the docking and screening powers for the different solvent models (Sections 3.3 and 3.4, respectively).

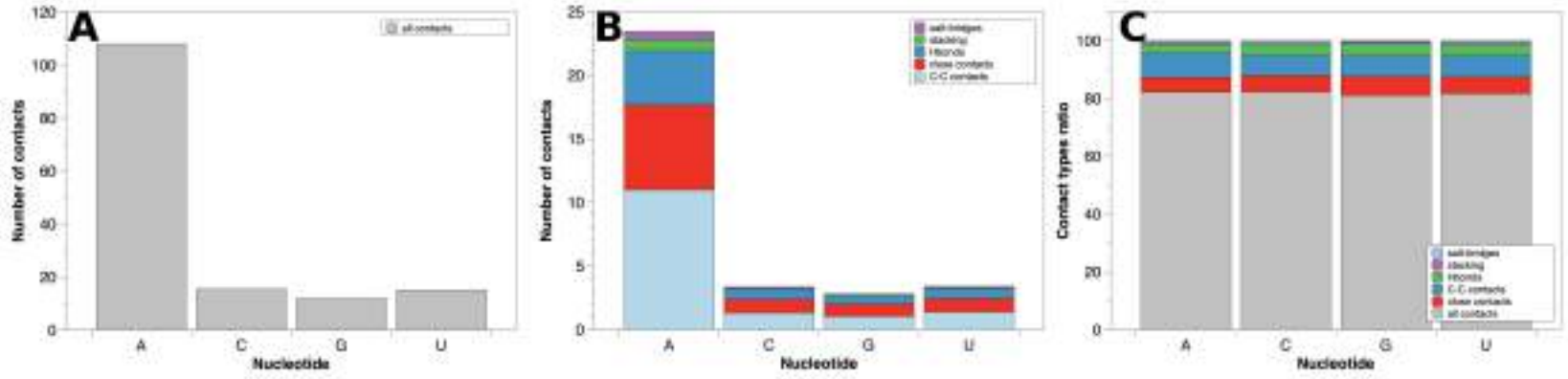


Figure 3.3: *Nucleotide breakdown of atomic contacts.* ***A:*** *all contacts;* ***B:*** *specific contacts (C-C contacts, close contacts, H-bonds, stacking contacts, salt-bridges);* ***C:*** *ratio of each type of specific contacts. The number of contacts correspond to the average value over the full benchmark.*

3.2 . Models and poses

The identification of native poses, according to standard criteria (Section 2.1.5), depends primarily on the number of generated poses and the quality of the sampling. The first MCSS parameters evaluated are the nucleotide ligands: R010 to R410 (Figure 2.1). Since their charge and size differ, they are evaluated in combination with the different solvent models. The raw distributions generally include up to several thousands of poses. The total number of poses generated depends mostly on the solvent model and the phosphate patch to a lesser extent.

The presence of explicit water molecules partially reduces the molecular volume accessible for nucleotides in the binding region. Thus, the number of poses generated with the SCAL model is much larger than that generated with any of the hybrid solvent models: SCALW, FULLW, and STDW (Figure 3.4). The comparison of the raw and clustered distributions also shows that the SCAL model exhibits the highest redundancy in the generated poses, demonstrated by the larger difference between the raw and clustered distributions for each patch.

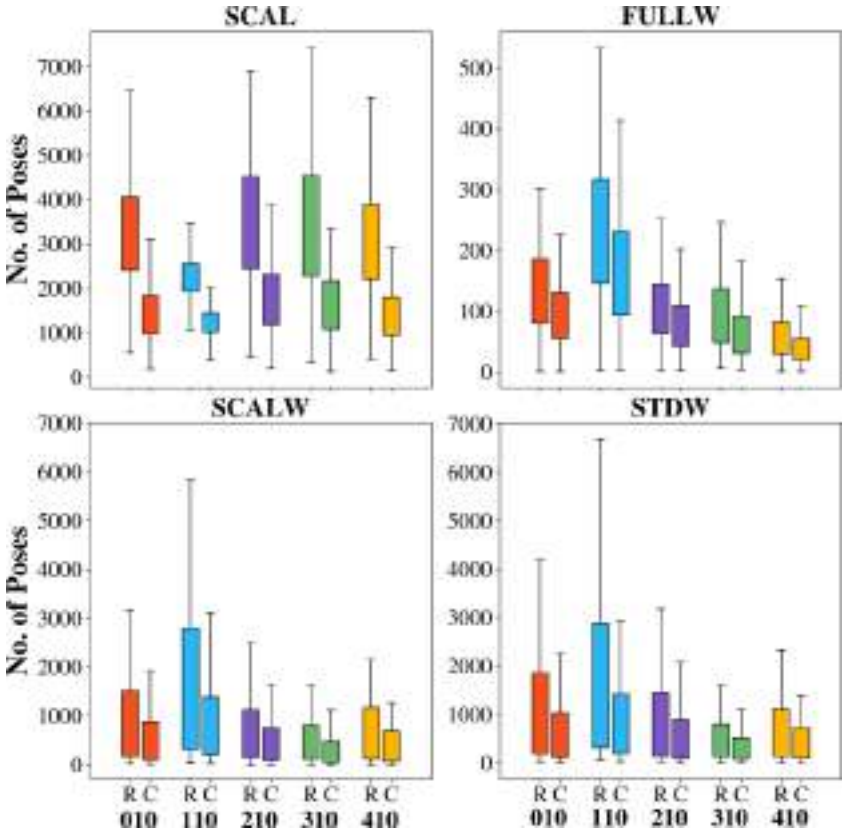


Figure 3.4: *Boxplot representation of the number of poses generated for the 121 protein-nucleotide complexes for each 5' patched nucleotide (010, 110, 210, 310, 410). Results for raw (**R**) and clustered (**C**) distributions are shown.*

Although the electrostatic contribution is not the major one in the default scoring function with an implicit solvent model (Figure 3.3), it significantly impacts the number of generated poses. Both the charge and the dielectric model have to be considered. In the SCAL model based on a distance-dependent dielectric, the observed trend is that the more negative the charge on the phosphate group (from R110 to R010, R210/R310, and R410), the higher the number of generated poses except for the more charged patch R410 (Figure 3.4). The more charged the phosphate group is, the higher the electrostatic contribution, and the more likely the pose can pass the energy threshold value of the MCSS score. The R210 and R310 patches give equivalent results with the same net charge on the phosphate group. On the other hand, a too highly charged phosphate group (R410) may also produce unfavorable interactions with negative charges at the protein surface.

In the other hybrid models, the trend is not dominated by the charge but rather by the fragment's size. The larger the patch is (from R110 to R010, R410, R210, and R310), the lower the number of generated poses and the lower the accessible volume, as mentioned above. The models based on a distance-dependent dielectric, SCALW and STDW, also follow this trend, given that R210 and R410 only differ by a proton. In the particular case of the constant dielectric model FULLW, both the charge and size effects explain why R410 is not on the lines with the other patches (Figure 3.4).

The fraction of native poses over the entire MCSS distribution for all solvent models and patches is shown in Figure 3.5. This fraction is similar for all patches in the four models, except for R310. The patch R310 carries a methyl group in one of the phosphate oxygen. This group confers the ability to establish more hydrophobic contacts than other patches. The SCAL model shows a significantly lower fraction of native poses than solvated models despite a much larger number of generated poses (Figure 3.4). As for the number of poses, the raw and clustered distributions are more scattered in the absence of water molecules. In solvated models, the fractions of native poses for SCALW and STDW are very similar. On the other hand, the FULLW model has more cases where no native pose is found, as seen by the displacement to zero of the first interquartile section for the boxplots (Figure 3.4).

3.3 . Docking power

The performance in docking power is evaluated on all models and patches using the standard metrics based on the native poses found in the Top-1 to Top-100 scores with the intermediate ranks: Top-5, Top-10, and Top-50 (see Section 2.1.5). The best performances are obtained with the SCALW and STDW models, whatever the patch used (Figure 3.6). The STDW model slightly outranks the SCALW model in the Top-1 and Top-10 for all the patches (except for R310, where the performance is equivalent for the Top-10), while the performance is pretty similar for the Top-50 and Top-100.

The best performance is obtained for patch R310. It has a success rate of 45% in the Top 1, more than 60% for the Top 10, and more than 80% in the Top-100. However,

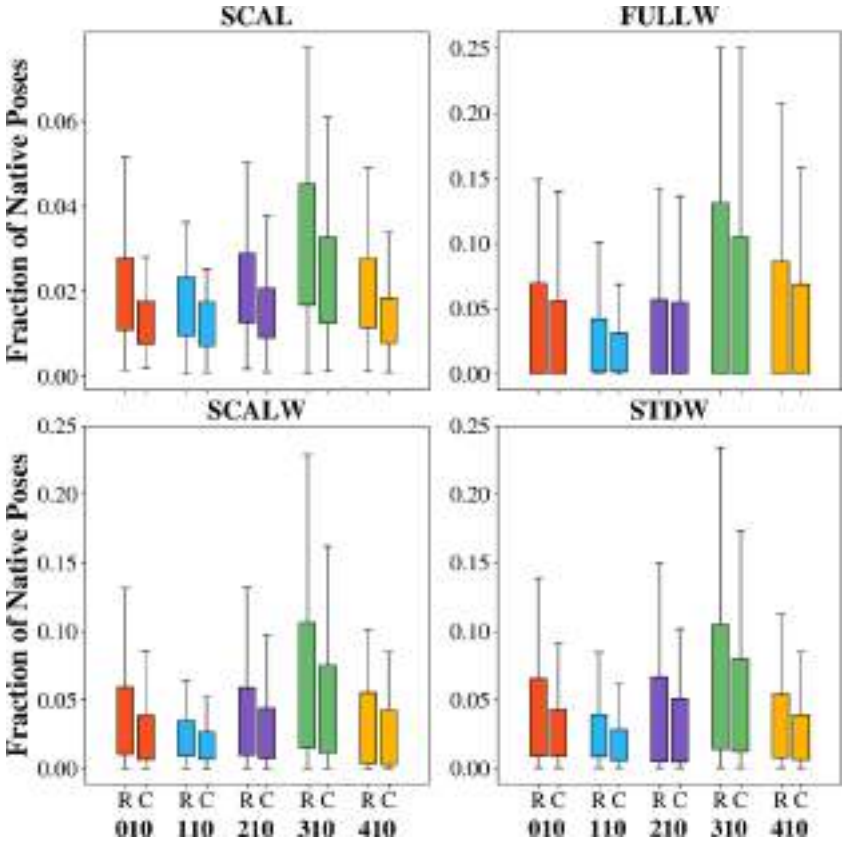


Figure 3.5: *Boxplot representation of the fraction of native poses generated for the 121 protein-nucleotide complexes for each 5' patched nucleotide (010, 110, 210, 310, 410). Results for raw (**R**) and clustered (**C**) distributions are shown.*

the gain in performance concerning the other patches is tiny in the Top-10 and Top-50. The clustering does not change the general trends observed in the raw distributions, but it slightly increases the performance in the Top-100 and, to a lesser extent, in the lower Top-i.

The better performance of hybrid solvent models SCALW and STDW over the SCAL implicit model is partly due to the conformational penalty term (Equation 1.13) corresponding to the deformation of the fragment from its optimal conformation. Although this term is generally a minor contribution, it may vary depending on the non-bonded model.

We can compare the torsion angles observed in the MCSS minima to the known ideal values and values observed in the native bound conformations of the nucleotides from the benchmark (Annex Figure 9.6). The absence of water molecules in the SCAL model reveals a few biases where, for example, the *syn* conformation is more populated than expected as compared with the experimental or the ideal values collected from the experimental structures of nucleic acids[263,264]. The SCALW and the STDW model are also biased but to a lesser extent. Only the FULLW model is exempted.

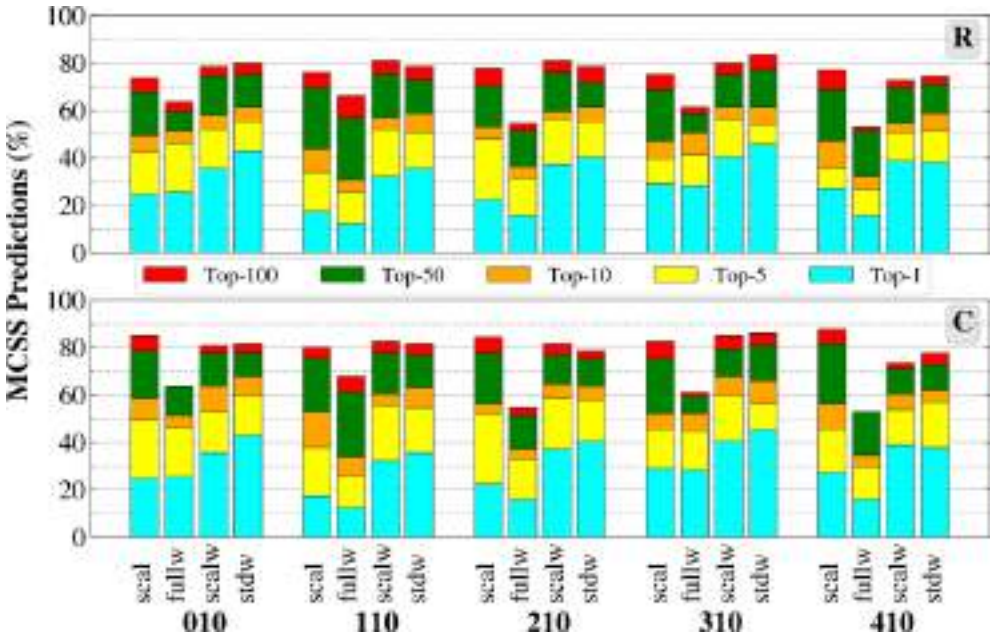


Figure 3.6: *Stacked histogram representation of the Top-i ranked native poses generated for the 121 protein-nucleotide complexes for each nucleotide patch.* ***R:*** *raw (upper) and* ***C:*** *clustered (bottom) distributions are shown.*

Another common bias in models (except for FULLW) is the over-representation of the ribose's C2'-endo conformation, while the initial conformation is always a C3'-endo conformation. It is partly due to the non-bonded model and the absence of complete solvation of the ribose moiety. In FULLW, the C3'-endo/C2'-endo representation is more balanced. Still, the phosphodiester backbone (torsion angles α and β) deviates from the optimal values because of some distortion of the phosphate group, which is highly charged and tends to stick closely to the protein surface in the absence of any screening effect (constant dielectric model).

Implicit solvent models such as MM-GB models[110,111] have been applied to the re-scoring of MCSS minima. A few other scoring functions also perform well in the CASF challenges[32,67]. Six alternative scoring functions have been selected; two correspond to MM-GB models (see Methods, section 2.1.5). The results show that the standard MCSS scoring function corresponding to the SCAL model (MCSS-SCAL) has a similar performance to Vina, slightly below that of $\Delta_{vina}RF_{20}$ (Figure 3.7). The Vinardo scoring function performs slightly better than both MCSS-SCAL or $\Delta_{vina}RF_{20}$. The other three scoring functions (ITscorePR, MM-GBSW, MM-GBMV) have a low performance. The clustering protocol described in Section 2.1.4 improves the performance of MCSS-SCAL, slightly exceeding that of Vina or $\Delta_{vina}RF_{20}$ (Annex Figure 3.8).

The MCSS scoring function associated with the STDW model still outperforms all the alternative scoring functions in the Top-1 to Top-10 in both raw and clustered distributions (Figure 3.6). In the CASF-2016 benchmark, the docking power ranges from around

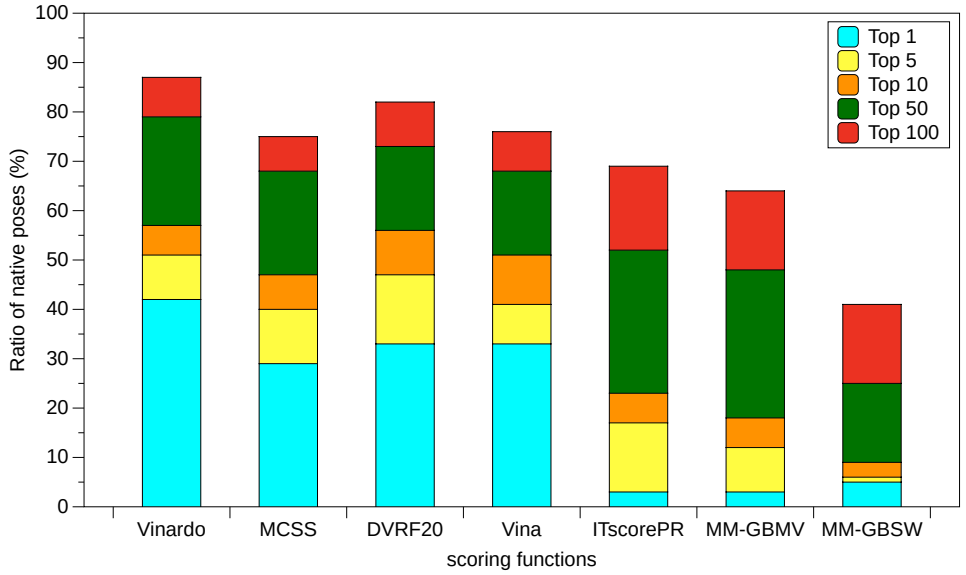


Figure 3.7: *Docking powers (Top-1 to Top-100) for Vinardo, MCSS, $\Delta_{vina}RF_{20}$, Vina, ITscorePR, MM-GBMV, and MM-GBSW using the patch R310. The two MM-GB models use the molecular mechanics terms from CHARMM (MCSS with SCAL model) and the solvation contribution from the respective Generalized Born models implemented in CHARMM (see Section 2.1.5).*

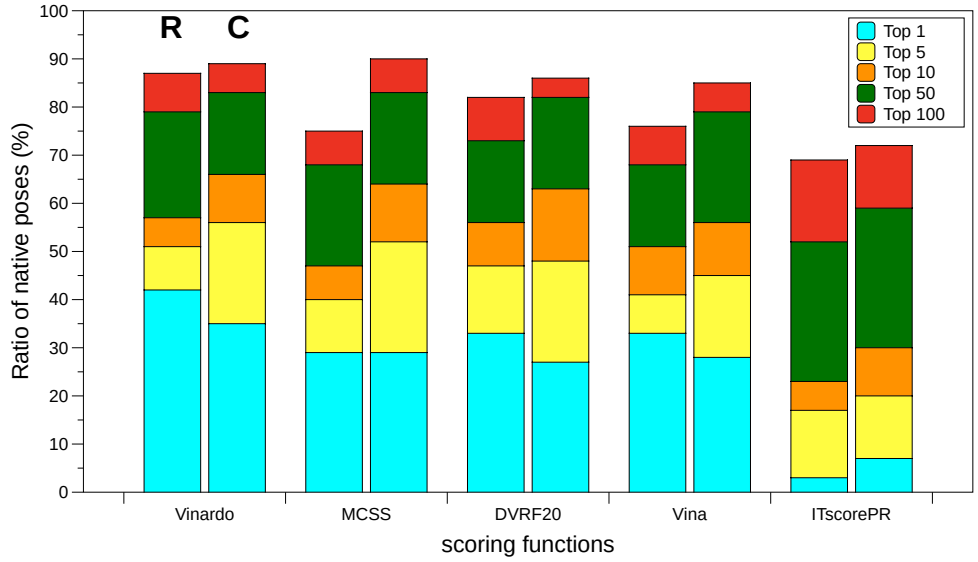


Figure 3.8: *Docking powers (Top-1 to Top-100) for Vinardo, MCSS, $\Delta_{vina}RF_{20}$, Vina, and ITscorePR and the impact of the clustering filtering (using the patch R310).* ***Left bar (R):*** *no clustering;* ***Right bar (C):*** *clustering.*

30% to 90% for a variety of scoring functions[67]. The docking power is around 90% for both Vina and $\Delta_{vina}RF_{20}$. On the current benchmark, their performance is only 33%,

indicating the challenging task of scoring charged ligands such as nucleotides. Vinardo performs slightly better (42%) and also MCSS-STDW (45%).

Because of the composition bias in the benchmark, the performance was then analyzed by nucleotide type. Since the adenosine is over-represented in the benchmark, the performance for that specific nucleotide generally follows the global trend described above (Figure 3.9). However, the performance for guanosine decreases for the larger patches R210 to R410, whatever the model used. Only the smaller patches R010 and R110 give a similar performance or better in some cases; the success rate with R110 is even better from Top-1 to Top-50, indicating the existence, as discussed before, of a size effect that drives down the performance (guanine is slightly more voluminous than adenine).

Consistently, the performance generally improves for pyrimidines (C or U), which are smaller than purines. On the other hand, the performance is degraded in the smaller nucleoside ligands (R110) that do not carry any phosphate group (uncharged). The pyrimidic nucleotides are better predicted, especially for the two best models, SCALW and STDW, with R310. The predictions are equivalent or degraded for the more highly charged patch R410, especially with U. The analysis of the clustered distributions confirms the observed trends of the raw distributions, with improved performances reaching 90% to 100% for the Top-100 in a more significant number of models and patches (Annex Figure 9.1).

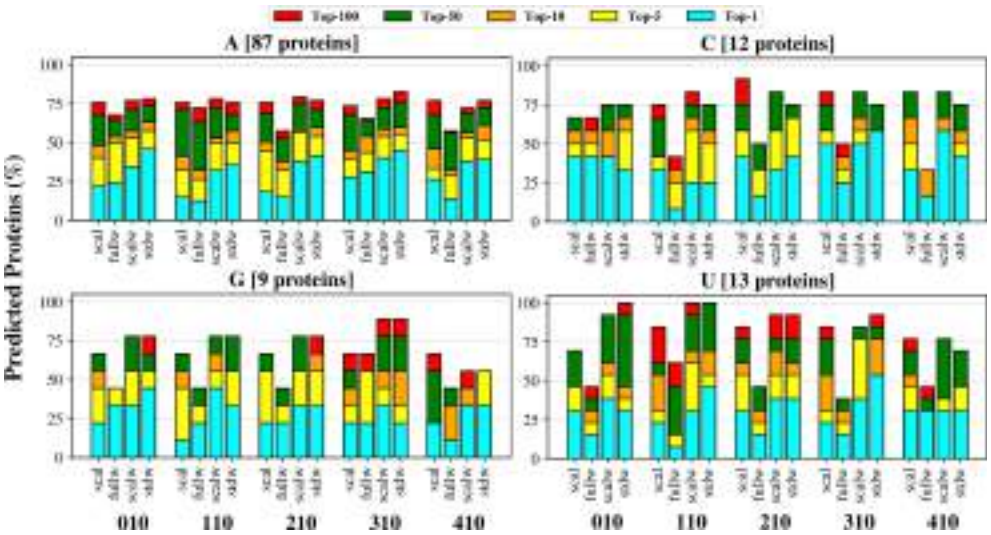


Figure 3.9: *Nucleotide decomposition of the success rates obtained for each solvent model and patch. The data are shown for the raw distribution (without clustering) and each Top-i.*

3.4 . Screening power

In the benchmark, we assume that the crystallized nucleotide is always the native and more specific nucleotide, *i.e.*, it is the only nucleotide ligand with a detectable affinity or

the best binder among the four nucleotides. Based on this assumption, we defined the *screening power* as the ability to rank the native nucleotide ahead of the other three nucleotides. In that case, we will refer to optimal predictions as the native pose is identified, and the native nucleotide is ranked first. The other predictions are considered poor even if native poses are found for the native nucleotide. As an illustration, we show the results obtained for one protein-nucleotide complex (PDB ID: 1KTG) for both SCAL and STDW models (Figure 3.10).

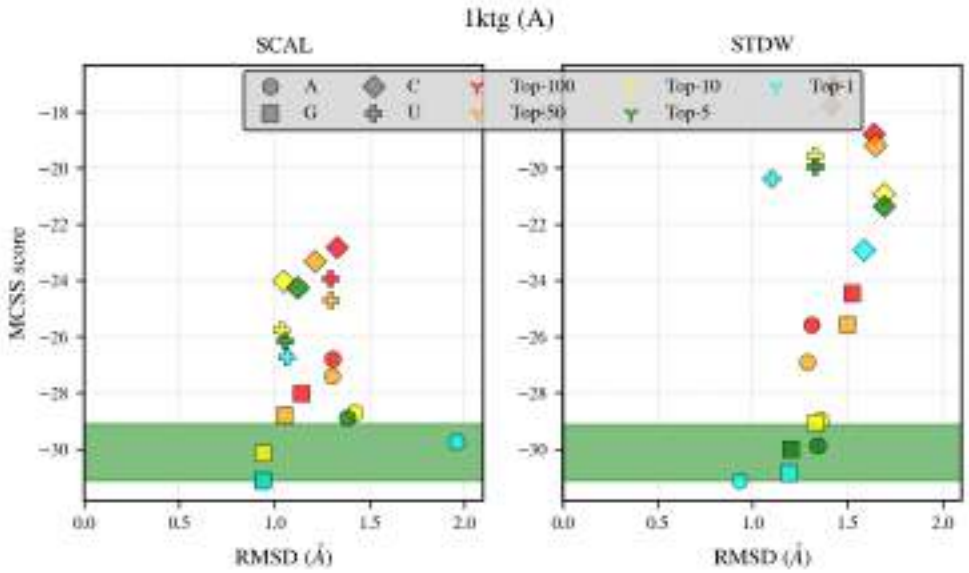


Figure 3.10: *Binding selectivity predictions for 1KTG.* ***Left:*** *SCAL model (R310);* ***right:*** *STDW model (R310); the interval of MCSS scores corresponding to a 2 kcal/mol range is indicated by the green bar. Each Top-i for $i > 1$ is represented by a single point corresponding to all its members' average RMSD and score.*

The best-ranked nucleotide is the native one (A) in the STDW model; other poses of the native nucleotide are also identified (Top-5, Top-10, etc.), but only one is within the 2 kcal/mol score range (good prediction). Some poses corresponding to non-native G nucleotides are within the MCSS score range of 2 kcal/mol. The prediction is optimal in the STDW model since the best-ranked pose corresponds to the native nucleotide. In the SCAL model, the pose with the best score corresponds to a non-native G nucleotide, but the Top-1 for the native nucleotide is within the 2 kcal/mol range; it is not considered optimal but a good prediction. The other poses for the native nucleotide, which lie out of the 2 kcal/mol range (Top-5, Top-10, etc.), correspond to poor predictions.

The results' analysis focuses on comparing the standard SCAL model (without explicit solvent) and the hybrid STDW model with the R310 patch. The STDW model shows a significant performance gain with explicit water molecules (Figure 3.11). In the optimal predictions, the STDW outperforms by 15 to more than 30% from the Top-1 to Top-100, respectively. In all Top-i, the STDW optimal predictions consistently exceed the SCAL

total predictions. Moreover, the ratio of optimal/good predictions is always much higher in STDW (Figure 3.11).

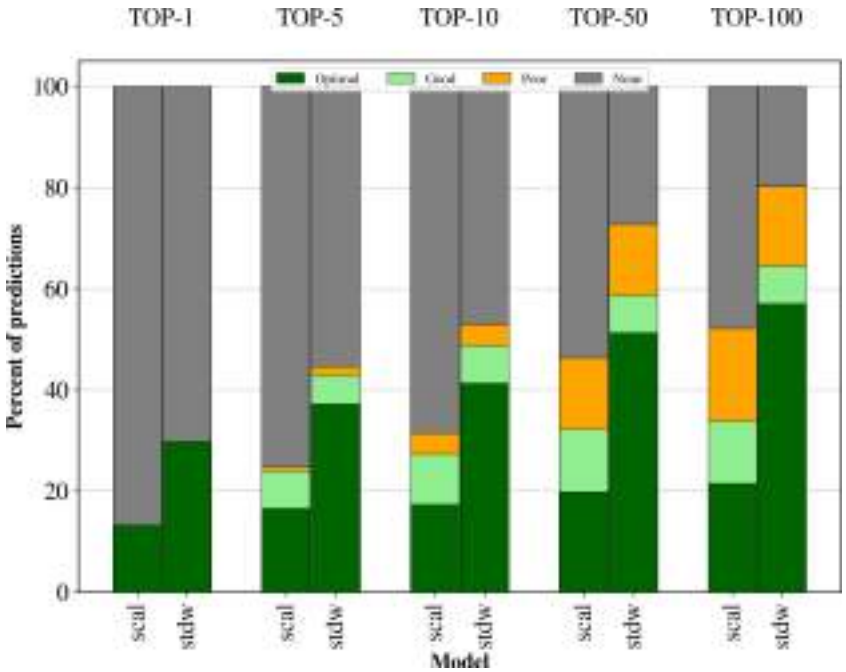


Figure 3.11: *Binding selectivity predictions.* ***Optimal:*** *native nucleotide as the best ranked;* ***good:*** *native nucleotide ranked within a 2 kcal/mol range from the best ranked non-native nucleotide;* ***poor:*** *native nucleotide ranked out of the 2 kcal/mol range.*

The docking power determines in part the magnitude of the screening power, *i.e.*, the more native poses, the more likely the native nucleotide is well ranked and associated with an optimal or good prediction. Considering only the cases where both models generate at least one native pose in the respective Top-i, we exclude the contribution of the docking power to the screening power (Figure 3.12).

These results show that the STDW model still has a better screening power, indicating that the hybrid solvent model can intrinsically better discriminate the native nucleotide from the non-native ones. The analysis of the score distributions by nucleotide type suggests that the reason for the better screening power of STDW lies in a scoring bias. In the SCAL model, purines that are composed of more atoms are slightly better scored than pyrimidines (C or U), with a preference for G over A nucleotides (Figure 3.13).

In contrast, A nucleotides scored better in the STDW model, while the other three have similar distributions. Another difference is the much more extensive range of scores for all four nucleotides. The more favorable scoring of A is consistent with more tightly binding modes, a known bias of the benchmark as mentioned previously (Figure 3.3). Moreover, the nucleotide decomposition of the screening power shows no significant difference in performance between A and the other three nucleotides, although it is slightly better in the Top-100 (Figure 3.14). Thus, the absence of any apparent bias in the

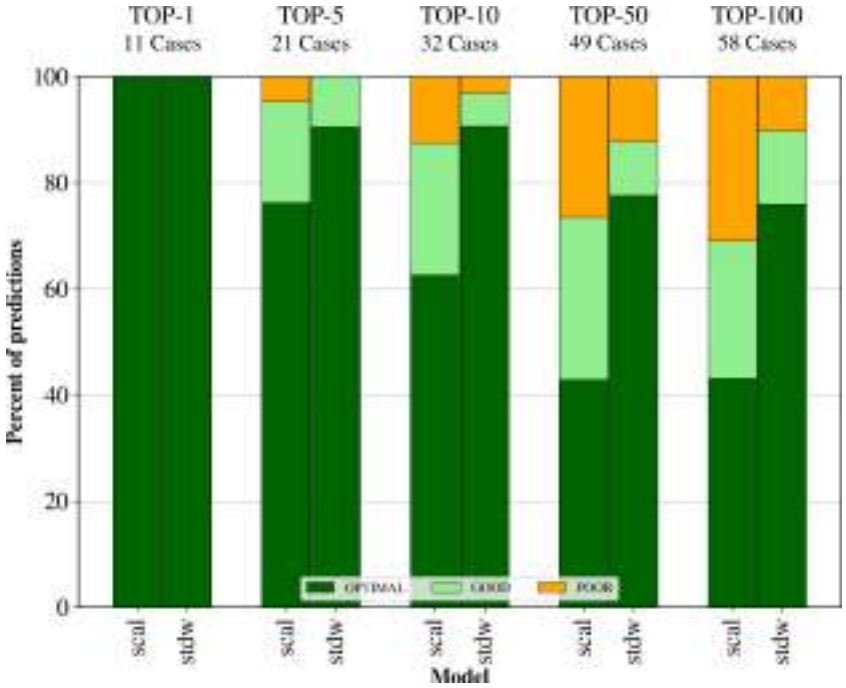


Figure 3.12: *Screening powers on the benchmark subset corresponding to the predictions common to the SCAL and STW models.* ***Optimal:*** *native nucleotide as the best ranked;* ***good:*** *native nucleotide in the ranked within a 2 kcal/mol range from the best ranked non-native nucleotide;* ***poor:*** *native nucleotide ranked out of the 2 kcal/mol range.*

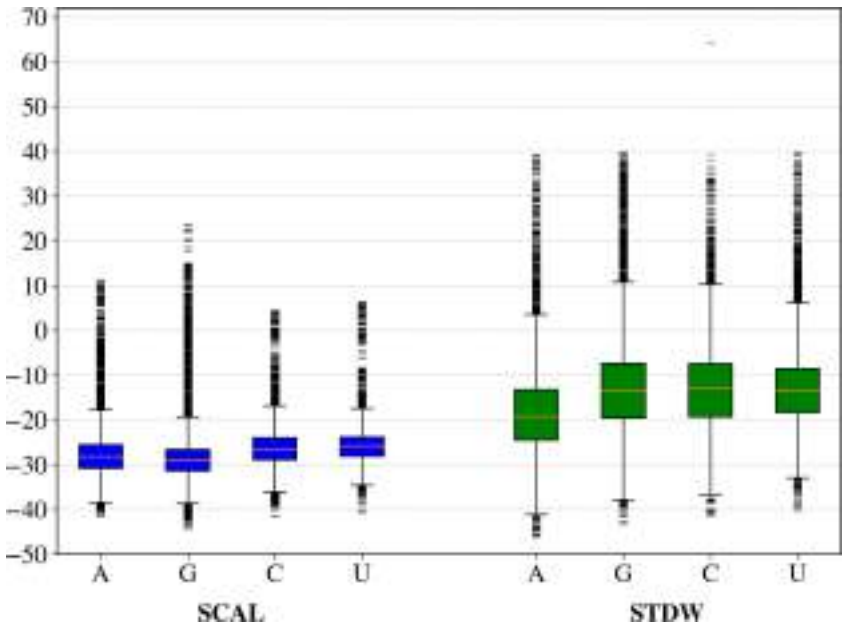


Figure 3.13: *Distributions of the nucleotide-dependent MCSS score for the SCAL or STDW models (R310).*

STDW scoring makes it more efficient in terms of screening power. The main difference between the SCAL and STDW models is the presence of explicit water molecules, leading to increased sampling and scoring performance.

The current scoring functions (tested on the CASF-2016 benchmark) do not exhibit high screening powers, which reach 30% or a bit more than 40% for the highest success rates in the Top1% and a bit more than 60% in the Top10%[67]. The comparison with the results of this study (Figure 3.11) is risky because the Top1% or Top10% would represent a two- or three-fold number of poses (Figure 3.4) concerning the approximate 1000 poses generated in the CASF-2016 scoring benchmark[67]. Furthermore, the molecular diversity of the four nucleotides is limited to the few atoms of the nucleic acid base, making the discriminatory scoring much more challenging.

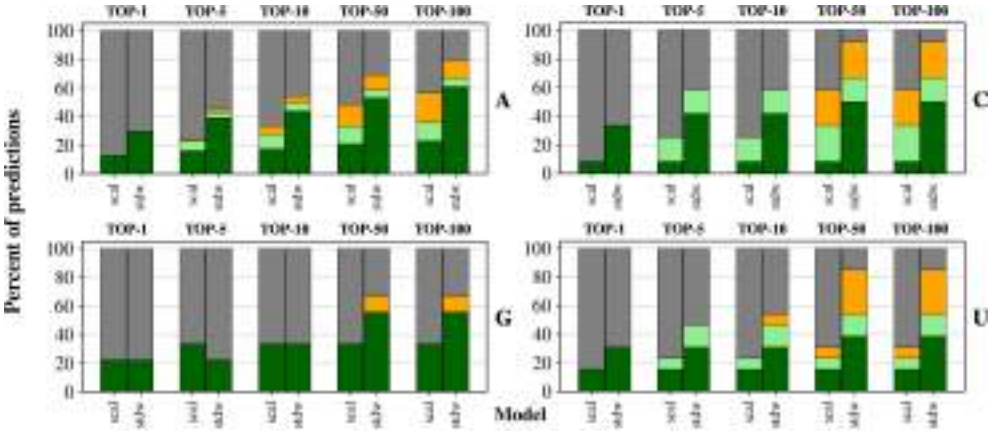


Figure 3.14: *Decomposition of screening powers per nucleotide type.* ***Optimal:*** *native nucleotide as the best ranked;* ***good:*** *native nucleotide in the ranked within a 2 kcal/mol range from the best ranked non-native nucleotide;* ***poor:*** *native nucleotide ranked out of the 2 kcal/mol range.*

3.5 . Molecular features

We define a series of representative features for nucleotide ligands to understand better the role of solvent and other molecular properties associated directly or indirectly with water molecules. Then, we determine the relationships between these features and the lack of prediction, which are represented by logic diagrams (Upset plots). We classify the features into three main groups related to: (i) the binding site properties (volume, number of water molecules, presence of metals or other nucleotidic ligands), (ii) the conformational properties (purine/pyrimidine, *syn*/*anti*), and (iii) the interaction properties (contacts, clashes, stacking, salt bridges). Whether a feature is statistically significant is determined by its relative frequency in the subset of the benchmark with no prediction (Section 2.1.6).

The only binding site feature that correlates significantly with the absence of prediction is a low volume of the binding site (Figure 3.15A), as calculated by PyVOL[254] (Section 2.1.6). On the contrary, a low number of water molecules within the binding

site is not particularly detrimental. Metal ions usually stabilize the phosphate group and occupy some volume in the binding site (it is correlated with a low volume of the binding site and a low number of water molecules). Although it is removed from each protein target, its absence in the calculations is not particularly detrimental either.

Among the conformational features, none is an impacting feature (Annex Figure 9.2). It is noteworthy that the syn conformation is not associated with the lack of prediction (Annex Table 9.6), while the initial conformation of all nucleotides is anti, confirming the quality of the MCSS sampling. On the other hand, three interaction features negatively impact the performance: the absence of salt bridges, the presence of clashes with water molecules, and to a lesser extent, the absence of stacking contact (Figure 3.15B). Among these latter contacts, the π-π interactions contribute more to the negative impact on the predictions (Annex Figure 9.4). The presence of clashes with water molecules might induce some distortions within the binding site during the protein target's preparation.

Suppose we focus on the non-predicted cases specific to the STDW model with the R310 patch. In that case, the observations described above remain valid with very similar trends for all the molecular features (Annex Table 9.4). Nevertheless, the syn conformations are slightly more frequent in the no-prediction cases (Annex Table 9.4), indicating a less efficient sampling for the larger R310 patch in size. In the non-optimal predictions, which fail to score the native nucleotide as the best ranked (i.e., good predictions, Figure 3.11), similar trends are again observed but with two specificities associated with the metals and stacking contacts (Annex Table 9.1). First, metals' presence negatively impacts the performance suggesting that metals contribute directly or indirectly to the nucleotide selectivity. Second, the absence of stacking contacts makes it more challenging to score the native nucleotide properly; the binding selectivity of purines versus pyrimidines, in particular, can be easier to identify in the presence of stacking contacts.

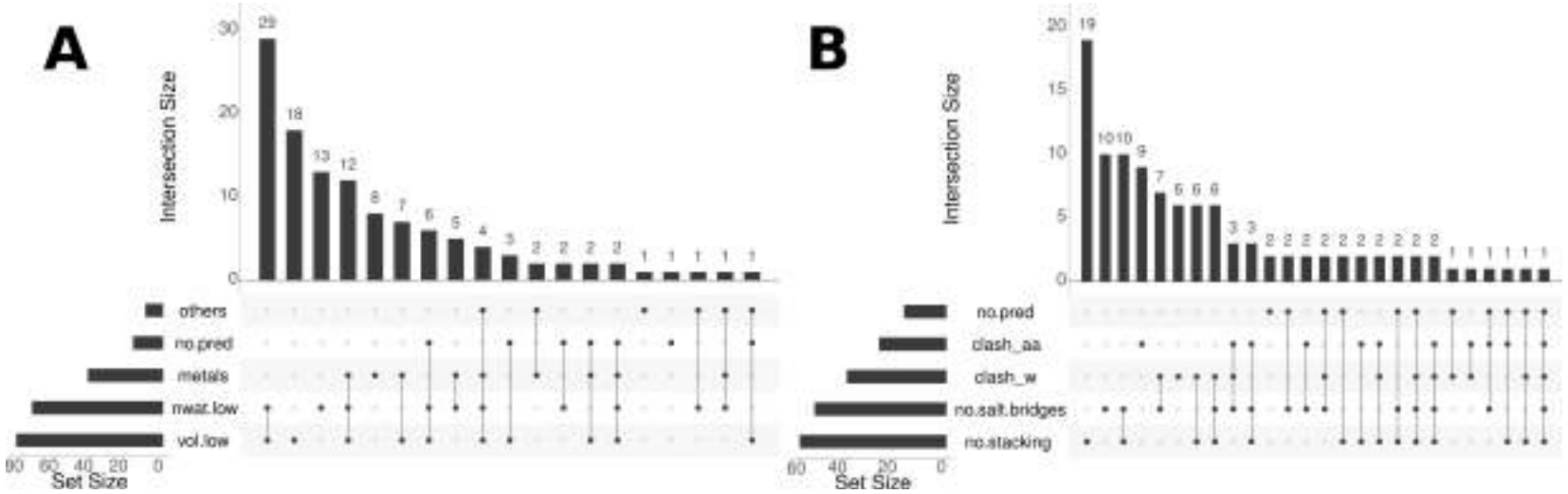


Figure 3.15: *Upset diagrams of the impact of molecular features on the Top-10 predictions.* ***A:*** *binding site features.* ***B:*** *interaction features. The intersections with only one member are not shown;* ***others:*** *presence of additional nucleotidic (nucleic acid) fragment in the binding site;* ***no.pred:*** *no prediction;* ***metals:*** *presence of metal(s) in the binding site;* ***nwat.low:*** *presence of a number of water molecules below the threshold value;* ***vol.low:*** *volume of the binding site below the threshold value;* ***no.base.contacts:*** *absence of contacts with the nucleic acid base;* ***clash_aa:*** *clash(es) with amino-acid residues;* ***clash_w:*** *clash(es) with water molecules;* ***no.salt.bridges:*** *absence of salt-bridge;* ***no.stacking:*** *absence of stacking.*

As described above, a low volume of the binding site is detrimental *per se* to the prediction performance. Once the experimental structure is optimized after the removal of the ligand (metal and the water molecules in the SCAL model), the volume can sometimes undergo significant variations: either decreasing or increasing (Annex Figure 9.3). The average variation shrinks the binding site by 27 to 30 $Å^3$ for the SCAL and STDW models, respectively. In two-thirds of the benchmark, the binding site shrinks by an average of 87 (SCAL) to 92 $Å^3$ (STDW). In one-third of the benchmark, the binding site expands by an average of 92 (STDW) to 95 $Å^3$ (SCAL). Thus, a similar trend of variations is observed for both SCAL and STDW models.

However, only the STDW is significantly impacted in the performance for the prediction of the Top-10 (Annex Table 9.3); the shrinking of the binding site combined with the presence of water molecules prevents the identification of any native pose in the Top-10 in the concerned cases. This is confirmed by the fact that 9 of the 17 proteins in the subset with no predictions in the Top-10 exhibit recovered predictions in the upper Top-i with a smaller patch such as R110 (Annex Table 9.2). In six other cases, the absence of predictions with the STDW model can be imputed to the presence of water molecules (Annex Table 9.2). Finally, only two cases do not provide any prediction in the Top-i.

4 - REINVENTING THE WHEEL OF MOLECULAR CLUSTERING

There is a popular advice for beginner researchers that intends to avoid them a waste of time and efforts whenever a valid solution already exist for they problems: **Do not reinvent the wheel !** As much solid as this maxim may sound, there are circumstances when a "wheel" should (or must) be reinvented to accomplish a particular need. We were persuaded (and we proved it right later) that in the clustering field of molecular ensembles, a re-optimization of popular and effective algorithms was possible and needed in order to fulfill our particular goals and potentially, those of many other users.

In the workflow we are following to design oligonucleotides through a fragment-based approach, there is a compulsory need to perform clustering analyses (see Section 1.6). Although not included in the work described by this thesis, the conformational dynamics of generated inhibitor candidates are envisaged to be analyzed via MD simulations, whose trajectories would also require efficient clustering algorithms to be processed.

In this chapter, we present our efforts to diminish the spatial resources of four geometrical clustering algorithms already applied to molecular ensembles: (i) the Quality Threshold (QT, Section 4.1), (ii) the Daura (Section 4.2), (iii) the Density Peaks (DP, Section 4.3), and (iv) the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN, Section 4.4) algorithms.

The implementations proposed in this work were benchmarked against the most widely used related alternative methods. It is important to note that the benchmarks were designed to compare software tools that may have differing time and spatial complexities. While complexity analysis provides insight into algorithmic scaling, empirical benchmarks on real-world datasets can still offer a meaningful performance assessment from the user perspective. Thus, the presented comparisons remain valid as they quantify trade-offs and aid in software selection for specific use cases.

It should be noted that not all methods were tested on the same trajectory datasets. This situation occurred as the inherent non-linearity of scientific research led to the completion of some implementations earlier than others, and trajectories available for recent alternatives had not yet been generated initially. Although an ideal benchmark would gather all algorithms and datasets, this elegance was sacrificed due to practical constraints regarding access to computational resources. Despite variations in trajectory lengths across methods, these benchmarks constitutes helpful evaluations. The central insights and relative performance discussions can characterize the strengths and weaknesses of each clustering procedure.

4.1 . BitQT: a graph-theoretical approach to the QT clustering

The advantages of QT clustering were already discussed in Section 1.6.2.1. The guarantee to produce clusters in which the collective similarity of elements is preserved captures the attention of many users. However, when cautiously examined, we noted that the mainstream alternatives available were either flawed or significantly inefficient . This fact led us to develop a proof of concept to demonstrate their wrongness and later to conceive a computationally accessible solution that could be readily used to process molecular ensembles.

4.1.1 . Inaccurate implementations of QT

The QT original formulation has two clear implications (see Section 1.6.2.1): (i) No pairwise distance inside a cluster can be greater than a predetermined threshold (a measure of quality), and (ii) the cluster diameter must be minimum. The first criterion is the heart of the algorithm (and the one that should be preserved in any valid variation), while the second merely limits the size of recovered clusters. Visibly, there is no sense in guaranteeing the latter if the former is not met, and no implementation of this algorithm should be taken as correct if those conditions do not hold. Variations of QT do exist[265,266] in which the core idea of respecting a quality threshold is never violated.

The VMD's documentation stated that the *measure cluster* command was based on the QT algorithm from version 1.9 (released in 2011) to 1.9.4a20 (under development in 2019). As a consequence, numerous reports in the literature (even in book chapters) describe the use of VMD to perform QT. A report of the QT algorithm was published by Melvin *et al.* in 2016[223]. The source code (referred to as pyMS from now on) exhibits a similar workflow to that spotted in the *measure cluster* command of VMD (see Figure 4.1). The same results of pyMS are retrieved WORDOM[222], which proposes a clustering option supposedly similar to QT. In Figure 4.1 we demonstrate that all these variants are wrong in their claim of performing QT.

Clustering analysis of a short MD trajectory corresponding to the tau peptide[256] was conducted using (i) the VMD's *measure cluster* internal command, (ii) the WORDOM's qt-like method, (iii) the pyMS script, and (iv) an implementation of the original QT algorithm developed by us (available at https://github.com/rglez/qt). For all runs, an RMSD cutoff of 4 Å was set and the first five clusters requested with an atom selection corresponding to all protein atoms.

Next, RMSD pairwise distances between all the elements inside every cluster returned by each method were measured and plotted in the corresponding graph (Figure 4.1). If any algorithm could perform QT, none of the plotted values would have been greater than the specified cutoff; *i.e.*, the quality guarantee of at least 4 Å should have held between all pairs of elements inside a cluster. As shown in Figure 4.1, only our in-house implementation of QT satisfies the discussed constraint.

The pyMS script gives precisely the same results as WORDOM and is not visible in Figure 4.1 because of superposition. Nitpicked examinations of VMD, WORDOM, and pyMS were conducted to realize the causes behind the displayed inconsistencies. The source code of the *measure cluster* and that of pyMS revealed that they are implementing an algorithm commonly credited to Daura[224] (see Section 1.6.2.2) that is available in the

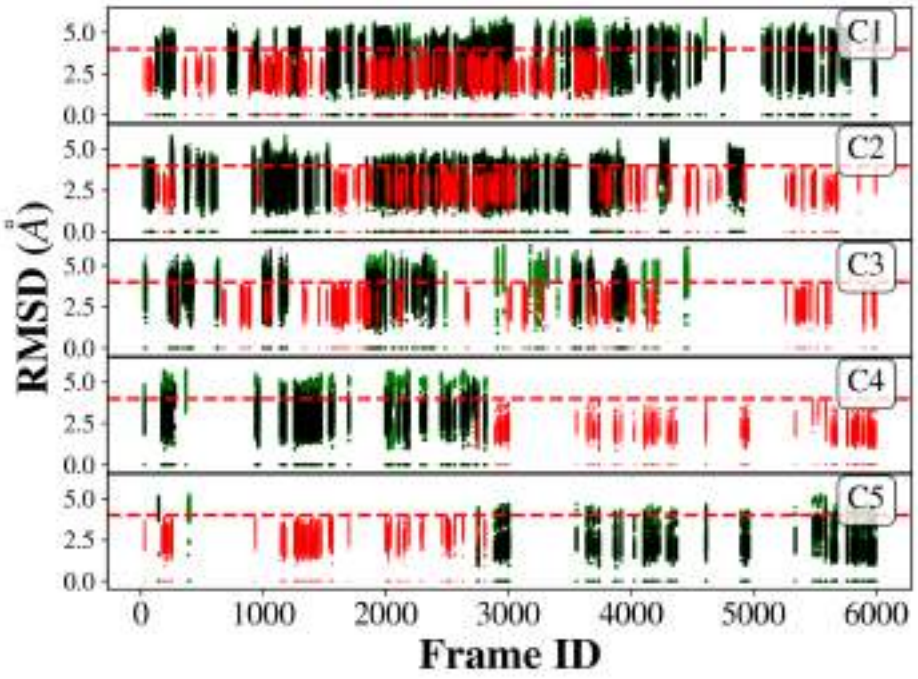


Figure 4.1: *All vs. all RMSD values of structures contained in each of the first five clusters (C1, C2, C3, C4, and C5, respectively) retrieved by supposedly QT clustering algorithms implemented in VMD (black), pyMS (blue), and WORDOM (green, invisible due to superposition with WORDOM values). Our implementation of the original algorithm is highlighted in red. The broken line indicates the specified cutoff of 4 Å.*

GROMACS package through the gromos clustering option. The same statement applies to the qt-like method in WORDOM, which returns the same information as pyMS.

The simplistic approach followed in Daura clustering steps only guarantees that all elements inside a cluster have a similarity distance less than a specified threshold when compared to the seed of the cluster (see Section 1.6.2.2). No restriction concerning the collective intra-cluster similarity is applied, which explains why the snapshots represented in Figure 4.1 repeatedly exceed the threshold compared to all others in the same cluster. Some scientific reports inaccurately claiming to perform QT clustering, as well as the potential implications of their confusion are detailed in Annex 9.2.2.

Although we proposed a freely available implementation of QT for MD (referred to as QTPy from now on), it is not suitable to be used with relatively big molecular ensembles because it takes long run times. However, this proof of concept could be used to analyze small molecular ensembles or refine clusters obtained by other algorithms saved as independent trajectories. From a developer's point of view, our proposal can serve as a gold standard to benchmark future versions aiming to be faster. In light of the precedent situation, we created BitQT, a heuristic variation of QT that can output equivalent results to the original algorithm at a much less computational cost. It has been devised using a parallel with the Maximum Clique Problem (see Section 1.5.2).

4.1.2 . From QT to the Maximum Clique Problem

We have already discussed that the crucial aspect of the QT algorithm lies in its ability to guarantee that all pairwise similarities inside a cluster will remain under a threshold k. During the execution of the algorithm (see Algorithm 1), two conditions must hold to populate clusters under creation: Condition 1- the entering element minimizes the increase of the cluster diameter under construction, and Condition 2- the diameter of the cluster under construction does not exceed the threshold k.

To make a parallel between QT and the MCP, we can represent each element of an MD trajectory as a node of an undirected graph in which edges depict RMSD similarity between nodes. Only edges with an RMSD less or equal to the threshold k are allowed. In that context, QT can be seen as an iterative search of cliques. However, QT cliques are not necessarily maximum due to Condition 1 of the algorithm, which ensures that they should have a minimum weight instead of a maximum cardinality. Condition 1 requires the diameter of the clusters to be minimum. Still, it is Condition 2 that ensures not to exceed the quality threshold in the pairwise similarity of retrieved clusters.

Conveniently, the QT algorithm could be redefined to search for maximum-sized clusters instead of minimum-weighted ones without compromising the pairwise similarity assured by the second condition. In most clustering applications, maximizing the size of the clusters is a desirable feature. Relaxation of Condition 1 automatically converts QT in an MCP problem, accessible by the graph theory tools. This approach profoundly impacts how molecular similarity can be encoded and the efficiency of algorithms used to solve the problem, as discussed in the following sections.

4.1.3 . Binary encoding of RMSD pairwise similarity

As the ultimate goal of our clustering proposal is to partition all MD trajectory elements, all the pairwise similarities should be analyzed. This information can be saved in RAM as a matrix to accelerate the algorithm's run time. However, instead of using floats as the numeric type, we followed a different approach to diminish the value of m in Table 1.2.

If we conceive the QT algorithm as an MCP problem, after considering the relaxation of Condition 1 our search will be focused on finding cliques of maximum cardinality, and no helpful information is extracted from the weight of the edges other than its absence or existence. This information can therefore be encoded as a binary matrix M where $M_{ij} = 1$ if nodes i and j are similar ($\mathrm{RMSD}_{ij} \leq k$) or 0 otherwise. Note that M contains the same information that the adjacency matrix of the graph except for the diagonal, which in this case will always be one instead of zero ($\mathrm{RMSD}_{ii} \equiv 0.0$). For the sake of simplicity, we will refer to M as the adjacency matrix of the trajectory graph.

By using the binary adjacency matrix, we reduce the RAM consumption of this object ($m = \frac{1}{8}$) by 16, 32, or 64 times compared to other software that deals with half, single or double-precision float values to represent the RMSD (see Table 1.2). Besides the RAM saving, expressing similarity as a binary matrix offers the possibility to perform the search of cliques using binary operators (AND and XOR, see Section 1.5.5), contributing to the

speedup of the heuristic clique search algorithm we propose in the following section.

4.1.4 . A heuristic search of big cliques

Next, we describe the workflow of the BitQT clustering algorithm, which is built upon a not previously published heuristic for searching big cliques. A formal review of the many MCP heuristics available is out of this thesis scope and can be found elsewhere (see reference 267). In our case, we want to keep the common similarity of QT clusters, but their size is not a big concern. After all, the original QT does not provide either maximum cliques.

We start with calculating the binary similarity matrix that will be stored in RAM. The float vector containing the one-versus-all RMSD similarity of each element is transformed into a bit-vector Bi (B1 to B9 in Matrix 1, Figure 4.2) in which $B_{ij} = 1$ if $\mathrm{RMSD}_{ij} \leq k$, zero otherwise. Each vertex's degree is calculated as the total number of switched-on positions in the Bi vector (D column in Matrix 1, Figure 4.2). Note that Bi vectors always have 1 at the i^{th} position ($\mathrm{RMSD}_{ii} == 0 \leq k$), so D column actually contain $degree+1$ of each vertex in the trajectory graph. Then, the subsequent steps are followed.

1- Vertex coloring: Each vertex of the input graph (Graph 1, Figure 4.2) is ranked (column R, Matrix 1, Figure 4.2) in descending order of their corresponding degrees (column D, Matrix 1, Figure 4.2. Following the rank order, each vertex takes a color label that it shares with all other vertices that are neither colored nor neighbors (column C, Matrix 1, Figure 4.2).

2- Clique search from the maximum degree node: After all vertices are colored, the search of a clique starts considering only neighbors of the maximum degree node of the graph (Graph 1A, Figure 4.2) which is called the seed of the clique (node 1 in Matrix 1A, Graph 1A, Figure 4.2). Neighbors of the seed are strictly ordered for further processing by following three criteria (DCg ordering); descending order of their degrees, ascending order of their color class, and ascending order of the degeneracy of the color class (columns D, C, and g, respectively, Matrix 1A, Figure 4.2). For our purposes, degeneracy is perceived as the number of nodes of the color class in the context of the neighbors of a seed node, not in the entire graph (in which case, using it for order would be meaningless).

Following this ordering, the first node is selected to start a clique, and subsequent nodes will be added to it if they have a still-not-explored color and are adjacent to previously explored nodes (clique propagation).

BitQT performs this search using bitwise operations. The bit-vector Bi corresponding to the maximum degree node is set as the clique bit-vector (B1 in Heuristic search of Graph 1A, Figure 4.2. Following the DCg ordering, an AND operation is performed between the clique bit-vector and the next node bit-vector if it has a new color (B6 in Heuristic search of Graph 1A, Figure 4.2. Indices corresponding to bits that become zero by this operation are discarded from further consideration (B2, B3, B4, and B5) as they are not adjacent to processed nodes (B1 and B6). The resulting bit-vector becomes the new clique bit-vector used for the AND operation, with the next candidate following the DCg ordering (B9). The bit-vector resulting from the iterative AND operations contains the members of the first clique.

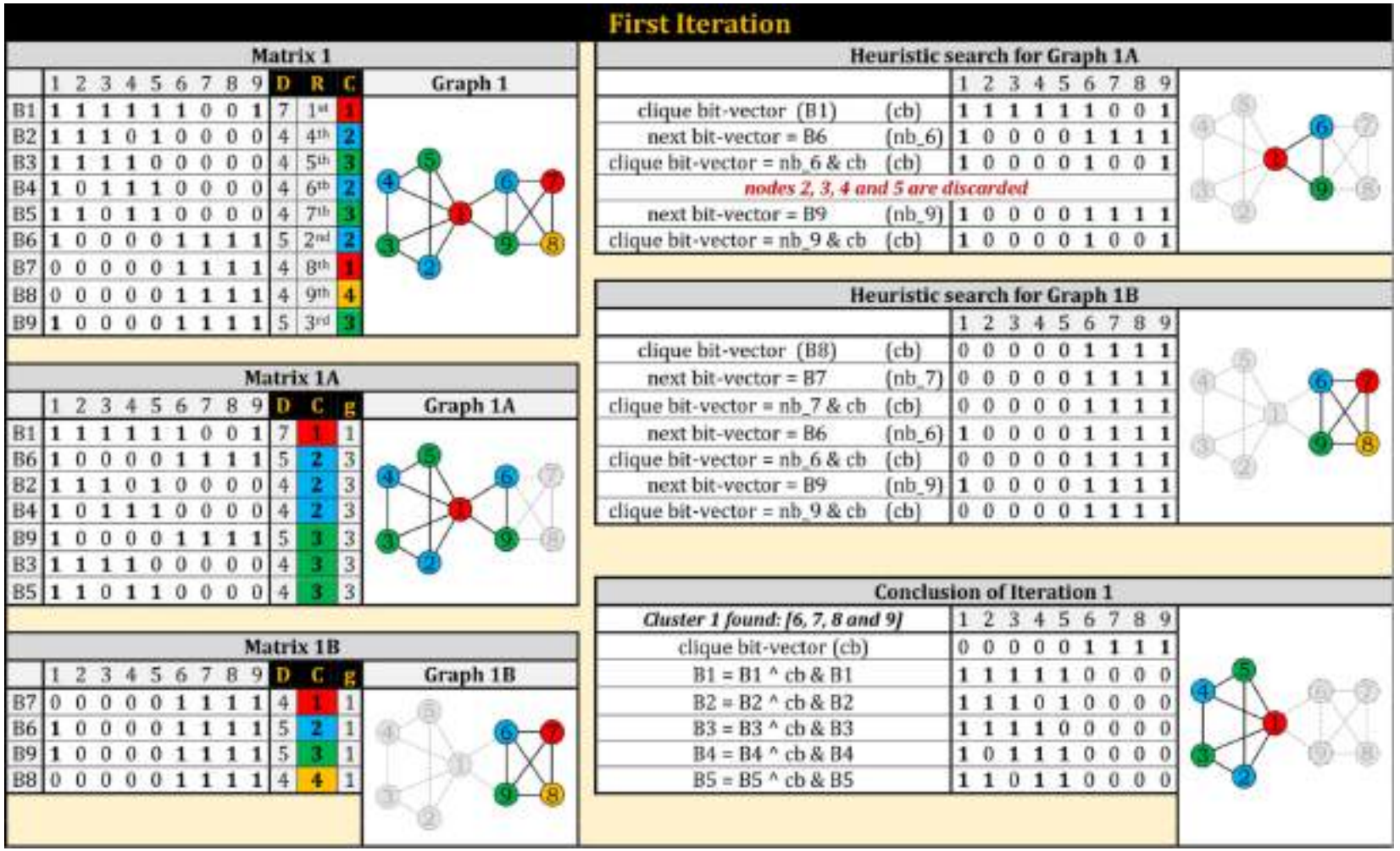


Figure 4.2: *First iteration of the binary heuristic for searching cliques implemented in BitQT.*

3- Clique search from promising nodes: Once the clique retrieved by using the maximum degree node as the seed is found in the previous step, the same exploration strategy is conducted for every promising node in the original graph (Graph 1). A promising node (B8 in Graph 1, Figure 4.2 is defined as a node with a color not present in the first clique and whose degree is higher than the number of nodes in the first clique. Using such nodes as seeds for propagation might lead to forming a more prominent clique (Heuristic search of Graph 1B, Figure 4.2).

4- Conclusion and updating: When the maximum degree node and all promising nodes have been used as seeds, the maximum clique found is picked as a cluster, and its members removed from the input graph (the corresponding Bi vectors removed from the binary matrix). An updating of the remaining bit-vector is necessary to set all entries corresponding to nodes that formed the cluster as zero, which will not be available for subsequent iterations. This updating is bitwise encoded as a consecutive AND/XOR operation between the remaining bit-vectors and the clique bit-vector (Conclusion of iteration 1, Figure 4.2. The same steps are repeated from Step 2 until no more cliques can be found.

During the execution of BitQT, some scenarios leading to ties may arise, for instance, selecting the node of the highest degree as seed (in "2-Clique search from the maximum degree node" and "3-Clique search from promising nodes"), or selecting the maximum clique (in "4-Conclusion and updating"). BitQT solves these cases by choosing the element with the lowest index among the available options as the "winner" of the tie. These ties can also appear in the original QT algorithm (when selecting the candidate cluster with most neighbors as a cluster). Choosing one or another "winner" does impact the outcome of algorithms in terms of cluster composition. However, the choice of a "winner" in a tied scenario will never invalidate the discussed guarantees of BitQT or QTPy.

4.1.5 . Performance benchmark of valid QT variants

In this section, we compare the run time and memory usage of BitQT, QTPy and *qtcluster*, which are the only QT implementations for molecular ensembles that we have found in the literature. These parameters are shown in Table 4.1 for the clustering of six different MD trajectories described in Section 2.2.1 (6, 30, 50, 100A, and 250 kF).

Given that software under evaluation is programmed by using distinct algorithms and programming languages (Fortran 90 for *qtcluster* and Python 3 for BitQT and QTPy), we are only able to provide general insights into the disparate performances observed in Table 4.1.

Table 4.1: *Run time and RAM consumption of analyzed QT implementations on different trajectories.*[1]

		BitQT		*qtcluster*		QTPy	
Traj. Name	# atoms (selection)	Run time *h:mm:ss*	RAM peak *GB*	Run time *h:mm:ss*	RAM peak *GB*	Run time *h:mm:ss*	RAM peak *GB*
6 kF	217 (all)	0:00:08	0.101	0:08:21	0.529	0:04:36	0.181
30 kF	64 (CA)	0:02:15	0.470	0:18:55	0.270	3:41:11	2.710
50 kF	78 (no H)	0:12:34	0.435	1:14:08	1.526	181:51:57	7.101
100A kF	660 (backbone)	1:15:37	4.355	**0:00:49**	**81.014**	**200:00:00**	**18.626**
250 kF	160 (backbone)	6:36:04	8.128	130:18:06	17.476	**0:00:03**	**116.415**

[1] Bold entries denote either a time crash (job taking more than 200 h) or a memory crash (job carrying more than 64 GB). In memory crash cases, the run time it took until crashing and an estimate of the lowest RAM needed to run the job is presented.

Of the three options, QTPy is the only one that always creates a square float matrix for saving the RMSD distances, so its RAM peak is expected to be the highest. The only exception is 6 kF, where the pairwise matrix uses only about 69 MB of RAM, so other data structures (or merely the molecular trajectory) will be responsible for the peak. RAM usage of BitQT also grows quadratically with the number of elements in the trajectory. However, as it uses bits instead of half-precision floats, there is a 16X memory saving in this object's construction compared to QTPy.

The memory usage of *qtcluster* may be confusing at first sight, as it can process a 250 kF trajectory but produces a memory crash when dealing with a simulation of 100 kF elements. It is clear why *qtcluster* crashed at 100A kF but could process 250 kF; the 100A kF trajectory contained 660 atoms and 250 kF only 160. Substituting in the V_{RAM} formula of qtcluster (Table 1.2) and taking $m = 4$ we obtain 81 GB for 100A kF and about 12 GB for 250 kF. Inconveniently, *qtcluster* can analyze big trajectories only when the number of selected atoms is relatively small.

While the three algorithms have quadratic spatial complexity, the costs of BitQT and QTPy are governed by the trajectory size. In contrast, *qtcluster* is dependent on the size of the atomic selection.

Run time reported in Table 4.1 exhibits a general trend; QTPy is the slowest choice, followed by *qtcluster*, which is greatly outperformed by BitQT. QTPy is the only one that implements the original version of QT[214]. As we have commented, the original QT has a very high computational cost evinced in the QTPy run times. The RMSD computation step can be safely discarded as the main contributor to the slow time performance of QTPy because it employs the same library that BitQT for this purpose (MDTraj)[257]. QTPy applications are limited to processing small trajectories or as a reference for developing future QT variants applied to the MD field.

qtcluster was designed as a high-speed QT alternative for the partitioning of MD. The similarity metric employed by this script (Equation 1.19) is cheaper than the more customary RMSD and avoids alignment. Perhaps the essential feature that makes *qtcluster* a fast QT implementation lies in the fact that it only preserves one condition from the originally formulated QT; that one assuring the collective similarity of retrieved clusters. For big trajectories, however,it is not a fast option.

Comparatively, BitQT has the best run time performance allowing it to handle rel-

atively long MD trajectories. The accelerated computing of optimal RMSD distances through the MDTraj engine joined to the developed binary-based heuristic for searching cliques are the cornerstones of its cheaper cost.

4.1.6 . Equivalence between BitQT and QT

As we discussed earlier in Section 4.1.2, BitQT was conveniently designed to relax the Condition 1 regarding the diameter of the cluster under construction, but it must carefully preserve the Condition 2 concerning the clusters collective similarity. The previous claim implies that all groups returned by BitQT must have a diameter less or equal than the user-defined quality threshold k.

Figure 4.3 shows the distribution of all clusters' diameter for every analyzed trajectory. It is appreciated that pairwise distances between elements of the same cluster never exceed the predefined quality threshold k (4 Å for 6 and 30 kF, 3 Å for 250 kF, and 2 Å for 50 and 100A kF).

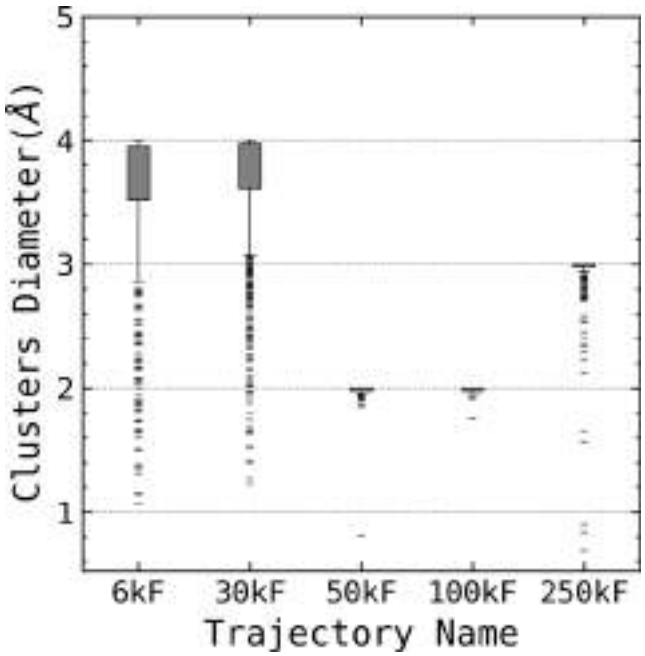


Figure 4.3: *Distributions of clusters diameters returned by BitQT for each analyzed trajectory.*

Figure 4.3 also demonstrates that BitQT clusters are cliques in the MD trajectory graph. As mentioned in Section 4.1.3, an edge between two nodes i and j of the trajectory graph is set if and only if $d_{ij} \leq k$. If all pairwise distances between elements in every cluster are under k, then the corresponding nodes of the trajectory graph are pairwise connected, implying that clusters are indeed cliques.

The respect for collective similarity indicates an essential equivalence between BitQT and QT, but it does not quantitatively compare the outcomes of these algorithms. An Adjusted Rand Index (ARI) analysis between partitions obtained with QTPy (Q) and

BitQT (B) for trajectories 6, 30, and 50 kF is shown at Figure 4.4. Note that instead of reporting just the global ARI between Q and B, we explicitly compared the ARI between both partitions at the top-X clusters (Q_X and B_X), taking X from 1 (the first cluster) to C (the total number of clusters). Consequently, the global ARI between Q and B corresponds to the last point of each curve. The remaining points indicate the correspondence between the first X clusters of Q and B.

The global ARI for 6, 30, and 50 kF are 0.87, 0.87, and 0.91 respectively, indicating a good agreement between clusters produced by QTPy and BitQT. An even higher index is reported for the first X clusters with sizes bigger than 1% of the trajectory size ($\mathrm{ARI}_{1\%}$). These most populated clusters are often considered the most relevant of the trajectory as they groups the representative conformational states explored in an MD simulation. $\mathrm{ARI}_{1\%}$ (represented by a bold point in Figure 4.4A-C) is 0.96, 0.88 and 0.88 for trajectories 6, 30, and 50 kF, respectively. This indicates an excellent agreement between the most popular clusters obtained by QTPy and BitQT.

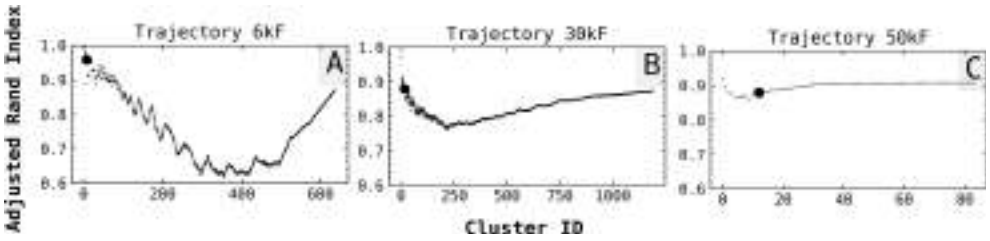


Figure 4.4: *Adjusted Rand Index (ARI) between partitions obtained with QTPy and BitQT for all clusters in trajectories 6 kF, 30 kF, and 50 kF (A, B, and C respectively). The ARI of the first X clusters with sizes bigger than 1% of the trajectory size is located to the left of the bold point of each graph.*

Observed ARI fluctuations at different top-X are expected because both algorithms pick their seeds to form clusters differently. It is possible that at a given value of X, clusters formed by QTPy were still not recovered by BitQT or *vice versa*. However, fluctuations are more pronounced for the less populated clusters.

4.2 . BitClust: the first binary implementation of Daura clustering

The advantages and limitations of Daura algorithm are reviewed in Section 1.6.2.2. This methodology stands up as a compromise alternative to QT when big molecular ensembles must be processed within a reasonable run time. However, after carefully inspecting its most popular alternatives, we noted that they were either inefficient (in terms of run time and RAM consumption) or even inaccurate.

Hence, we propose BitClust, a novel, faster implementation of Daura algorithm designed to process big molecular ensembles. BitClust offers a classic trade-off, RAM for speed, to boost time performance by increasing memory storage. However, the necessary amount of RAM has been significantly optimized by encoding pairwise similarity

distances as bits.

4.2.1 . Translating Daura clustering to bitwise operations

The first step in BitClust is similar to that of BitQT; the binary encoding of RMSD pairwise similarity. Both implementations of these objects encoding are analogous and already discussed in Section 4.1.3.

With the binary matrix loaded in RAM, two special vectors are constructed from the beginning of the clustering process and iteratively updated during the following steps. Firstly, a bit vector A is declared (iteration 1, Figure 4.5), which contains as 1 those indices available to form a cluster at a given moment. In the beginning, all A bits are set to 1 as all elements are available to form clusters. As the algorithm iterates, indices already clustered are set to 0 in A, reflecting that they are no longer available for consideration to form other clusters.

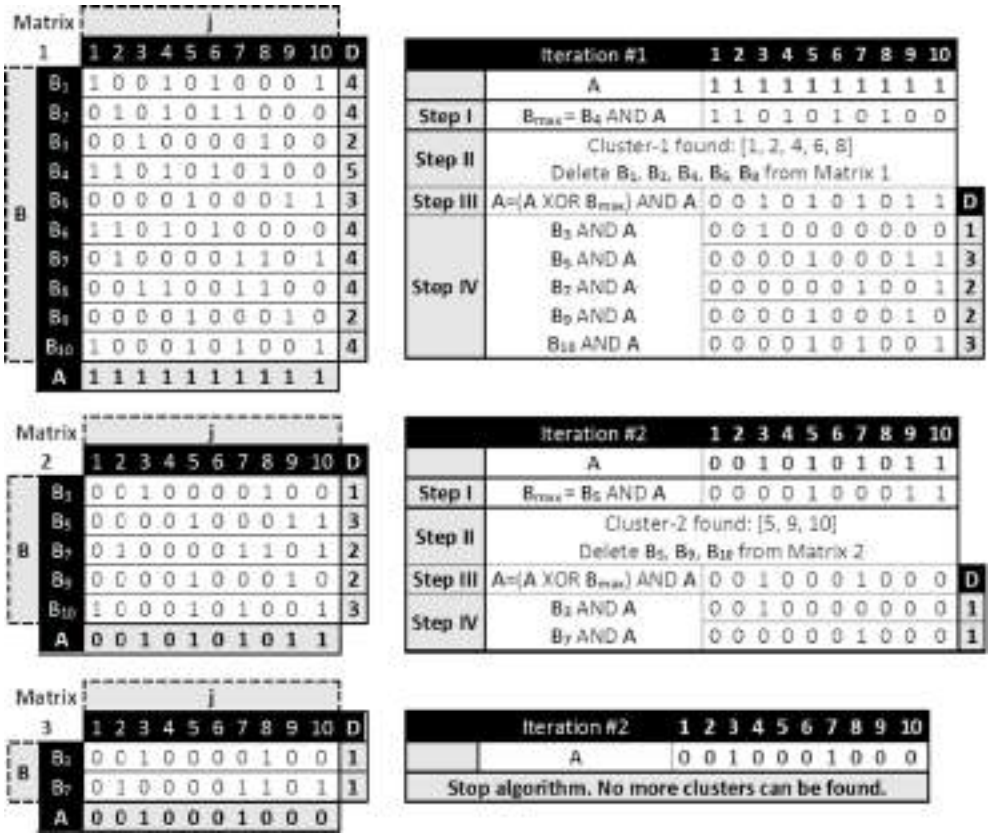


Figure 4.5: *Workflow of the binary Daura algorithm implemented in the BitClust code*

Secondly, a vector D of integers is declared (matrix 1, Figure 4.5), containing the degree (Di) of each Bi present in the matrix. Di is defined as the total number of indices 1 in the bit vector that results from the bitwise operation $Bi\ AND\ A$ ($Di = sum(Bi\ AND\ A)$) and not as the total number of indices 1 of Bi. The vector A here acts as a bit-mask that helps to elucidate how many elements of Bi are available to form

clusters. The initial bit values of Bi are never altered, only A and D change between iterations (matrix 1, 2 and 3 in Figure 4.5).

Once A and D vectors have been initialized, the workflow of BitClust has four main steps that repeat iteratively until no more elements can be clustered: (i) detect Bi with the maximum degree and define as $Bmax$ the bit vector resulting from the binary operation $Bi\ AND\ A$ (step I, Figure 4.5).

Note that if two or more Bi vectors have the same maximum degree, the algorithm arbitrarily processes the Bi with the lowest index i as if it were the maximum, (ii) saves all set indices of $Bmax$ as members of a cluster and delete their corresponding Bi vectors from the matrix (step II, Figure 4.5), (iii) update the bit vector of available conformations A by setting as 0 those entries that were clustered in the previous step (step III, Figure 4.5), and (iv) update degrees of remaining Bi (step IV, Figure 4.5).

4.2.2 . Performance benchmark of Daura variants

A set of commonly used software for clustering MD simulations has been chosen for performance comparison against BitClust. Run time and memory consumption of each method is reported in Table 4.2 for the three trajectories 6, 100B, and 500 kF (see Section 2.2.1).

The clustering method selected in each case was as follows; Daura for BitClust and GROMACS (through the gromos option), quality threshold for py-MS and VMD, qt-like for WORDOM and median-linkage for TTClust.

We want to stress that despite the native denomination in their original software, the chosen algorithms (except the median-linkage) correspond all to Daura (see Section 4.1.1). In the case of VMD, we decided to show the performance of processing five (VMD-5, the default value) and all (VMD-ALL) clusters to evaluate the usefulness of its implementation, which is specially conceived for preserving memory resources.

Table 4.2: *Run time and RAM consumption of analyzed Daura implementations on different trajectories.*[1]

Software	Trajectory 6 kF		Trajectory 100B kF		Trajectory 500 kF	
	Run time *mm:ss*	RAM Peak *GB*	Run time *hh:mm:ss*	RAM Peak *GB*	Run time *hh:mm:ss*	RAM Peak *GB*
BitClust	0:04	0.15	1:25:28	9.41	6:00:08	33.84
pyMS	0:11	0.41	1:46:02	61.54	**3:03:49**	**— [2]**
WORDOM	1:52	0.10	**1:26:13**	**37.25**	**2:54:23**	**931.32**
VMD-5	0:14	0.09	4:59:22	6.97	29:34:43	4.13
VMD-all	2:05	0.10	5:08:10	6.97	**60:00:00**	4.14
TTClust	1:30	1.16	**0:01:59**	**74.51**	**0:00:37**	**1862.65**
GROMACS	2:10	0.16	26:13:29	52.22	**01:09:03**	**931.32**

[1] VMD-5 and VMD-ALL notations refer to clustering jobs performed using VMD software and requesting five and all clusters, respectively. Bold entries denote either a time crash (jobs taking more than 72 h) or a memory crash (jobs carrying more than 64 GB). In memory crash cases, the run time it took until crashing and an estimate of the lowest RAM needed to run the job is presented. [2] Impossible to determine as this algorithm does not load a matrix in RAM.

The calculation of the pairwise similarity values occurs in the first place for every soft-

ware. This is the most CPU-consuming step that governs overall run time. GROMACS is the slowest option (6 and 100B kF trajectories for GROMACS in Table 4.2) because its implementation of this first step is not parallelized.

Run times for processing five and all clusters with VMD (VMD-5 and VMD-ALL respectively in Table 4.2) are significantly different. This behavior is associated with the fact that VMD does not save the pairwise similarity information, which is calculated to retrieve a particular cluster. Instead, it recalculates all the information every time a cluster is found. While this characteristic puts VMD as the most memory-efficient alternative among the studied ones, it is the cause of its time performance of VMD when more than a few clusters are requested.

Suppose the trajectory has few clusters grouping most of the elements (trajectory 100B kF in Table 4.2). In that case, the recalculation takes less time as fewer elements are available to form clusters (VMD-ALL, Table 4.2). However, suppose many evenly distributed clusters are present in the trajectory (trajectories 6 and 500 kF in Table 4.2). In that case, recalculations are more time-consuming as many elements are available to form clusters after each iteration (VMD-ALL, Table 4.2).

WORDOM has an intermediate time performance in the trajectory it could handle (trajectory 6 kF for WORDOM in Table 4.2). The faster options are BitClust and py-MS implementations which can interface to the fast MDTraj engine[257] for the calculation of the pairwise similarity matrix. Even though TTClust does not implement Daura, it interfaces with MDTraj for calculating the RMSD matrix. We included it to illustrate some aspects of memory management.

Among the analyzed options, the most memory-consuming is TTClust. It saves the pairwise similarity matrix directly in RAM using a 64-bit float for every entry ($m = 8$ in V_{RAM} equation of Table 1.2). For short trajectories (as TTClust authors point out in their manuscript), this is affordable for most workstations, but as trajectory size gets bigger, TTClust starts showing a prohibitively RAM consumption.

Although it also employs a 64-bit float for every entry of the similarity matrix, py-MS has an improved approach. It saves the square matrix to a temporary file in the disk and loads in RAM only those values having an RMSD distance less than the specified cutoff. While the py-MS strategy can save RAM, enough space must be available on disk. In the worst case (for higher values of cutoff leading to a situation where all elements are neighbors), the whole file is loaded to memory, incurring the same expensive behavior that TTClust.

GROMACS saves the similarity matrix into memory using a 32-bit float ($m = 4$ in V_{RAM} equation of Table 1.2) for each entry, requiring more than 50 GB to process the 100B kF trajectory (see Table 4.2). WORDOM saves the matrix using the same numeric type (32-bit float) that GROMACS. However, other internal data structures make it a more memory-consuming alternative, impeding the processing of the 100B kF trajectory.

VMD is the unique alternative that does not create a matrix in RAM or the disk. It only retains the most significant cluster found during iterations. As a result, it must recalculate redundant information every time a cluster is found, leading to poor time

performance when many clusters are requested. Note that this is the only variant that requires the number of clusters to retrieve as a mandatory argument.

As discussed in Section 4.2.1, BitClust saves the whole square pairwise similarity matrix directly in RAM. Nevertheless, it does not use float values but bits, encoding if a particular entry has an RMSD value less equal or greater than the specified cutoff. The use of bits results in a significant reduction of the memory requirement of the RMSD matrix. The less precise float (half-precision float) consumes 2 bytes or 16 bits to represent a value. On the other hand, BitClust only needs 1 bit, as discussed earlier, so a saving of 16X is achieved for the total resources consumed by the matrix. It is worth noting that most implementations use 4 or 8 bytes (32 or 64-bit) for the single and double precision float values, which means that BitClust builds an RMSD matrix that saves RAM by a factor of 32X or 64X concerning such variants.

4.2.3 . Equivalence between BitClust and Daura

In order to assess the correctness of BitClust implementation, we compared elements of the first five clusters retrieved by Daura implementations from the 6 kF trajectory. In Table 4.3, it is reported not only the number of elements of every cluster but also the percentage of elements contained in the corresponding cluster returned by BitClust. To effectively compare the resemblance of two conformations, they should be superposed before the RMSD calculation; however, some analyses might require ignoring this superposition step. In the case of WORDOM, both options were reported in Table 4.3.

Table 4.3: *Number of elements and percent of elements shared by the first five clusters retrieved by Daura alternatives against BitClust.*[1]

Cluster ID	No. of elements (% intersection)					
	BitClust	pyMS	WORDOM (superpose)	WORDOM (no superpose)	VMD	GROMACS
1	465 (100)	465 (100)	465 (100)	444 (100)	444 (100)	628 (6)
2	346 (100)	346 (100)	346 (100)	343 (100)	343 (100)	380 (16)
3	153 (100)	153 (100)	153 (100)	141 (86)	141 (86)	205 (9)
4	146 (100)	146 (100)	146 (100)	140 (90)	140 (90)	202 (1)
5	136 (100)	136 (100)	136 (100)	133 (100)	133 (100)	187 (1)

[1] Superposition of elements before RMSD calculation was explicitly requested in the case of WORDOM (superpose).

BitClust, py-MS, and WORDOM (superpose) retrieve exactly the same information for the first five cluster reported (see Table 4.3). It is worth noting that WORDOM qualifies its qt-like method as a variant of the QT algorithm, and py-MS affirms to be performing QT. This comparison further asserts that WORDOM and py-MS implement Daura algorithm (see discussion in Section 4.1). VMD documentation also reports the application of the QT algorithm. However, the comparison between VMD and WORDOM (no-superpose) in Table 4.3 demonstrates that VMD results coincide with those coming from a Daura implementation when no superposition is performed on the analyzed elements.

GROMACS results invite a detailed analysis as none of the retrieved clusters exhibited a similarity higher than 20% to BitClust. It should be noted that only the GRO-

MACS documentation explicitly recognizes Daura algorithm application through the gromos method, referring the readers to the original publication of this routine. However, we think the analyzed version of gromos algorithm is incorrectly implemented. From the 6000 elements analyzed in trajectory 6 kF, only 1000 are reported in the output file. In addition, all conformations appeared six times in different clusters except for the first (index zero), which appeared seven times in the output file. After these inconsistencies, it is an unreliable option. Besides our findings, a descriptive report in the literature explains in detail the implementation pitfall contained in the source code of this ubiquitous tool (`https://mailman-1.sys.kth.se/pipermail/gromacs.org_gmx-users/2015-April/096367.html`).

4.3 . DP+: Reaching linear spatial complexity in DP clustering

The theoretical background of the DP algorithm is discussed in Section 1.6.2.3. Even though a myriad of variants exists for this clustering procedure, none has been able to eliminate its quadratic memory complexity so far. Here we propose DP+, a methodology to derive the exact DP partitioning of elements without constructing a square similarity matrix. Instead, a double-heap approach produces an oriented tree where every node is connected to its nearest neighbor of higher density by a weighted edge.

Built upon DP+, we designed RCDPeaks, a refined variant of the original DP. Employing DP+, RCDPeaks processed a one-million elements trajectory using less than 4.5 GB of RAM, a task that would have taken more than 2 TB (and about 3X more time) with the most competitive alternative.

4.3.1 . Computing an oriented tree instead of a complete graph

The typical workflow used in exact or modified DP variants saves the pairwise similarity of elements into a square float matrix. This strategy may offer a fast determination of ρ_i and δ_i (see Section 1.6.2.3) but inconveniently limits the algorithm's application to problems whose similarity matrix could fit in available RAM. Next, we describe DP+, an alternative approach to DP that avoids the construction and storage of such a matrix and hence can be applied to treat much longer ensembles.

DP+ exploits the graph-theoretical view of a molecular ensemble by considering it as a graph T in which all nodes are pairwise connected. In T, nodes represent 3D coordinates, and their pairwise similarity distance weights undirected edges (Figure 4.6A). If ρ values are assigned as the weights of T nodes, then the goal of DP can be stated as transforming T into an oriented tree T' that contains only one outgoing edge per node pointing to its nearest neighbor of higher ρ. The weights of edges in T' correspond to δ values in Equation 1.25 (Figure 4.6B).

For every element i, DP+ computes ρ_i from the i-versus-all RMSD vector (RMSD_{ix}), by counting the number of elements j whose $\mathrm{RMSD}_{ij} < d_c$. As δ_i refers to the distance from i to its nearest neighbor of higher ρ, computing this magnitude requires iterative queries to the sorted RMSD_{ix} vector. However, the complete sorting of RMSD_{ix} is an expensive $O(n * log(n))$ operation. DP+ makes a faster partial ordering ($O(n)$ time

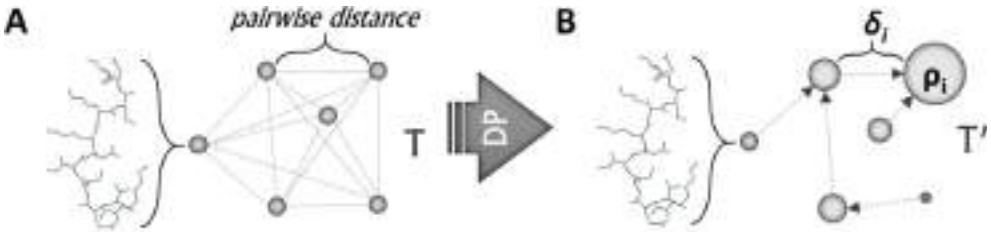


Figure 4.6: *Graph-theoretical view of an MD trajectory before and after applying DP. A-) Complete graph T in which nodes correspond to 3D coordinates and undirected edges denote pairwise similarity B-) Oriented tree T' obtained after applying DP to T. Each node (weighted by its ρ value) contains a single outgoing edge pointing to its nearest neighbor of higher density.*

complexity) of RMSD_{ix} at the k^{th} position and then a complete ordering of the much smaller k-neighborhood (denoted as η from now on). The value of k is internally defined as $0.02 * N$ (although users can modify it), where N is the total number of elements in the ensemble. DP+ relies on the assumption that most elements will find their nearest neighbor of higher ρ inside this sorted η.

Figure 4.7 illustrates the previous procedure using the RMSD_{0x} vector of a ten-elements trajectory where $d_c = 0.36$ nm and $k = 5$. In Figure 4.7A, ρ_0 (the number of elements j for which $\mathrm{RMSD}_{0j} < d_c$) is set as 7 (bold entries). In 4.7B, the partial sorting of RMSD_{0x} at $k = 5$ is exemplified. Note that this process returns the first unsorted k elements with the lowest values. Figure 4.7C shows the last ordering stage in which only the first k elements of RMSD_{0x} are completely sorted. This vector corresponds to η_i.

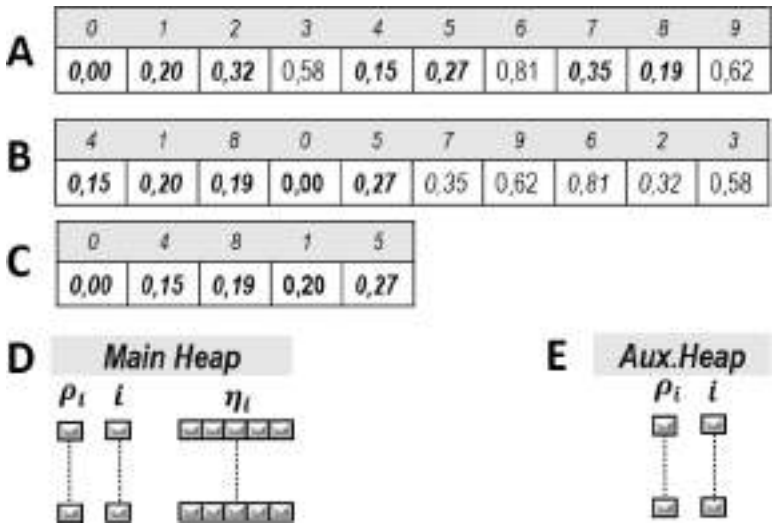


Figure 4.7: *DP+ main objects and operations involved in the computation of ρ_i and η_i for a ten-elements ensemble ($d_c = 0.36\ nm$ and $k = 5$). A-) RMSD_{0x} vector. Bold entries correspond to elements closer than d_c from element o. B-) RMSD_{0x} partially sorted at $k = 5$. C-) Complete ordering of first k values of RMSD_{0x} (η_0). D-) Main heap. E-) Auxiliary heap.*

DP+ gradually constructs T' using data distributed in two separate heaps to avoid storing T information as a square matrix. The main heap will contain the ρ_i, i, and η_i for a subset of elements (Figure 4.7D), while an auxiliary heap will store those elements whose nearest neighbor of higher density could not be found inside their η_i (Figure 4.7E).

The importance of using a heap data structure lies in its ability to quickly retrieve an extreme value (minimum in our case) of the collections it contains. If we introduce several tuples containing ρ_i and η_i, a so-called "min heap" can return the minimum weighted element and its corresponding η_i in logarithmic time. Through heaps, DP+ speeds up the construction of T', exploiting the observation that elements with lower ρ are more likely to find their nearest neighbor of higher density inside η.

Concretely, after defining a local density cutoff d_c, DP+ follows the next steps to construct T' (see Algorithms 6 and 5): A still not analyzed element i is chosen from the trajectory. This action will occur whenever the main heap is empty. RMSD_{ix} is then calculated and ρ_i computed counting the number of elements j with $\mathrm{RMSD}_{ij} < d_c$. Through the already mentioned sorting strategy, η_i is obtained and DP+ proceeds to search the first element $X_j \in \eta_i$ having $\rho_j > \rho_i$. If such an element is found, a directed edge from i to j is created, and δ_i is set to d_{ij}. During this process, all inspected j for which $\rho_j \leq \rho_i$ are transferred to the main heap as a tuple containing ρ_j, j index and η_j. If the opposite situation happens, *i.e.*, an element j whose $\rho_j > \rho_i$ is not found in η_i, then a tuple containing ρ_i and i index is passed to a secondary heap for future processing. The previous process goes on until all elements have been considered.

At that point, the elements i not finding their nearest neighbor inside η_i are already stored in the auxiliary heap. For each one of them, DP+ recalculates RMSD_{ix} and finds the element j with $\rho_j > \rho_i$ to set δ_i. In the special case where i has the maximum value of ρ (so it is impossible to find $\rho_j > \rho_i$), δ_i is set to $max(\mathrm{RMSD}_{ix})$. Experiments show that the average size of the auxiliary heap is always a small percent of N.

4.3.2 . Refining the exact algorithm of DP

As explained in Section 4.3.1, DP+ is an exact implementation of the original DP. DP+ avoids the quadratic memory complexity using heap-based data structures. Having equivalent results, DP and DP+ share the same shortcomings, among which are: (i) the consideration of very similar center candidates as independent cluster seeds (in the user-selected region of the decision graph, no checking is performed on centers to ensure their pairwise geometrical separation), (ii) the impossibility of running an automatic job (given that ρ and δ cutoffs must be manually selected from the decision graph), and (iii) the excessive flexibility of core and halo definitions for applications regarding molecular ensembles (see Figure 9.10). In Section 9.2.4, we describe Refined-Core Density Peaks (RCDPeaks), built upon DP+, and addressing the limitations mentioned above.

4.3.3 . Performance benchmark of DP variants

The run time and RAM consumption of RCDPeaks, cpptraj, and CLoNe when processing different MD trajectories are compared in Table 4.4. To our knowledge, these three software are the only publicly available DP implementations specifically designed

to deal with MD simulations. While cpptraj implements the original algorithm, CLoNe was inspired by DP to overcome several of its limitations.

Table 4.4: *Run time and RAM consumption of analyzed DP implementations.*[1]

		RCDPeaks		cpptraj		CLoNe		
Trajectory	No. of atoms (selection)	Run time *h:mm:ss*	RAM peak *GB*	Run time *h:mm:ss*	RAM peak *GB*	Run time *h:mm:ss*	RAM peak *GB*	Disk space GB
6 kF	217 (all)	0:00:05	0.14	0:00:10	0.09	0:00:40	2.35	0.21
30 kF	64 (CA)	0:00:42	0.16	0:01:46	1.78	0:23:22	39.72	6.30
50 kF	78 (no H)	0:02:00	0.19	0:05:59	4.71	**0:11:29**	**64.00**	**15.00**
100A kF	660 (backbone)	0:41:59	0.92	2:10:23	19.38	**NR**	**74.51**	**57.22**
250 kF	160 (backbone)	1:14:04	0.87	**0:00:04**	**125.50**	**NR**	**465.66**	**359.06**
500 kF	217 (all)	6:47:12	2.03	**0:00:09**	**499.99**	**NR**	**1862.65**	**1430.51**
1 MF	304 (backbone)	33:21:10	4.16	**0:00:26**	**2048**	**NR**	**7452.07**	**5723.20**

[1] Bold entries denote a memory crash (jobs carrying more than 64 GB). The run time software took until crashing and an estimate of the lowest RAM needed to run the job (and also the amount of HDD space for CLoNe) is presented. **NR** means Not Ran Job.

As shown in Table 4.4, CLoNe has the highest RAM consumption, which only permitted processing the small trajectories of 6 and 30 kF. This variant also uses substantial disk space resources if the similarity metric is not euclidean (RMSD in our case), as the user must provide a text file with the pairwise similarity information. Although CLoNe also has the slowest run time (about 30X slower than RCDPeaks for the 30 kF trajectory), this is not a critical aspect when dealing with the short trajectories it can manage.

The cpptraj alternative is considerably less RAM consuming than CLoNe. The memory peak for each analyzed trajectory roughly corresponds to the storage of a half-precision float square matrix (pairwise RMSD information). For short and medium-sized MD trajectories (see 100A kF in Table 4.4), cpptraj has an affordable memory cost. However, if relatively long trajectories must be processed, the quadratic RAM complexity of cpptraj becomes a major limitation. Regarding run time, cpptraj is also faster than CLoNe but still about 3X slower than RCDPeaks. It is worth noting that developers of cpptraj have marked their implementation as experimental. This software will produce neither the calculated clusters' core nor the boundary regions.

The fastest and the most memory-efficient software is RCDPeaks. The key factors contributing to the speed up of this variant are the use of MDTraj[257] for computing the optimal RMSD distances and, to a lesser extent, the sorting procedure to get η_i (see Section 4.3.1). On the other hand, the RAM consumption of RCDPeaks is remarkably low, mainly due to the small size of the main heap (see Section 4.3.1).

4.4 . MDSCAN: efficient RMSD-based HDBSCAN

We highlighted in Section 1.6.2.4 that we found no report of an HDBSCAN implementation specifically designed for handling molecular ensembles. Although a deeply optimized and widely spread software exists for conducting HDBSCAN analysis using several low-dimensional metrics (termed as HDBSCAN* by its authors), it excludes the RMSD, a metric that remains the *de facto* choice in molecular similarity analyses. HDB-

SCAN* can receive a pre-processed RMSD float square matrix, but its explicitly fixed double-precision float data type bounds the range of applications to tiny datasets.

Next, we propose MDSCAN, a fast and memory-efficient RMSD-based implementation of HDBSCAN that is suitable for processing big molecular datasets with no distance matrix involved. MDSCAN, similar to HDBSCAN*, is an approximate approach to the reference implementation[236] whose moderate deviations make a suitable compromise between the computational cost of the clustering job and the quality of returned clusters.

The encoding of molecular ensembles as a distinctive variant of VP-TREEs that join leaves into buckets of elements or sub-datasets (decreasing the run times of RMSD computation up to a half), and a double-heap approach to calculate a quasi-minimum spanning tree of the MD trajectory's complete graph (significantly decreasing the RAM usage) are the significant methodological contributions of this work.

4.4.1 . Dual-heap construction of a quasi-MST

Although HDBSCAN can be helpful in multiple fields, we will focus on how simple observations lead to substantial memory savings of optimal RMSD-based implementations for processing big datasets.

In Section 1.6.2.4, $\kappa(i)$ was defined as the distance from i to its k^{th} nearest neighbor. Consequently, the k-neighborhood of i (denoted by $\eta(i)$) can be defined as the set of nodes j for which $d(i,j) \leq \kappa(i)$. As a derivation of Equation 1.26, it can be stated that: Any node j belonging to $\eta(i)$ and having a core distance $\kappa(j) \leq \kappa(i)$, leads to a minimum value of $d_{mr}(i,j) = \kappa(i)$ (Equation 4.1).

$$d_{mr}(i,j) = \{\kappa(i) \mid \forall j \in \eta(i) : \kappa(j) \leq \kappa(i)\} \tag{4.1}$$

This means that if we would like to construct an MST from T nodes, joining i to any node $j \in \eta(i)$ with $\kappa(j) \leq \kappa(i)$ would create a minimum weighted edge of that MST. In the hypothetical case in which all nodes of T get connected as a tree following the particular case in Equation 4.1, we would end up with an MST from which the final steps of HDBSCAN may continue (see Section 1.6.2.4).

Although possible, the described scenario is unlikely to occur for a real dataset. The principal assumption of MDSCAN is that it happens for most nodes, generating not an MST but a Minimum Spanning Forest (MSF), a collection of disconnected minimum spanning trees of T nodes. The number of MST inside the MSF equals the number of nodes that could not find another node j satisfying Equation 4.1. For those disconnected i, it is still possible to find a node j in one of the MST $\in$ MSF whose $d_{mr}(i,j)$ though not minimal, would be small enough.

Joining all disconnected nodes in the MSF without creating cycles gives a quasi-MST (instead of an exact one). This quasi-MST could be used later to perform subsequent steps of the original formulation of HDBSCAN. The capital importance of this workflow to get a quasi-MST is that no similarity matrix must be stored in RAM.

Nodes i more likely to find a neighbor j satisfying Equation 4.1 are those with a high value of $\kappa(i)$. As MDSCAN continuously checks for Equation 4.1 to hold, we used a heap

data structure that retrieves the node with maximum $\kappa(i)$ found so far in logarithmic time. MDSCAN uses two heaps; the first (main heap) will contain some next-to-analyze nodes, while an auxiliary heap involves already analyzed nodes failing Equation 4.1 that will be re-processed after exhaustion of the main heap.

The algorithm starts by randomly choosing a not-analyzed node i from the dataset. This will occur whenever the main heap is empty, so it is impossible to choose its first element (highest core distance found so far). To retrieve the $\eta(i)$ and $\kappa(i)$ of every node, MDSCAN queries the vantage tree data structure described in Section 1.5.4. Then, the software searches nodes $j \in \eta(i)$ having $\kappa(j) \leq \kappa(i)$. If such a node is found, a directed edge from i to j is created, and its weight set as $d_{mr}(i,j) = \kappa(i)$.

During this process, all inspected j for which $\kappa(j) \geq \kappa(i)$ are transferred to the main heap as a tuple containing $\kappa(j)$, j index, and $\eta(j)$. If the opposite situation happens, i.e. a node j whose $\kappa(j) \leq \kappa(i)$ is not found in $\eta(i)$, then a tuple containing $\kappa(i)$ and i index is passed to an auxiliary heap for future processing. The previous process goes on until consideration of all nodes. At the end of this stage, an MSF is achieved. The deviation from an exact MST will thus arise from the connection of nodes in the auxiliary heap. The less populated this heap is, the smaller the deviation.

To join the remaining nodes at the auxiliary heap, MDSCAN runs the following steps for each. First, the RMSD_{ix} vector, containing distances from i to all other nodes in T, is calculated. Then, the $d_{mr}(i,j)$ is calculated for all nodes j that are not in the same tree that i. The smallest value of these MRD is taken as the weight of a directed connection from i to j. Once the quasi-MST is constructed, MDSCAN continues with the building of a cluster hierarchy and the extraction of the most stable clusters just as it was described in Section 1.6.2.4 (see Algorithm 4 and Figure 1.13).

4.4.2 . Performance benchmark of HDBSCAN variants

HDBSCAN* implementations were compared against MDSCAN in terms of run time and memory consumption: (i) the HDBSCAN* 's generic option using RMSD, (ii) the HDBSCAN* 's generic options using Euclidean distance, and (iii) the HDBSCAN*'s Prim option using Euclidean metric. The Prim and generic labels of HDBSCAN* refer to the approach followed for constructing the quasi-MST (see Annex 9.2.5). Note that if the Prim algorithm is specified, it is impossible inHDBSCAN* to pass a similarity matrix.

In Table 4.5 it is appreciated that MDSCAN is the fastest option in all cases except for the smallest 6 kF dataset, where the effort of constructing a vp^b-tree is significant compared to the time taken by the computation of pairwise similarities. MDSCAN's time efficiency comes mainly from the accelerated RMSD computations offered by the MDTraj suite and from our vp^b-tree encoding. Indeed, results showed that while the quasi-MST's weight computed by MDSCAN with and without using vp^b-tree is equivalent, the run time decreases up to a half when the VP-TREE encoding is exploited (see Annex 9.2.6).

Table 4.5: *Run time and memory consumption of* HDBSCAN* vs. *MDSCAN on different datasets.*[1]

			MDSCAN [RMSD]		HDBSCAN* [generic-RMSD]		HDBSCAN* [generic-Euclidean]		HDBSCAN* [Prim-Euclidean]	
Traj. Name	Traj. Size *(GB)*	# Atoms *(selection)* [2]	Run time *(hh:mm:ss)*	RAM peak *(GB)*	Run time *(hh:mm:ss)*	RAM peak *(GB)*	Run time *(hh:mm:ss)*	RAM peak *(GB)*	Run time *(hh:mm:ss)*	RAM peak *(GB)*
6 kF	0.02	217 (all)	0:00:06	0.18	0:00:04	1.49	0:00:02	1.29	0:00:30	0.15
30 kF	0.02	64 (CA)	0:00:19	0.21	0:00:53	35.90	0:00:25	29.01	0:02:14	0.18
50 kF	0.05	78 (no H)	0:01:37	0.26	**0:00:02**	**83.82**	**0:00:44**	**74.60**	0:04:28	0.26
100A kF	0.75	660 (back)	0:36:42	1.78	**0:00:09**	**335.28**	**0:00:02**	**299.49**	2:31:14	2.58
250 kF	0.47	160 (back)	0:37:46	1.20	**0:00:03**	**2103.87**	**0:00:02**	**1863.54**	5:52:37	1.64
500 kF	1.25	217 (all)	6:42:18	2.83	**0:00:08**	**8381.90**	**0:00:03**	**7453.01**	21:00:04	4.41
1 MF	3.47	304 (back)	21:01:06	7.67	**0:00:35**	**33534.32**	**0:00:06**	**29809.12**	**72:00:00**	**14.23**

[1] Bold entries denote either a time crash (jobs taking more than 72 h) or a memory crash (jobs carrying more than 64 GB). In memory crash cases, the run time it took until crashing and an estimate of the lowest RAM needed to run the job is presented. [2] all: all atoms, CA: alpha carbon atoms, no H: non-hydrogen atoms, back: backbone atoms.

While the generic implementations of HDBSCAN* (RMSD and Euclidean-based) have run times comparable to MDSCAN (Table 4.5), their high RAM consumption only permitted them to process the two smallest datasets. HDBSCAN* 's generic implementations are fast primarily because they use a single-linkage approach to get a tree and do not provide an exact MST.

The HDBSCAN* 's Prim-Euclidean alternative could analyze all datasets (except the 1 MF job that was stopped after running for 72h) but taking up to nine more times than MDSCAN (see Traj.250 kF in Table 4.5). This option neither constructs an exact MST from the input data, though a less simplistic approach than a single-linkage is followed to construct a tree (see Annex 9.2.5).

Regarding the RAM management, the only efficient alternatives are MDSCAN and HDBSCAN* 's Prim-Euclidean, which does not employ square pairwise similarity matrices to derive the required tree. They both have a similar consumption for the smallest datasets (from 6 to 50 kF, Table 4.5), with MDSCAN displaying the best behavior for the longest cases (from 100A kF to 1 MF, Table 4.5).

Within MDSCAN, the RAM is consumed mainly by the dataset file. One copy of this object is needed to instantiate the vp^b-tree data structure and to make the similarity recalculations to complete the final quasi-MST. Another copy is created when producing the vantage tree's buckets. As it is shown in Annex 9.2.6, using MDSCAN without a vp^b-tree encoding is more memory-friendly (as the second copy never gets created). However, we are persuaded that this small memory trade-off is justified given the speed reached with the VP-based alternative.

The generic-RMSD and generic-Euclidean options of HDBSCAN* carry the highest memory consumption because, besides the input, these implementations produce another four double-precision float matrices living simultaneously on memory. The estimated RAM consumption of these five objects (Equation 4.2) is reported in Table 4.5 for those datasets that produced a memory crash.

$$V_{\mathrm{RAM}} = \frac{(M * m_1 + 4m_2) * N}{2^{30}} \tag{4.2}$$

In Equation 4.2, m_1 is the size of the float data type employed in the input matrix ($m_1 = 4$), m_2 is the size of the float data type employed internally by HDBSCAN*

($m_2 = 8$), N denotes the number of conformations in the dataset, and M represents the number of columns in the input matrix ($M = N$ for the generic-RMSD, while $M = 3 * number\_of\_atoms$ in the generic-Euclidean variant).

4.4.3 . Equivalence between MDSCAN and HDBSCAN* alternatives

As stated before, the HDBSCAN clustering alternatives analyzed in this work are expected to produce distinct partitions for each dataset, mainly because the algorithm for the MST construction drastically varies among them. However, it is worth assessing the impact of this methodological divergence on the outcome clusters produced by each variant.

The upper triangle of each matrix in Table 4.6 shows the global ARI between the MDSCAN (A), the generic RMSD-based HDBSCAN* (B), the generic Euclidean-based HDBSCAN* (C), and the Prim Euclidean-based HDBSCAN* (D) implementations. As envisioned, the global similarity of clusterings is far from 1.00 in every case. However, there is an appreciable resemblance in clusterings produced by generic implementations of HDBSCAN* using RMSD or Euclidean metric in the 6 and 30 kF datasets. In the particular case of the 30 kF dataset, all clusterings coming from HDBSCAN* are correlated but not analogous to MDSCAN outcomes. For datasets bigger than 30 kF, only MDSCAN and Euclidean-based HDBSCAN* produced results, and they were also divergent.

Table 4.6: *Adjusted Rand Index (ARI) of clustering outputs obtained with different HDBSCAN implementations for each analyzed dataset. The upper triangle of each matrix corresponds to the global ARI, while in the lower triangle, only the ARI of those clusters whose population is higher than 1% of the dataset size is depicted.*[1]

	6 kF				30 kF				50 kF			
	A	*B*	*C*	*D*	*A*	*B*	*C*	*D*	*A*	*B*	*C*	*D*
A	1.00	0.34	0.35	0.34	1.00	0.24	0.44	0.48	1.00	—	—	0.00
B	0.74	1.00	0.66	0.32	0.34	1.00	0.61	0.64	—	1.00	—	—
C	0.74	1.00	1.00	0.38	0.46	0.49	1.00	0.76	—	—	1.00	—
D	0.74	0.99	0.99	1.00	0.50	0.55	0.76	1.00	0.00	—	—	1.00
	100A kF				250 kF				500 kF			
	A	*B*	*C*	*D*	*A*	*B*	*C*	*D*	*A*	*B*	*C*	*D*
A	1.00	—	—	-0.02	1.00	—	—	-3.66	1.00	—	—	-27.61
B	—	1.00	—	—	—	1.00	—	—	—	1.00	—	—
C	—	—	1.00	—	—	—	1.00	—	—	—	1.00	—
D	0.80	—	—	1.00	-4.8	—	—	1.00	1.00	—	—	1.00

[1] Each different HDBSCAN variant is represented by a letter: A-) MDSCAN, B-) the generic RMSD-based HDBSCAN*, C-) the generic Euclidean-based HDBSCAN*, and D-) the Prim Euclidean-based HDBSCAN*.

When clustering MD simulations, users are often more interested in the representative, more significant clusters found in datasets. The lower triangle of each matrix in Table 4.6 shows the pairwise ARI of previous clusterings when considering only clusters whose population is higher than 1% of the corresponding dataset size. The ARI between MDSCAN and the other alternatives doubles 6 kF. For this dataset, the most populated

clusters reported by all software are similar (clusterings from all variants of HDBSCAN* are equivalent). This fortuitous agreement is dataset-dependent, as appreciated in the 30 kF case for which no such prominent improvement of the clustering analogy is attained.

While the ARI can globally inform on partitions similarity, it yields no clues on the equivalence of individual clusters fetched by every software. In Annex 9.2.7, we present the quantitative equivalence between representative clusters (those whose population is higher than 1% of their corresponding trajectory size) detected in trajectories 6 (Table 9.8) and 30 kF (Table 9.9) with *MDSCAN* (A) and the HDBSCAN* variants; generic RMSD-based (B), generic Euclidean-based (C), and Prim RMSD-based (D).

As a general trend, clusters from all implementations are interconnected. MDSCAN often produces smaller groups of frames with the advantage of them having a higher collective similarity (shorter average diameter). The smaller size of some clusters produced by MDSCAN (together with the fact that they are tighter than those seized with Euclidean-based options of HDBSCAN*) is a direct consequence of the pairwise superposition followed when using the RMSD metric. Having tight clusters is the desired behavior when clustering MD as more related frames get included in the same group, facilitating the visual analysis or quantitative averages calculated from clustered structures.

4.5 . Spatial complexity of proposed algorithms

Our motivation for enhancing widely used or promising clustering algorithms in the molecular simulation domain is primarily driven by the bottleneck created by their quadratic spatial complexity. Accordingly, this section provides a succinct overview of our proposed methods' performance in this context. It is crucial to underscore that while reducing the time complexity of these procedures or their equivalents is undeniably significant, this was not the objective pursued in the current study. Consequently, an in-depth assessment, despite our implementations' often-accelerated performance, lies beyond this manuscript's scope.

QTPy (refer to Section 4.1.1) was only suggested as a proof of concept to highlight the inaccuracies of other clustering alternatives that flawlessly claim to perform QT. It does not constitute a proper optimization, and its spatial complexity is $O(n^2)$. Nonetheless, we employed half-precision float values to represent RMSD, thereby enabling the QTPy similarity matrix to consume half the space required by alternatives that utilize similarity matrices of single-precision floats, such as the gromos option of GROMACS.

BitQT (see Section 4.1) and **BitClust** (refer to Section 4.2) adopt the same approach to shorten the memory requirements of the Daura and QT algorithms by encoding RMSD pairwise distances as bits. Therefore, despite their quadratic spatial complexity, they can process considerably larger trajectories than existing implementations. Given the inherent sparsity across its columns, compressing the binary matrix could feasibly yield sub-quadratic performance for most real-world scenarios. However, at the time of implementation, the *bitarray* library used for binary encoding did not possess this capability.

The **DP+** algorithm represents an attempt to alleviate the quadratic spatial com-

plexity intrinsic to Density Peaks (DP) approaches. Instead of constructing a complete matrix, our method operates on transient vectors of size N (where N is the number of frames) and two heaps that may expand memory usage. The secondary heap comprises tuples (i, ρ), representing the frame's index and density value. In the worst-case scenario, this heap expands to N entries with a spatial complexity of $N \cdot (O(1) + O(1)) \equiv O(N)$, thereby demonstrating linear scaling with trajectory length. The primary heap includes the above-mentioned elements and a smaller subset η_i of size $0.02 \cdot N$. If we treat this as a constant factor c, even if the main heap reaches its maximum size of N tuples, the complexity remains $N \cdot (O(1) + O(1) + O(c)) \equiv O(cN)$, still linear. It is worth noting that such extreme scenarios are unlikely with prudent parameter selection.

MDSCAN's primary objective was to mitigate the quadratic complexity of available HDBSCAN* alternatives when utilizing high-dimensional metrics like the RMSD. Strikingly similar to DP+, MDSCAN employs the same heap data structures and transient vectors already detailed. The sole distinction is in the type of information these heaps contain; hence, the worst-case spatial complexity analysis presented for DP applies to MDSCAN. However, MDSCAN processes the trajectory file differently from all other software proposed here, as a vantage point tree is constructed, necessitating a copy of the trajectory. This processing does not increase the algorithm's spatial complexity, although it requires more space compared to DP.

5 - NUCLEAR: AN EFFICIENT ASSEMBLER FOR THE FBDD OF CMOs

As we stated in this document's introductory Chapter , our main goal is the *in silico* fragment-based design of oligonucleotides showing selective affinity for their intended target. After we assessed the reliable docking and screening powers of the MCSS scoring function in Chapter 3, the next natural stage of the design is to link the most promising fragments into oligonucleotide chains.

Although several tools for linking fragments are available in the literature, none is suitable to be included in our approach mainly due to the following reasons: (i) they cannot work with the file formats coming from MCSS and CHARMM software (PSF, DCD), (ii) they are not designed to link oligonucleotides from C5' to O3' guaranteeing clash-free solutions, and most important (iii) they are unable to process high volumes of data.

In this Chapter, we present a novel software called NUCLEotide AssembleR (NUCLEAR), addressing the above limitations. NUCLEAR can perform different kinds of oligonucleotide searches and retrieve hotspots in the receptor from the distributions of docked fragments.

Section 5.1 succinctly presents the workflows available in NUCLEAR. Next, Section 5.2 details the protocol for searching hotspots at the receptor's surface. In Section 5.3 the sequence- and spatial-constrained searches of oligonucleotides are described. Finally, Section 5.4 is devoted to the reproduction of four crystal structures using NUCLEAR, to discuss the computational cost of the algorithm's main steps, and to delve into its limitations.

5.1 . NUCLEAR overview

In a NUCLEAR single job, users can request one of two exclusive explorations (as depicted in Figure 5.1): either a molecular hotspots search (to gain insights into the most accessible regions of the receptor) or an oligonucleotide search. In the latter case, the oligo-nucleotide sequence and the receptor region to consider can also be specified.

There are two common steps for hotspots and oligonucleotide search; (i) the parsing of the main configuration file and (ii) the processing of MCSS docking distributions. In the former, the specification of all parameters (visit `https://rglez.github.io/nuclear_docs/` for the complete list) occurs through a user-customized configuration file. NUCLEAR parses this file and checks that all keys have reasonable values specified (*e.g.*, the minimum length of requested sequences must be an integer greater than 1). Some essential options to specify in this file are the size range of sequences to search, the number of best-scored solutions to output, and the possibility to cluster the input docking distributions (in terms of RMSD similarity). Besides, the inter-related parameters are

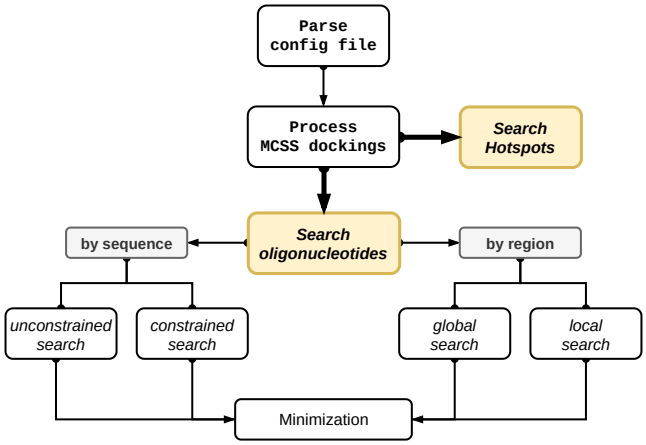


Figure 5.1: *Diagram of NUCLEAR workflows. Users may request one of two exclusive explorations in a single job:* ***(i)*** *a molecular hotspots search or* ***(ii)*** *an oligonucleotide search (in which case the oligonucleotide sequence and the receptor region to consider can also be specified).*

checked for consistency (*e.g.*, if clustering is not requested, no distance cutoff should be specified).

On the other hand, (ii) consists mainly of parsing and clustering MCSS docking distributions, which may involve many poses exhibiting some geometrical redundancy (see Section 3.2). NUCLEAR offers two ways for reducing the number of input poses to process: by filtering them according to their number of contacts with the protein (poses with fewer contacts than a user-specified cutoff are discarded) and by performing a BitClust-inspired clustering step (see Section 2.1.4). Both alternatives are optional and can be used together. Note that these reductions occur independently for each distribution (every single CRD file out of the several specified is independently reduced). The similarity metric employed between poses is their pairwise RMSD (discarding hydrogen atoms).

Once the MCSS distributions have been parsed and eventually reduced through the number of contacts filter or by the clustering procedure, all remaining poses (coming from each considered CRD distribution) are concatenated. Hence, they yield the NUCLEAR search space, which can be used to search hotspots or oligonucleotide sequences.

5.2 . Search of hotspots

Detecting protein hotspots is a usual step of FBDD workflows. Though several definitions exist, a hotspot is associated with either of two complementary concepts; (i) a residue or cluster of residues contributing majorly to the binding free energy, or (ii) a site on a target protein having a high propensity for ligand binding[210]. NUCLEAR can detect hotspots following the latter interpretation, aiding to discriminate the most favorable sites where considered fragments could bind and subsequently be joined to make an oligonucleotide.

Figure 5.2 shows the main steps to perform the NUCLEAR's hotspots search. Each

MCSS exploration comes in a CRD file from which every replica (coordinates, names, and scoring) is extracted and their Cartesian coordinates saved as a kd-tree data structure. The Cartesian coordinates of the protein are also kept as a kd-tree. The fingerprints are then computed for each replica in each MCSS exploration provided. In this context, a *fingerprint* is the contact between a particular replica and the protein. NUCLEAR considers as a *contact*, the protein's first atom under a cutoff distance from a ligand atom. Note that fingerprints can be expressed at least with two resolutions: high-res (or atomic-res): where the residue and the particular atom involved in the contact are indicated, and low-res (or residue-res): where only the residue involved in the interaction is indicated.

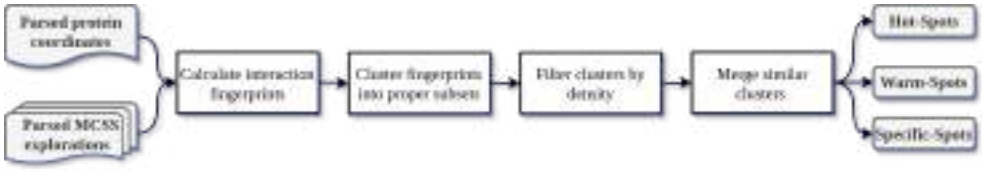


Figure 5.2: *Workflow for the hotspots identification in a receptor protein using* NUCLEAR.

NUCLEAR uses low-res fingerprints to locate and cluster the interacting regions of the protein (see Figure 5.3A-B). The rationale behind this clustering approach is that more extensive contact zones can represent many smaller ones (Figure 5.3B-C), and the most populated clusters will inform on suitable binding sites.

The clustering of fingerprints occurs as follows: (i) all fingerprints are sorted by increasing order of their size (from biggest to smallest) (Figure 5.3E), (ii) the most extensive fingerprint is taken as the seed of a cluster in which all other fingerprints that are proper subsets of the seed will be grouped (Figure 5.3F), (iii) clustered fingerprints are removed from consideration, and (iv) the process restarts from (ii) until no more fingerprints are available for clustering.

Once we have obtained the subset clusters, they can be filtered by size. However, the size of the clusters is not normalized by the number of residues it contains. To leverage this situation, NUCLEAR employs a cluster's normalized ([0, 1]) density as a population criterion to select the most representative. The density of a particular cluster is calculated as the number of replicas it contains divided by the count of unique residues. These numbers are normalized, and users can effectively discern the most representative clusters in size through a density cutoff parameter.

Each subset cluster already described has an associated seed that "represents" all other smaller fingerprints in the same cluster. As these seeds correspond to a protein region, they may overlap, giving rise to the potential need to merge similar areas. This merging of clusters happens in NUCLEAR right after the density filter step and will use the Tanimoto Index (TI) between seeds to decide if two clusters should be joined. The process always merges worst-ranked clusters with the best-ranked ones if their TI is below the user-specified threshold.

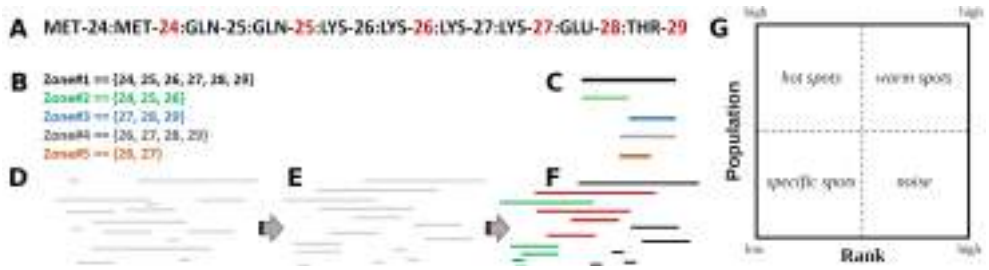


Figure 5.3: *Hotspots search using NUCLEAR's low-res fingerprints.* ***A:*** *An example of low-res contacts between a replica and a protein. Unique residues are colored in red, and a colon separates them).* ***B:*** *Five low-res fingerprints (represented as sets of residue numbers).* ***C:*** *Overlap of the fingerprints in B.* ***D:*** *An schematic representation of several fingerprints before clustering.* ***E:*** *Fingerprints are sorted by increasing order of their size.* ***F:*** *Clusters of fingerprints. Each color denotes a cluster whose seed is the biggest fingerprint not previously clustered and whose members are all fingerprints that are proper subsets of the seed.* ***G:*** *Classification of clusters (spots) depending on their best-ranked replica score and their population*

NUCLEAR reports as spots, every cluster generated from the procedure described above. Depending on the best-ranked replica score contained in a cluster and each cluster's population, a qualitative classification of spots can be made into four categories (see Figure 5.3G): (i) hotspots (clusters with high population and with a low-ranked replica), (ii) warm spots (clusters with high-ranked replicas but highly populated), (iii) specific spots (clusters with a low-ranked replica and low population), and (iv) noise (small clusters with high-ranked replicas).

5.3 . Search of oligonucleotides

When the attention goes to searching oligonucleotide sequences, NUCLEAR's primary goal is simple: to find all geometrically suitable arrangements by joining *reachable* poses. Two poses i, j are considered *reachable* if the distance between the C5' atom of i and the O3' atom of j is under a distance cutoff and other atoms do not produce steric clashes. NUCLEAR stores this information in a non-symmetric binary matrix R, the reachability matrix (see Figure 5.4A), whose length equals the number of poses in the search space. In R, each position $R_{ij} = 1$ if atoms C5' of i and O3' of j are under a cutoff distance and no pair of ij heavy atoms produce clashes ($R_{ij} = 0$ otherwise). A directed graph G, the reachability graph, can be pictured from matrix R (See Figure 5.4D).

The sequence search followed by NUCLEAR comprises retrieving all simple paths found in G (iteratively starting with each node as the source of the paths) following a Depth-First Search (DFS) strategy. However, this DFS is constrained because, in the expansion of a particular source, produced paths must not contain nodes having steric clashes, in which case the entire branch under expansion gets pruned. While reachable nodes in G will not produce clashes, there is no way to assert this for two non-adjacent

A Reachability Matrix (R)

from \ to	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	0	0	0	0
2	0	0	1	0	0	0	1	0	0
3	0	0	0	1	1	1	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	0	0
8	0	1	0	0	0	0	0	0	0
9	0	1	1	0	0	0	0	0	0

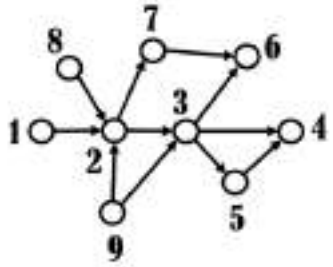
B Clash Matrix (C)

	1	2	3	4	5	6	7	8	9
1	0	0	0	1	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	1	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	1	0	0	0

C Score of nodes

1	2	3	4	5	6	7	8	9
0.62	0.22	0.92	0.98	0.84	0.88	0.50	0.35	0.10

D



Reachability Graph

E Sequence search for node 9

Current sequence	Score	Clashes
$9_{2,3}$	0.10	6
9_3-$2_{3,7}$	0.16	6,4
9_3-2_7-3_5	0.36	6,4
9_3-2_7-3-5	0.57	6,4
9_3-2-7	0.33	6,4
9-$3_{4,5}$	0.51	6
9-3_5-4	0.67	6,1,5
9-3-5	0,62	6

Figure 5.4: *Graph view of the* NUCLEAR *sequence search procedure.* ***A:*** *Reachability matrix.* ***B:*** *Clash matrix.* ***C:*** *Score of nodes.* ***D:*** *Reachability graph.* ***E:*** *Example of sequence search starting from node 9.*

ones. That is why NUCLEAR stores another symmetrical binary matrix C, the clash matrix (see 5.4B) of the same length that R. In C, each position $C_{ij} = 1$ if there is at least one clash between heavy atoms of poses i and j ($C_{ij} = 0$ otherwise).

The example in Figure 5.4E starts from node 9, which can be expanded to node 2 or 3. NUCLEAR always starts expansions with the lowest index, so sequence 9-2 is created in a second step. Following this behavior, the sequence 9-2-3 is created. NUCLEAR can expand node 3 to nodes 4, 5, or 6. However, adding nodes 4 or 6 would produce clashes in the growing sequence (with 2 and 9, respectively), so the only possibility is that 9-2-3-5 gets created. As the only expansion of node 5 (4) would produce clashes, the branch expansion stops, and NUCLEAR backtracks to the next feasible branch, in this case, 9-2-7. As neither node 2 nor 7 has expansion choices, NUCLEAR backtracks again to node 9, producing 9-3, 9-3-4, and 9-3-5.

Although in Figure 5.4E, we have illustrated the sequence search process in terms of intuitive operations, NUCLEAR implements the most important steps as binary operations, profiting from the binary encoding displayed by R and C matrices. For instance, clashes are recorded by an OR operation of the bit vectors in C that corresponds to the current sequence's nodes. Similarly, the constrained DFS is conducted by pushing to and popping from a stack of bit-vectors calculated through a consecutive XOR/AND operation that removes the clashes from the currently expanded sequence.

To have an exhaustive exploration, the described DFS must be repeated, starting from all nodes in G that are not leaves and saving all paths with two or more nodes. However, the number of retrieved paths quickly explodes even for small search spaces. Considering that only the subset of better-ranked sequences is of interest at the end of the search, NUCLEAR proposes a pruning strategy that considerably reduces the computational cost of the described approach.

Concretely, a fixed-size heap data structure is used to store generated sequences. Using a heap, it is possible to quickly retrieve the worst-scored sequence generated so far and use that scoring value to stop the expansion of paths leading to a worse score, as these paths will never be selected in the output top-N. Another advantage of this approach is that not all sequences get written to the disk (a considerable bottleneck when millions of solutions are available), but only the top-N better-scored ones.

NUCLEAR can localize the search for oligonucleotides in a particular receptor region if a valid atomic selection is provided in the configuration file. This local search causes only fingerprints intersecting the selected residues get considered (see local search in Figure 5.1). Equally appealing is the NUCLEAR ability to specify oligonucleotide sequence constraints to the search (see constrained search in Figure 5.1). Both options significantly help to reduce the computational cost of the DFS procedure in knowledge-guided explorations.

Once the top-N best-scored sequences have been written to disk as PDB files, NUCLEAR generates a CHARMM minimization script for each one. In the declared minimization process, both the receptor and ligands can move. Slight deviations in the pre and post-minimization oligonucleotide may appear because of the phosphodiester links

reorganization.

5.4 . Case studies

5.4.1 . Evaluated parameters

As stated, NUCLEAR's primary goal is the fragment-based design of oligonucleotides exhibiting an affinity for a particular therapeutic target (usually a protein). In line with this general objective, only fragments in contact with the receptor are considered in the different available searches. Although this peculiarity does not entirely prevent the tool from reproducing experimental ssRNA-protein complexes, only cases where contacts between crystallized ligands and the receptor exist would be suitable for reproduction. The success of these *in-silico* reproductions is then greatly conditioned by the ability of the docking software (MCSS in our case) to capture native-like conformations of the fragments involved, as discussed in Section 3.3.

With the previous limitation in mind, the NUCLEAR ability to detect experimental binding modes of oligonucleotides crystallized within a protein was evaluated through four high-resolution complexes involving Ribonucleic Acid Binding Proteins (RBPs) belonging to the three families most represented in humans[268]: (i) the RNA recognition motif (RRM) of the protein Nab3 bound to its UCUU recognition sequence (PDB ID: 2XNR)[269], (ii) a group of three CCCH zinc finger (Zn-Fs) domains of the Unkempt protein linked to a UUAUU chain (PDB ID: 5ELH)[270], (iii) the KH2 domain of the MEX-3C protein linked to a CAGAGCU chain (PDB ID: 5WWX)[271], and (iv) the poly(A)-binding protein in complex with an AAAAAAAA chain (PDB ID: 1CVJ)[272].

For the three first cases, only three successive nucleotides in the RNA sequence are directly involved in specific recognition, so only these triplets were kept to define the RNA chains used as reference (UCU for 2XNR, UUA for 5ELH, and AGA for 5WWX). Note that the discarded nucleotides either show ambiguous electron density or make little or no contact with the domain of the asymmetric unit.

When designing inhibitors against a receptor protein, it is common practice to restrict the search space of docking simulations to regions encompassing the known binding site. We denoted as *"global searches"* those performed using all replicas present in this relatively ambiguous (17 $Å^3$) and not restricted to specific residues region of the protein. A more constrained but still valid approach would be to limit the oligonucleotide search space (not the docking region) to replicas involved in interactions with (or close to) the protein residues interacting with ligands in experimental structures. These latter kinds of jobs are referred to as *"local searches"*.

As detailed in Section 5.3, NUCLEAR uses a fixed-size heap to keep only the most energetically favored oligonucleotides out of the potentially millions of generated solutions. This strategy guarantees that only a relevant subset of chains is retained to save on disk and minimize subsequently. However, there is no robust way to estimate the energy that retrieved oligonucleotides would have after minimization. NUCLEAR uses the pre-scoring to decide which chains will be included in the heap. This pre-scoring

can be calculated in several ways, but we evaluated it as the arithmetic or the geometric mean of the MCSS score of replicas composing the oligonucleotide chains. In theory, the geometric mean would account for an inherent weighting of significantly different scored replicas appearing in the same oligonucleotide.

One of the most neuralgic points of NUCLEAR is the combinatorial explosion that quickly occurs as the number of replicas to analyze increases. So a pre-clustering step (described in Section 2.1.4) can be set to reduce the number of MCSS poses for each fragment (as exemplified in Section 5.1). We evaluated the effect of no clustering and using 1 or 2 Å as RMSD cutoff for clustering MCSS docking distributions previous to NUCLEAR searches.

The last parameter we varied in the present case studies was the maximum distance from C5' to O3' of contiguous nucleotides to be considered linkable. This parameter strongly impacts the number of chains NUCLEAR can mark as valid. The larger the distance, the less restrictive the search becomes, as more continuous nucleotides would be considered linkable. The distance to consider a clash between ligand and protein atoms was fixed to 1.5 Å .

From now on, we will use the **XYij** notation when referring to NUCLEAR explorations conducted with the parameters detailed so far. Under this nomenclature, **X** refers to the exploration's extension and can be either G or L for a global or a local search, respectively. **Y** denotes the kind of pre-scoring employed by NUCLEAR and can be A (arithmetic) or G (geometric). The RMSD clustering cutoff **i** can take values of 0 (no clustering), 1, and 2 Å, while **j** being the d(C5'-O3') takes values of 3, 4, 5, or 6 Å. We will denote as *equivalent searches* those with the same **ij** and a pertinent combination of **XY**; GAij/GGij and LAij/LGij (where the effect of arithmetic vs. geometric pre-scoring can be assessed) or GAij/LAij and GGij/LGij (where the effect of global vs. local search space can be assessed). Note that each **XYij** exploration produces a *distribution* of linkable oligo-nucleotide candidates from which only the top 5000 best-scored (heap size) were minimized.

5.4.2 . Trends in reproducing experimental binding modes

In each individual protein, the initial number of mono-nucleotides ($\#mono$ in Tables 5.1-5.3) is identical for equivalent searches because this magnitude gets computed after clustering but before selecting the region to consider. Evidently, $\#mono$ suffers a considerable reduction if a clustering step is conducted. The higher the RMSD cutoff, the more considerable this reduction becomes (up to 70, 67, and 73% for 2XNR, 5WWX, and 5ELH respectively when passing from no clustering to a 2 Å cutoff clustering).

As an expected trend, for a particular combination of **XY** ($Method$ in Tables 5.1-5.3), increasing the maximum allowed linkable distance from C5' to O3' ($d(C5' - O3')$ in Tables 5.1-5.3), the number of oligo-nucleotides that NUCLEAR outputs also increases in two or three orders of magnitude in the explored range of 3-6 Å, reaching more than 34, 98, and 25 million of solutions for 2XNR, 5WWX, and 5ELH respectively in the searches where d(C5'-O3') was set to 6 Å and no clustering was performed.

The effect of conducting a local vs. global search in the number of oligo-nucleotides

Table 5.1: NUCLEAR *explorations to retrieve native-like structures for the 2XNR protein.*

Method	RMSD [Å]	d(C5'-O3') [Å]	#mono	#oligo	min E [kcal/mol]	#native	native-like most similar					native-like best ranked				
							E [kcal/mol]	RMSD [Å]	pre-Rank	post-Rank	dRMSD [Å]	E [kcal/mol]	RMSD [Å]	pre-Rank	post-Rank	dRMSD [Å]
GA	0	3	9075	15694	-45.90	-	-	-	-	-	-	-	-	-	-	-
		4		677268	-50.33	-	-	-	-	-	-	-	-	-	-	-
		5		6492116	-57.23	235	-47.35	1.05	578	644	1.86	-51.47	1.40	1335	95	1.68
		6		34213514	-59.90	541	-48.51	0.93	4071	571	1.84	-53.06	1.43	778	53	1.40
	1	3	5552	3849	-44.95	-	-	-	-	-	-	-	-	-	-	-
		4		164885	-50.38	-	-	-	-	-	-	-	-	-	-	-
		5		1595613	-54.16	58	-48.99	1.00	973	46	2.58	-50.55	1.75	5	24	1.98
		6		8447693	-54.16	168	-48.05	1.06	4419	234	1.80	-51.68	1.61	1388	23	1.84
	2	3	2696	338	-44.87	-	-	-	-	-	-	-	-	-	-	-
		4		17638	-46.76	-	-	-	-	-	-	-	-	-	-	-
		5		176381	-54.16	4	-42.08	1.73	97	145	1.93	-45.28	1.98	72	41	1.40
		6		953535	-54.16	11	-48.38	1.18	4903	21	2.02	-48.89	1.35	1122	15	1.50
GG	0	3	9075	15694	-45.90	-	-	-	-	-	-	-	-	-	-	-
		4		677268	-50.33	-	-	-	-	-	-	-	-	-	-	-
		5		6492116	-57.23	255	-48.99	1.00	3193	304	2.58	-51.47	1.40	699	96	1.68
		6		34213514	-57.28	495	-48.51	0.93	4237	599	1.84	-53.06	1.43	872	54	1.40
	1	3	5552	3849	-44.95	-	-	-	-	-	-	-	-	-	-	-
		4		164885	-50.38	-	-	-	-	-	-	-	-	-	-	-
		5		1595613	-54.16	53	-48.99	1.00	486	43	2.58	-50.55	1.75	8	22	1.98
		6		8447693	-54.16	166	-48.99	1.00	3643	129	2.58	-51.68	1.61	1201	23	1.84
	2	3	2696	338	-45.90	-	-	-	-	-	-	-	-	-	-	-
		4		17638	-46.76	-	-	-	-	-	-	-	-	-	-	-
		5		176381	-54.16	4	-42.08	1.73	54	138	1.93	-45.28	1.98	40	40	1.40
		6		953535	-54.16	12	-48.38	1.18	2863	22	2.02	-48.89	1.35	832	15	1.50
LA	0	3	9075	15694	-45.90	-	-	-	-	-	-	-	-	-	-	-
		4		677268	-50.33	-	-	-	-	-	-	-	-	-	-	-
		5		6492116	-57.23	235	-47.35	1.05	578	644	1.86	-51.47	1.40	1335	95	1.68
		6		34213514	-59.90	541	-48.51	0.93	4071	571	1.84	-53.06	1.43	778	53	1.40
	1	3	5552	3849	-44.95	-	-	-	-	-	-	-	-	-	-	-
		4		164885	-50.38	-	-	-	-	-	-	-	-	-	-	-
		5		1595613	-54.16	58	-48.99	1.00	973	46	2.58	-50.55	1.75	5	24	1.98
		6		8447693	-59.90	168	-48.51	0.93	4071	571	1.84	-53.06	1.43	778	53	1.40
	2	3	2696	338	-44.87	-	-	-	-	-	-	-	-	-	-	-
		4		17638	-46.76	-	-	-	-	-	-	-	-	-	-	-
		5		176381	-54.16	4	-42.08	1.73	97	145	1.93	-45.28	1.98	72	41	1.40
		6		953535	-54.16	11	-48.38	1.18	4903	21	2.02	-48.89	1.35	1122	15	1.50
LG	0	3	9075	15694	-49.90	-	-	-	-	-	-	-	-	-	-	-
		4		677268	-50.33	-	-	-	-	-	-	-	-	-	-	-
		5		6492116	-57.23	255	-48.99	1.00	3193	304	2.58	-51.47	1.40	699	96	1.68
		6		34213514	-57.28	495	-48.51	0.93	4237	599	1.84	-53.06	1.43	872	54	1.40
	1	3	5552	3849	-44.95	-	-	-	-	-	-	-	-	-	-	-
		4		164885	-50.38	-	-	-	-	-	-	-	-	-	-	-
		5		1595613	-54.16	53	-48.99	1.00	486	43	2.58	-50.55	1.75	8	22	1.98
		6		8447693	-54.16	166	-48.99	1.00	3643	129	2.58	-51.68	1.61	1201	23	1.84
	2	3	2696	338	-44.87	-	-	-	-	-	-	-	-	-	-	-
		4		17638	-46.76	-	-	-	-	-	-	-	-	-	-	-
		5		176381	-54.16	4	-42.08	1.73	54	138	1.93	-45.28	1.98	40	40	1.40
		6		953535	-54.16	12	-48.38	1.18	2863	22	2.02	-48.89	1.35	832	15	1.50

Table 5.2: NUCLEAR *explorations to retrieve native-like structures for the 5WWX protein.*

Method	RMSD [Å]	d(C5'-O3') [Å]	#mono	#oligo	min E [kcal/mol]	#native	native-like most similar					native-like best ranked				
							E [kcal/mol]	RMSD [Å]	pre-Rank	post-Rank	dRMSD [Å]	E [kcal/mol]	RMSD [Å]	pre-Rank	post-Rank	dRMSD [Å]
BA	0	3	22818	37619	-68.95	-	-	-	-	-	-	-	-	-	-	-
		4		1658706	-68.95	-	-	-	-	-	-	-	-	-	-	-
		5		17435561	-70.85	-	-	-	-	-	-	-	-	-	-	-
		6		98355759	-70.90	4	-61.81	1.06	994	827	1.23	-64.95	1.89	2827	256	2.17
	1	3	14664	9001	-67.86	-	-	-	-	-	-	-	-	-	-	-
		4		412382	-68.40	-	-	-	-	-	-	-	-	-	-	-
		5		4337966	-70.14	7	-60.82	1.06	3370	611	2.22	-64.22	1.35	3635	122	1.91
		6		24619477	-69.65	2	-61.73	1.96	1920	481	1.40	-64.19	1.97	2437	161	2.50
	2	3	7494	1130	-58.84	-	-	-	-	-	-	-	-	-	-	-
		4		50568	-64.05	-	-	-	-	-	-	-	-	-	-	-
		5		537549	-68.63	-	-	-	-	-	-	-	-	-	-	-
		6		3068691	-69.65	1	-61.73	1.96	675	149	1.40	-61.73	1.96	675	149	1.40
BG	0	3	22818	37619	-68.95	-	-	-	-	-	-	-	-	-	-	-
		4		1658706	-68.95	-	-	-	-	-	-	-	-	-	-	-
		5		17435561	-70.85	-	-	-	-	-	-	-	-	-	-	-
		6		98355759	-70.90	7	-61.81	1.06	923	1004	1.23	-64.95	1.89	2728	296	2.17
	1	3	14664	9001	-67.86	-	-	-	-	-	-	-	-	-	-	-
		4		412382	-68.40	-	-	-	-	-	-	-	-	-	-	-
		5		4337966	-70.14	7	-60.82	1.06	3075	620	2.22	-64.22	1.35	3222	122	1.91
		6		24619477	-73.53	2	-61.73	1.96	2184	534	1.40	-64.19	1.97	1711	182	2.50
	2	3	7494	1130	-58.84	-	-	-	-	-	-	-	-	-	-	-
		4		50568	-64.05	-	-	-	-	-	-	-	-	-	-	-
		5		537549	-68.63	-	-	-	-	-	-	-	-	-	-	-
		6		3068691	-69.65	1	-61.73	1.96	737	152	1.40	-61.73	1.96	737	152	1.40
GA	0	3	22818	35521	-68.95	-	-	-	-	-	-	-	-	-	-	-
		4		1564586	-68.95	-	-	-	-	-	-	-	-	-	-	-
		5		16313232	-70.85	-	-	-	-	-	-	-	-	-	-	-
		6		91164053	-70.90	4	-61.81	1.06	989	831	1.23	-64.95	1.89	2810	257	2.17
	1	3	14664	8267	-67.86	-	-	-	-	-	-	-	-	-	-	-
		4		382221	-68.40	-	-	-	-	-	-	-	-	-	-	-
		5		3999686	-70.14	7	-60.82	1.06	3310	616	2.22	-64.22	1.35	3572	122	1.91
		6		22544353	-69.65	2	-61.73	1.96	1893	486	1.40	-64.19	1.97	2407	162	2.50
	2	3	7494	1044	-58.84	-	-	-	-	-	-	-	-	-	-	-
		4		46327	-64.05	-	-	-	-	-	-	-	-	-	-	-
		5		491288	-68.63	-	-	-	-	-	-	-	-	-	-	-
		6		2783304	-69.65	1	-61.73	1.96	659	149	1.40	-61.73	1.96	659	149	1.40
GG	0	3	22818	35521	-68.95	-	-	-	-	-	-	-	-	-	-	-
		4		1564586	-68.95	-	-	-	-	-	-	-	-	-	-	-
		5		16313232	-70.85	-	-	-	-	-	-	-	-	-	-	-
		6		91164053	-70.90	7	-61.81	1.06	922	1005	1.23	-64.95	1.89	2721	296	2.17
	1	3	14664	8267	-67.86	-	-	-	-	-	-	-	-	-	-	-
		4		382221	-68.40	-	-	-	-	-	-	-	-	-	-	-
		5		3999686	-70.14	7	-60.82	1.06	3044	624	2.22	-64.22	1.35	3191	122	1.91
		6		22544353	-73.53	2	-61.73	1.96	2167	538	1.40	-64.19	1.97	1697	185	2.50
	2	3	7494	1044	-58.84	-	-	-	-	-	-	-	-	-	-	-
		4		46327	-64.05	-	-	-	-	-	-	-	-	-	-	-
		5		491288	-68.63	-	-	-	-	-	-	-	-	-	-	-
		6		2783304	-69.65	1	-61.73	1.96	726	153	1.40	-61.73	1.96	726	153	1.40

Table 5.3: NUCLEAR *explorations to retrieve native-like structures for the 5ELH protein.*

Method	RMSD [Å]	d(C5'-O3') [Å]	#mono	#oligo	min E [kcal/mol]	#native	native-like most similar					native-like best ranked				
							E [kcal/mol]	RMSD [Å]	pre-Rank	post-Rank	dRMSD [Å]	E [kcal/mol]	RMSD [Å]	pre-Rank	post-Rank	dRMSD [Å]
GA	0	3	8473	14965	-60.40	-	-	-	-	-	-	-	-	-	-	-
		4		562254	-71.13	-	-	-	-	-	-	-	-	-	-	-
		5		5008076	-71.75	-	-	-	-	-	-	-	-	-	-	-
		6		25863579	-70.04	7	-55.85	1.46	877	490	1.30	-61.64	1.71	138	63	1.29
	1	3	4881	3037	-58.29	-	-	-	-	-	-	-	-	-	-	-
		4		114771	-69.63	-	-	-	-	-	-	-	-	-	-	-
		5		1056292	-72.88	-	-	-	-	-	-	-	-	-	-	-
		6		5508112	-74.19	1	-51.42	1.82	2107	1133	1.37	-51.42	1.82	2107	1133	1.37
	2	3	2262	245	-53.98	-	-	-	-	-	-	-	-	-	-	-
		4		11335	-59.20	-	-	-	-	-	-	-	-	-	-	-
		5		110845	-72.88	-	-	-	-	-	-	-	-	-	-	-
		6		596755	-74.19	-	-	-	-	-	-	-	-	-	-	-
GG	0	3	8473	14965	-60.40	-	-	-	-	-	-	-	-	-	-	-
		4		562254	-71.76	-	-	-	-	-	-	-	-	-	-	-
		5		5008076	-71.75	-	-	-	-	-	-	-	-	-	-	-
		6		25863579	-70.04	7	-55.85	1.46	3930	522	1.30	-61.64	1.71	1026	76	1.29
	1	3	4881	3037	-58.29	-	-	-	-	-	-	-	-	-	-	-
		4		114771	-69.63	-	-	-	-	-	-	-	-	-	-	-
		5		1056292	-72.88	-	-	-	-	-	-	-	-	-	-	-
		6		5508112	-74.19	1	-51.42	1.82	3630	1223	1.37	-51.42	1.82	3630	1223	1.37
	2	3	2262	245	-53.98	-	-	-	-	-	-	-	-	-	-	-
		4		11335	-59.20	-	-	-	-	-	-	-	-	-	-	-
		5		110845	-72.88	-	-	-	-	-	-	-	-	-	-	-
		6		596755	-74.19	-	-	-	-	-	-	-	-	-	-	-
LA	0	3	8473	14710	-60.4	-	-	-	-	-	-	-	-	-	-	-
		4		540879	-71.13	-	-	-	-	-	-	-	-	-	-	-
		5		4766315	-71.75	-	-	-	-	-	-	-	-	-	-	-
		6		24379553	-70.04	8	-55.85	1.46	864	486	1.30	-61.64	1.71	138	64	1.29
	1	3	4881	2934	-58.29	-	-	-	-	-	-	-	-	-	-	-
		4		107945	-69.63	-	-	-	-	-	-	-	-	-	-	-
		5		981253	-72.88	-	-	-	-	-	-	-	-	-	-	-
		6		5074824	-74.19	1	-51.42	1.82	2077	1109	1.37	-51.42	1.82	2077	1109	1.37
	2	3	2262	222	-53.98	-	-	-	-	-	-	-	-	-	-	-
		4		10052	-59.20	-	-	-	-	-	-	-	-	-	-	-
		5		95757	-72.88	-	-	-	-	-	-	-	-	-	-	-
		6		513748	-74.19	-	-	-	-	-	-	-	-	-	-	-
LG	0	3	8473	14710	-60.40	-	-	-	-	-	-	-	-	-	-	-
		4		540879	-71.76	-	-	-	-	-	-	-	-	-	-	-
		5		4766315	-71.75	-	-	-	-	-	-	-	-	-	-	-
		6		24379553	-70.04	7	-55.85	1.46	3890	517	1.30	-61.64	1.71	1018	77	1.29
	1	3	4881	2934	-58.29	-	-	-	-	-	-	-	-	-	-	-
		4		107945	-69.63	-	-	-	-	-	-	-	-	-	-	-
		5		981253	-72.88	-	-	-	-	-	-	-	-	-	-	-
		6		5074824	-74.19	1	-51.42	1.82	3566	1201	1.37	-51.42	1.82	3566	1201	1.37
	2	3	2262	222	-53.98	-	-	-	-	-	-	-	-	-	-	-
		4		10052	-59.20	-	-	-	-	-	-	-	-	-	-	-
		5		95757	-72.88	-	-	-	-	-	-	-	-	-	-	-
		6		513748	-74.19	-	-	-	-	-	-	-	-	-	-	-

found by NUCLEAR ($\#oligo$ in Tables 5.1-5.3) pass from null in the 2XNR case (where equivalent searches gives the same number of solutions) to modest in 5WWX and 5ELH where local searches produce on average 7.5% and 7.4% less solutions than global ones, respectively.

As a global behavior, the minimum energy ($min\ E$ in Tables 5.1-5.3) of minimized oligo-nucleotides slightly decreases (or stays constant) with the increase of d(C5'-O3') for a given value of RMSD cutoff, meaning that bigger search spaces may help in finding less energetic solutions. This trend is consistently followed in the 2XNR case, with only an exception in 5WWX (LA15) and some slight anomalies in 5ELH (XY06 cases).

When $min\ E$ is compared between equivalent searches (GA vs. GG and LA vs. LG to assess the effect of arithmetic vs. geometric pre-scoring or GA vs. LA and GG vs. LG to assess the effect of global vs. local search space), there are only minor differences indicating no clear trend for crowning a best XY methodology.

As the search space grows (either by increasing d(C5'-O3') or by clustering with low values of RMSD cutoff), the probability of finding native-like structures ($\#native$ in Tables 5.1-5.3) also augments because there are more native-like mono-nucleotides being considered. For 2XNR, only searches performed at d(C5'-O3') of 5 or 6 Å produced native-like solutions, with a maximum of 541 structures (GA06, LA06) and a minimum of 4 (GA25, GG25, LA25, LG25). In 5WWX the number of native-like structures considerably drops down to a maximum of 7 (XY15, GG06, and LG06) and a minimum of 1 (XY26). In 5ELH the trend resembles 5WWX, with a maximum of native-like structures retrieved of 8 (LA06), and a minimum of 1 (XY16). In this latter case, explorations conducted after clustering with 2Å RMSD cutoff did not report any native-like structure.

As the primary goal of this section was to reproduce crystal structures, our attention was focused on two distinctive solutions for each XYij distribution; (i) the most similar solution to the experimental one and (ii) the best-scored solution among the native-like ones.

In the 2XNR case, the most similar solution to the experimental one is almost always the same for equivalent searches, except in GG05 vs. GA05, GG16 vs. GA16 LA16 vs. GA16, LG05 vs. LA05, and LG16 vs. LA16. The biggest energy gap between these divergent cases was 1.64 kcal/mol. These structures' similarities against the reference vary from 0.93 to 1.73 Å. For 5WWX and 5ELH the most similar solution to the experimental one is always the same for equivalent searches and their resemblance fluctuates from 1.06 to 1.96 Å and from 1.46 to 1.82 Å, respectively.

The most similar solutions are ranked before and after minimization in their corresponding XYij distribution. In 2XNR their ranking before and after minimization ranges from 54 to 4903 and from 21 to 644, respectively. In 5WWX the ranks fluctuates from 659 to 3370 before and from 149 to 1005 after minimization. For 5ELH pre-minimization ranks of the most similar solution to the experimental one ranges from 864 to 3930 while they drops from 486 to 1223 after minimization. This is indicative that the pre-score of oligo-nucleotides is not linearly related to the score they will have after minimization. Note that the energies of these structures are distant from the global minimum of their XYij

distribution (5.2-12.08 kcal/mol, 7.9-11.8 kcal/mol, and 14.2-22.8 kcal/mol for 2XNR, 5WWX, and 5ELH respectively). The RMSD deviation after the linking process of connectable nucleotides found by NUCLEAR with respect to the original positions initially found by MCSS are in the interval 1.80-2.58 Å in 2XNR, 1.23-2.22 Å in 5WWX, and 1.30-1.37 Å in 5ELH.

Concerning the best-scored solutions found by NUCLEAR among the native-likes for each XYij distribution, only in the 2XNR case there are divergences in equivalent searches (LA16 vs. GA16 and LA16 vs. LG16). The similarities of these best-scored native likes structures vary from 1.35 to 1.98 Å in 2XNR, from 1.35 to 1.97 Å in 5WWX, and from 1.71 to 1.82 Å in 5ELH. Let us analyze their ranking inside the XYij distribution in which they were obtained before and after minimization: 5-1388 and 15-96 for 2XNR, 659-3635 and 122-296 for 5WWX, and 138-3630 and 63-1223 for 5ELH. It is worth noting that the energy of these structures are distant from the global minimum of their XYij distribution (2.48-8.88 kcal/mol, 5.46-9.34 kcal/mol, and 8.4-22.7 kcal/mol for 2XNR, 5WWX, and 5ELH respectively). The RMSD deviation after the linking process of connectable nucleotides found by NUCLEAR with respect to the original positions initially found by MCSS are in the interval 1.40-1.98 Å in 2XNR, 1.40-2.50 Å in 5WWX, and 1.29-1.37 Å in 5ELH.

5.5 . NUCLEAR's complexity notes

In Figure 5.1, we detailed the general workflow for NUCLEAR software. Two different tasks can be accomplished: searching for either hotspots or oligonucleotides. In this Section we advance some notes on the temporal and spatial complexities of the different steps conducted by NUCLEAR. It is important to note that even if algorithm complexity gives idea on the worst performances an algorithm can have, real situations often are better than the worst case and carefully chosen parameters could help in reaching practical performances. Table 9.10 was constructed to grab an idea on the NUCLEAR's running times and memory consumption for the case studies presented in Section 5.4.2.

In Figure 5.1, we detailed the general workflow for the NUCLEAR software, which can accomplish two distinct tasks: searching for either hotspots or oligonucleotides. This section analyzes the time and space complexities of the various steps conducted by NUCLEAR. It is important to note that while algorithmic complexity provides insight into the worst-case runtime, real-world situations are often better than the theoretical worst case. Careful parameter selection can aid in achieving practical performance. Table 9.10 was constructed to showcase NUCLEAR's running times and memory usage on the case studies presented in Section 5.4.2, giving empirical measurements.

Though some steps exhibit high polynomial complexities, appropriate cutoffs allow NUCLEAR to perform reasonably on test cases, avoiding the deterioration predicted by worst-case analysis. Still, future work could aim to reduce the complexity of critical steps, such as clustering and construction of the clash/reachability matrices. With algorithmic improvements, NUCLEAR may scale more efficiently to large numbers of fragments and

replicas while retaining its demonstrated utility for FBDD efforts.

5.5.1 . Search of hotspots

In Figure 5.2, the main steps for a NUCLEAR oligonucleotide search are shown: (i) parsing molecular data, (ii) calculating interaction fingerprints, (iii) clustering fingerprints into proper subsets, (iv) filtering, and (v) merging similar clusters.

Parsing molecular files can be regarded as an $O(n)$ operation concerning the number of fragment conformations to analyze. Interaction fingerprints are computed using the protein coordinates for each fragment conformation and can be assumed as a constant operation $O(k)$ since the time to compute a given fingerprint is independent of the total number of fragments. As this step is executed for every fragment, the overall complexity is $O(kn) \equiv O(n)$.

The clustering step (described previously in Section 2.1.4) depends on the number of fragments. In the worst case where no fragment can be designated as similar, the procedure attempts for each fragment ($O(n)$) to compute its similarity ($O(k)$) against all available fragments ($O(n)$), thus reaching an overall complexity of $O(n) \cdot O(k) \cdot O(n) \rightarrow O(kn^2) \equiv O(n^2)$.

Filtering against a cutoff value is a constant action performed n times in the worst case, so this step is linear ($O(n)$). The worst case for the final operation occurs if no cluster can be merged, a failed constant action repeated n times ($O(n)$). After summing the previous contributions, we can conclude that the NUCLEAR search of hotspots has the quadratic time complexity of its dominant term, the clustering step.

The spatial complexity of hotspots search can be evaluated by analyzing the objects loaded into RAM at each previously mentioned step. Parsing molecular data requires linear space with respect to the number of fragments to analyze ($O(n)$). The fingerprint calculation generates one fingerprint ($O(k)$) for every fragment ($O(n)$), thus also consuming linear space. Clustering fingerprints necessitates loading all fragments into memory ($O(n)$). Since the similarity computations are performed *in situ*, once per element, and all elements appended to a cluster are eliminated from further consideration, this step also exhibits linear spatial complexity (no similarity matrix is used). The filtering and merging operations do not increase the spatial complexity either, so the dominant term of the complete procedure is linear.

5.5.2 . Search of oligonucleotides

The primary steps involved in NUCLEAR's oligonucleotide search are (i) clustering of MCSS docking distributions, (ii) construction of reachability and clash matrices, and (iii) linking mono-nucleotides to produce sequences.

As previously discussed, the BitClust-inspired clustering is executed independently for each docking distribution, requiring $O(n^2)$ time (n being the number of replicas per distribution). Thus, for m distributions, the time complexity of this step is quadratic, $O(mn^2)$. A KD-TREE containing two atoms per replica ($O3'$ and $C5$) is utilized to construct the reachability matrix. With the KD-TREE built, each $O3'$ atom searches in logarithmic time $O(nlogn)$ for neighboring atoms within a cutoff. The clash matrix uses

a KD-TREE of all replica atoms. Detecting clashes between a replica's atoms and others' is done by finding nearest neighbors within a cutoff, an $O(nlogn)$ operation.

Finally, mono-nucleotide linking follows a DFS traversal of the reachability graph in $O(n+e)$ time (n = replicas, e = graph edges) since each node and edge is visited once. Currently, NUCLEAR performs this from every node, increasing the complexity to $O(n^2+ne)$. Overall, quadratic terms plague NUCLEAR's performance. However, as Table 9.10 shows, carefully chosen cutoffs can prevent deteriorated performance.

As discussed in Section 5.5.1, NUCLEAR clustering has linear spatial complexity. The constructed binary matrices are quadratic, dominating overall spatial complexity. In the linking stage, each connected node saves a reachability matrix vector. In the worst case, where all nodes are linkable, this grows quadratically. Nevertheless, linkable nucleotides (b) are limited for practical use, giving $O(bn)$ complexity.

6 - *IN-SILICO* DESIGN OF SELECTIVE CMOs AGAINST BACE-X

Once the required methodological scaffold developed, we can tackle this work's primary goal: the *in-silico* fragment-based design of selective Chemically Modified Oligonucleotides (CMOs) as potential protein inhibitors. We will use the BACE-X protein as a relevant case study.

After presenting the nature of the chemically modified mono-nucleotides and the protein conformations under consideration (Section 6.1), we illustrate how to attack the problem of assembling oligonucleotides in two situations. Where practical information on protein-inhibitor interactions is limited or unavailable (Sections 6.2.1 and 6.2.2), and where knowledge on these interactions is accessible through molecular databases (Section 6.2.3). In the last part of the chapter, we then focus our attention on discerning if produced oligonucleotides have the potential to be selective against the BACE1 protein over BACE2 by implementing several selection constraints to their binding modes (Section 6.3).

6.1 . Modified nucleotides and BACE-X protein candidates selection

As already discussed in Section 1.4.1, most inhibitors designed against BACE1 present off-target interactions with the conformationally similar protein BACE2, a negative side effect that leads to the inhibition of both proteins. Hence the primary motivation of our workflow is to design oligonucleotides binding BACE1 preferentially.

After a full search for representative conformations of BACE1 and BACE2 proteins in the PDB (see Sections 2.4.1 and 2.4.2 for details), we chose four models of the first one (1SGZ, 4GID, 5MCQ, and 6UVV) and two of the latter one (2EWY and 3ZKM).

Once the protein models were established, the MCSS nucleotide fragments were selected. A total of 111 mono-phosphate (including standard and chemically modified nucleotides) were considered (see Section 2.4.3 for the complete list). The chemical modifications in the library of the MCSS include all of the natural nucleotides found in living organisms. They are available from commercial providers to guarantee that the designed oligonucleotides can be chemically synthesized.

Apart from the standard nucleotides (A, C, G, U/T), all the other nucleotides from the library include a modified nucleic acid base derived from the corresponding standard one (19 A-derived, 18 C-derived, 29 G-derived, 45 U/T derived). In addition to the nucleobase's modification, some also include a modification on the ribose with a 2'-OMe group. The 2'-OMe modification has been widely used to produce synthetic oligonucleotides and has been shown to improve the therapeutic index[273].

Each fragment (mono-nucleotide) was docked onto the entire surface of each protein

conformation (not only their binding site) via MCSS software, yielding 666 distributions of about 10000 replicas that constitute the underlying data for the rest of the design workflow.

6.2 . Definition of the receptor region to explore

When we analyzed the performance of NUCLEAR in Section 5.1, it became evident that the selection of even a moderate number of linkable nucleotides (docked at the protein surface) may lead to a combinatorial explosion of the oligonucleotide candidates to output. The most natural choice to deal with this problem is to reduce the number of nucleotides to connect in a particular search. This reduction can be accomplished in the NUCLEAR context at least by three strategies; (i) performing a clustering step before starting the oligonucleotide search, (ii) constraining the sequence to search for, and (iii) focusing the search on a docked sub-region of interest at the receptor.

Clustering the docked distributions reduces the search's exhaustiveness but at the expense of eliminating poses that might lead to the best hits. We evaluated this effect in Section 5.4 when struggling to find native-like poses of known oligonucleotide-protein complexes. So we consider that it is more appropriate to do so as an optional second pass to discover non-similar oligonucleotides.

As we are at the initial stages of the design of oligonucleotides, there is no particular reason to introduce a bias in favor of a given sequence neither, so limiting the search to a particular sub-region was the chosen way to deal with the aforementioned combinatorial explosion.

However, discriminating essential protein sub-regions from the irrelevant ones is not trivial in cases where there is no previous knowledge of how the protein binds to its inhibitors and, more specifically, to the particular (potentially novel) fragments we want to explore in the design. Next, we detail three approaches to restrict the exploration regions: through the detection of hotspots (Section 6.2.1), from selectivity analyses of the protein residues (Section 6.2.2), and from experimental knowledge of the already-known binding sites of the protein (Section 6.2.3).

6.2.1 . Region definition from NUCLEAR hotspots

Information about the protein regions having a high propensity for ligand binding, also known as hotspots, is essential in FBDD workflows. That is why NUCLEAR incorporates the ability to detect such hotspots, helping to ignore unimportant binding regions (relative to the fragments under concern) and consequently improving the performance of subsequent oligonucleotide searches.

For every conformation of BACE-X, we instructed NUCLEAR to execute a hotspots search (described in Section 5.2) from the corresponding distributions of non-clustered docked fragments. All protein atoms (but only no-patch fragment atoms) were evaluated for determining NUCLEAR fingerprints. Only spots with a relative density greater than 0.05 were considered, and those whose seeds' Tanimoto Index (TI) was under 0.25 were merged.

The spots found with the previous procedure for the 4GID conformation of BACE1 are depicted in Figure 6.1 for illustration purposes. Note that some of them may appear small because of the superposition with adjacent ones. In other applications, users may want to focus the search on a subset of adjacent spots. However, we decided to use them all for a more encompassing search area. So for every BACE-X conformation, all spots having at least one fingerprint coming from a fragment ranked in the first top 100 of its particular MCSS distribution were selected.

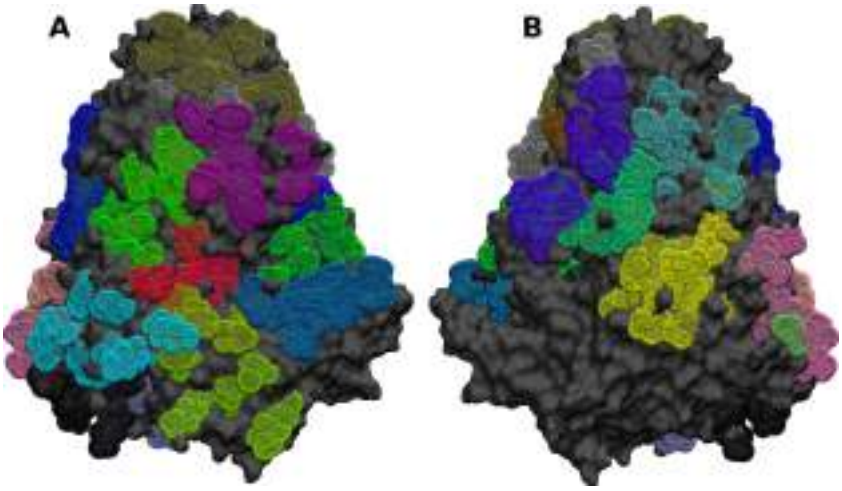


Figure 6.1: NUCLEAR *hotspots detected in 4GID protein. (A) Protein side containing the experimental binding site (in red). (B) The opposite side (180 deg. rotation around the y-axis from side A).*

6.2.2 . Region definition from residues' selectivity

The previous search region definition is relative to the particular conformation for which the NUCLEAR search of spots was conducted. Such an approach provides no insightful information on the average selectivity or binding preference of fragments against one or another protein. As we are interested in designing selective oligonucleotides against BACE1 (and not just against a particular conformation), we found it reasonable to define the exploration region upon those residues showing a consensus selectivity for BACE1 over BACE2. These details may be challenging to obtain from experiments but are recoverable from an *in-silico* approach.

Suppose that for a given BACE-X conformation, we compute the fingerprints of all fragment poses issued from the MCSS docking. Each of those fingerprints comprises several protein residues (see Section 5.2). By concatenating all fingerprints found on a given conformation j, it is possible to count the total number of times a protein residue i gets involved in contacts ($counts_{ij}$).

Let us define the BACE-X counts of a residue i, as its average count for each BACE-X conformation: $BX\_counts_i = \overline{counts_{ij}} : \forall\ j\ in$ BACE-X$_list$, where X can be 1 or 2 and BACE-X$_list$ comprises 1SGZ, 4GID, 5MCQ, and 6UVV conformations for

$X = 1$, and 2EWY and 3ZKM conformations for $X = 2$. In this context, the BACE1 selectivity of a residue i was computed as:

$$S_i(B1) = B1\_counts_i/(B1\_counts_i + B2\_counts_i) \quad (6.1)$$

While residues of the identical BACE-X conformations can be trivially re-numbered to have an equivalent enumeration, BACE1 and BACE2 do not have complete sequence identity. Then, a residue i on BACE1 may not exist or be displaced in the BACE2 sequence and vice versa. So, for determining the residue's selectivity, we faced the non-trivial problem of finding a three-dimensional equivalence between BACE-X residues identity and enumeration that we approached as explained in Section 2.4.5.

It is clear that the selectivity notion of Equation 6.1 offers just a piece of approximate information on residue binding preference. The panorama would have changed if another set of conformations were selected for BACE-X (or the evaluated fingerprints were restricted to the best-ranked top-X). In any case, the derived selectivity maps advance valuable clues on the overall binding preferences of proteins.

As it is verified in Figure 6.2, vast regions are selective to BACE1 (intense blue). Note that although the map has been projected onto a particular BACE1 conformation (4GID), the selectivity per residue could have been mapped to any other BACE-X conformation. Suppose we concentrate on the lower region of Figure 6.2B. In that case, we observe that it appears highly selective to BACE2, the reason behind hotspots' absence in this region for BACE1 conformations (see the equivalent region in Figure 6.1). So for every BACE-X conformation, the region search defined from selectivity information consisted of residues with more than $S >= 0.5$ for BACE1 and $S < 0.5$ for BACE2.

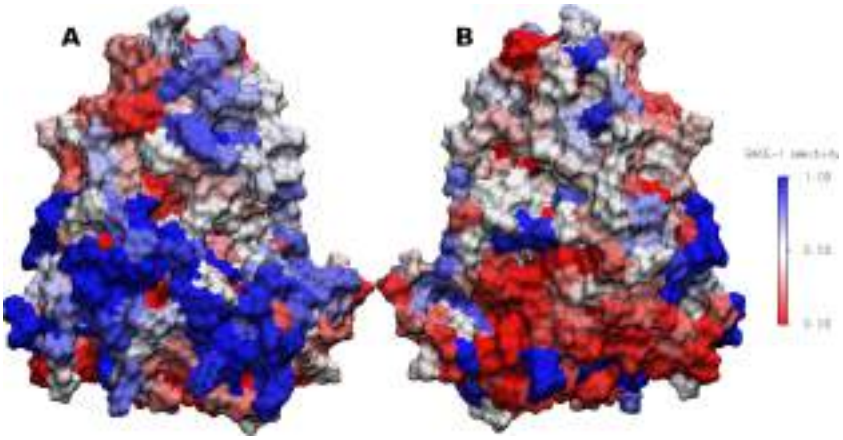


Figure 6.2: *Selectivity map of residues computed from NUCLEAR fingerprints projected onto the 4GID BACE1 conformation. (A) Protein side containing the experimental binding site. (B) The opposite side (180 deg. rotation around the y-axis from side A). BACE1 selectivity of residues are color coded from zero (red residues selective to BACE2) to one (blue residues).*

6.2.3 . Region definition from experimental protein-inhibitors contacts

Previous definitions of the region to explore lead to a considerable shrink of the receptor surface to evaluate. They are both *in-silico*-inspired reductions that have no ultimate link with experimental knowledge of the protein's binding site. Provided this knowledge exists, one could use it to define where to conduct NUCLEAR searches. As described in Section 2.4.1, we downloaded BACE-X inhibitors reported on the PDB database and computed their fingerprints using the BINANA software to gather the interacting residues.

In Figure 6.3, it is appreciated how by only selecting those residues that interact with an experimentally tested inhibitor, the search space is now much more reduced than the formerly defined in Sections 6.2.1 and 6.2.2. From a design perspective, this characteristic limits the amount of novel binding areas to consider. Nevertheless, at the same time, it provides a zone that may contain a lot of well-ranked oligonucleotides.

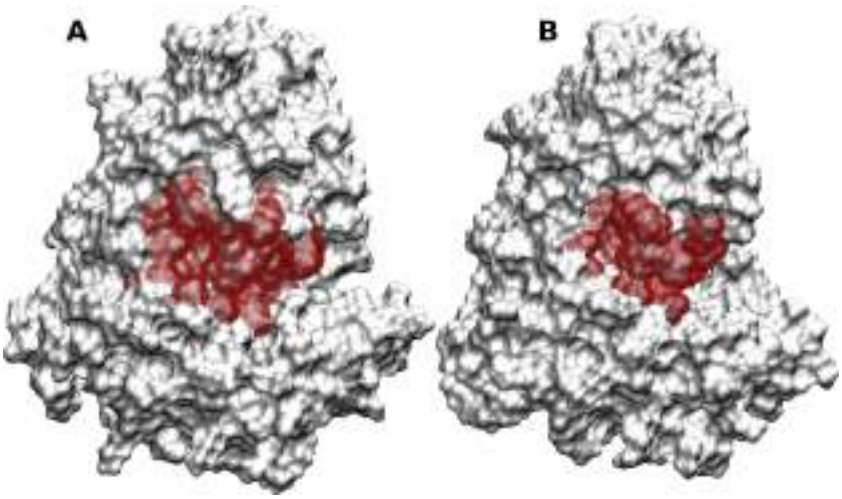


Figure 6.3: *Region definition from protein-inhibitors contacts computed with BINANA for (A) BACE1 and (B) BACE2 conformations.*

Note that as the number of BACE1 inhibitors reported in PDB is more prominent than for BACE2, the latter has a slightly smaller region defined by this methodology. They are spatially equivalent, in any case.

6.3 . Selectivity of CMOs against BACE-X

The general workflow to retrieve selective CMO against BACE-X is depicted in Figure 6.4. After choosing one of the approaches mentioned above to define the protein's region where the NUCLEAR oligonucleotide search should proceed (Sections 6.2.1-6.2.3), resulting selections were used to search for the 1000 best-ranked chains with sizes between 2 and 6 nucleotides onto the surface of each protein conformation. The first 100 best-ranked poses of every fragment distribution obtained by the MCSS docking were

considered. The distance to catch an atomic clash between protein and fragment atoms was set to 1.5 Å and the maximum distance to join nucleotides (from $O3'$ to $O5'$) to 6 Å.

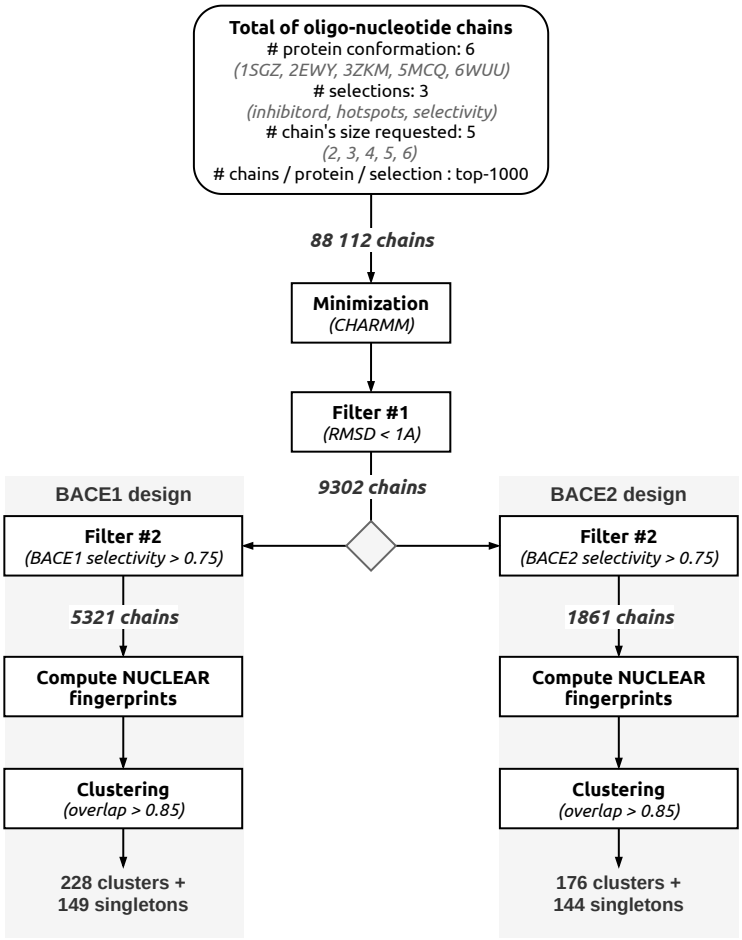


Figure 6.4: *General workflow to retrieve selective* CMO *for* BACE-X *proteins.*

The 88112 found oligonucleotides were then submitted to a minimization protocol using CHARMM (see Section 2.3.1). At this point, a necessary clarification should be made; for a given MCSS exploration conducted on a particular protein conformation, NUCLEAR searches deliver unique oligonucleotide conformations. In the same job, for example, it is possible to output the sequence **ADE GUA THY** more than once, ranked differently depending on the conformation it adopts upon protein binding after minimization.

In a post-minimization stage, the following descriptors were requested for each produced chain in order to derive rules for later selecting the most appropriate candidates to inhibitors: (i) their *interaction energy* (EINT) with the protein (requested in the CHARMM minimization script), (ii) their *post-minimization ranking* (after sorting by their EINT), (iii) their *number of contacts* with the protein (after minimization) (iv) the

Tanimoto Index (TI) of the pre and post-minimization NUCLEAR fingerprints, (v) the RMSD of the pre and post-minimized coordinates, (vi) their number of conformers, and (vii) their BACE1 selectivity (computed after Equation 6.1).

Selecting a potential inhibitor against BACE1 is a challenging task that can not be estimated using only these descriptors. We proceeded with that information because no other analyses were implemented when writing this manuscript. We are still confident that the chosen ones (easily derived with NUCLEAR) furnish users with practical power to discern potential hits from sub-optimal solutions. Also, the cutoffs used to restrict the number of sequences to analyze and the choice of the different top-X should not be blindly embraced as gold standards and are provided in a demonstrative basis.

The ensemble of chains was arranged in a tabular data structure and ordered by EINT (ascending), selectivity to BACE1 conformations (descending), and the number of contacts (descending). Then two ramifications were followed to generate BACE1 or BACE2 selective CMOs. Those sequences whose RMSD after minimization was less than 1 Å or whose BACE-X selectivity was above 0.75 were conserved (5321 chains for BACE1 and 1861 chains for BACE2).

The conserved sequences were submitted to the clustering procedure described in Section 2.1.4. The similarity criteria, however, was the overlap of their fingerprints (those overlapping more than 0.85 were considered similar). In this way, 228 clusters (plus 149 singletons) were obtained for BACE1 and 176 clusters (plus 144 singletons) were retrieved for BACE2. In Figure 6.5, the first five best ranked seeds for each protein are shown, while more information about these seeds is reported in Table 6.1.

As shown in Table 6.1, the clusters of BACE1 contain a higher percentage (13%) of the selected chains (5321 in total) compared to the clusters of BACE2, which account for approximately 6% of the selected chains (1861 in total). All the seeds in both clusters are hexamers, which aligns with the maximum chain size requested in the NUCLEAR jobs.

Despite the lack of direct comparability and variations in interaction energy among the different seeds, they are all well-ranked within their respective NUCLEAR distribution (from a minimum of 3 to 174 out of the 1000 solutions requested). It is noteworthy that the selected CMOs exhibit a low deviation before and after minimization, as evidenced by their TI and RMSD values. Lastly, it is important to highlight that all compared seeds demonstrate complete selectivity towards the protein in which they were identified.

The seeds of the first five clusters associated with potential selective inhibitors of BACE-X are depicted in Figure 6.5. In the case of BACE1, despite being the highest-ranked solution, C1 adopts a U-like conformation and does not contact the active site region. Similarly, C2 does not introduce a complete nucleotide into the active site; only the T6A nucleobase is present, resulting in a T-like conformation. In contrast, C3 and C5 exhibit a more conventional curvilinear arrangement, effectively crossing the active site region and establishing a characteristic set of interactions with the flap region through their common 7OU and OAU mono-nucleotides. The most atypical CMO binding mode retrieved for BACE1 is C4. In this case, the hexamer lands over the left side of the flap and extends until the final, unmodified GUA nucleotide, gets inside the active site.

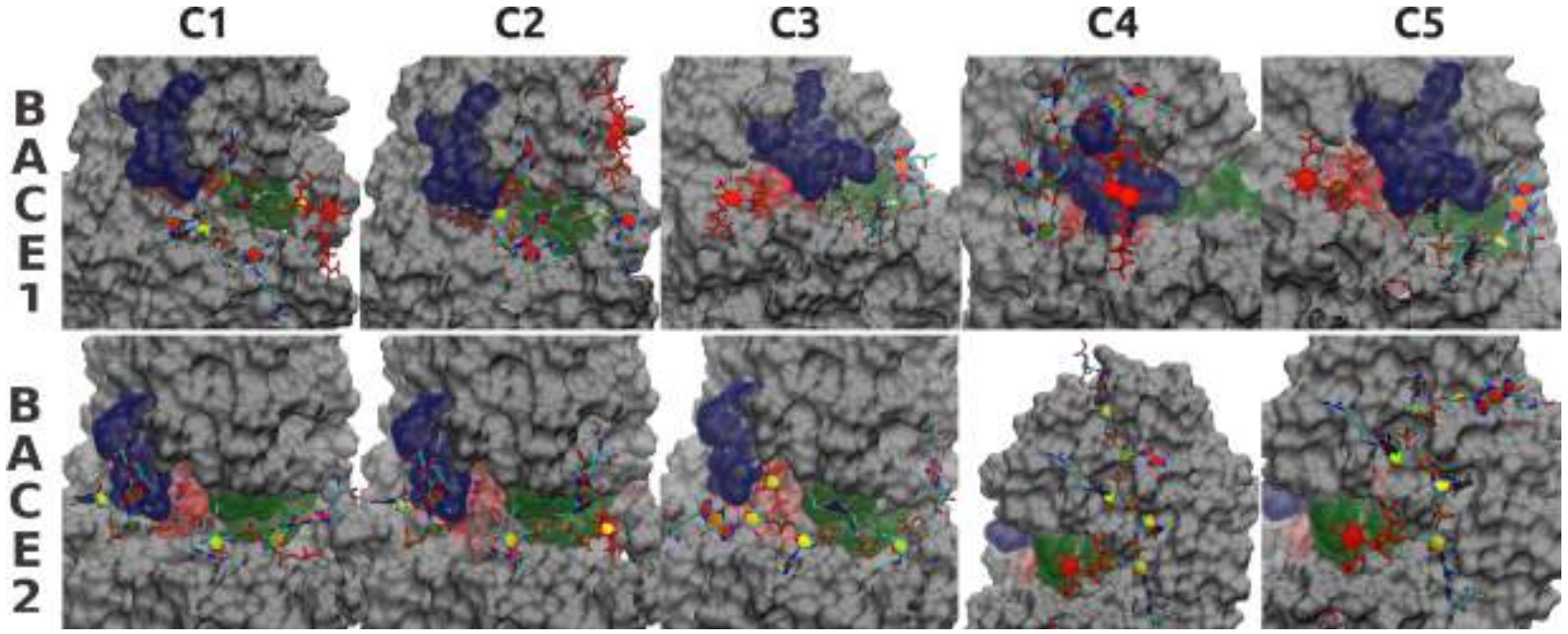


Figure 6.5: *First five clusters' seeds of BACE-X's potential selective inhibitors. Red, blue, and green regions correspond to the active site, the flap, and the 10s loop regions, respectively. Oligonucleotides are represented without protons and phosphorus atoms have been highlighted for improved visual clarity. The red nucleotide starts the oligomer whose sequence is shown in Table 6.1.*

Regarding BACE2, the hexamers C1 and C2 exhibit a shared terminal motif (PBG-OMG), giving rise to a distinctive interaction pattern. In this pattern, PBG covers the upper-right side of the flap, while OMG is inserted within the active site region. In a reversed orientation, C3 replicates the aforementioned binding mode, featuring a lateral interaction between BUG and the flap, while ADE is inserted into the active site. On the other hand, C4 and C5 seeds are situated significantly distant from the active site region.

6.3.1 . Key interactions of CMO-BACE-X complexes

The molecular basis underlying the binding modes of Figure 6.5 was investigated by examination of the close contacts, hydrogen bonds, salt bridges, hydrophobic, $\pi - \pi$, T-stacking, and $\pi-$cation interactions between the CMOs and BACE-X. This analysis was conducted utilizing the default's geometrical values of the BINANA program and for all members of the inspected clusters. The outcomes are presented in Figure 6.8 for BACE1 and Figure 6.9 for BACE2.

For BACE1, residues in the "*near-10s loop region*" displayed the highest interaction frequency (except for cluster 4, which does not extend to that area). This region includes the residues LYS-9, GLN-12, ASN-111, GLN-163, ARG-307, VAL-309, GLU-310, ASP-311, and LYS-321 (refer to Figure 6.6A). Interactions with LYS-9, GLU-310, and ASP-311 are particularly prevalent in clusters 1 and 2, exhibiting close contacts, hydrogen bonds, hydrophobic interactions, and salt bridges. ARG-307 and LYS-321 are also among the most frequent residues, forming similar types of interactions. Though rare, LYS21 can form $\pi-$cation interactions with a few exemplars. While GLN-12 can form hydrogen bonds, hydrophobic interactions are more common. VAL-309 only gives rise to hydrophobic interactions. ASN-111 and GLN-163 interact with members across the four clusters, though their frequency is low.

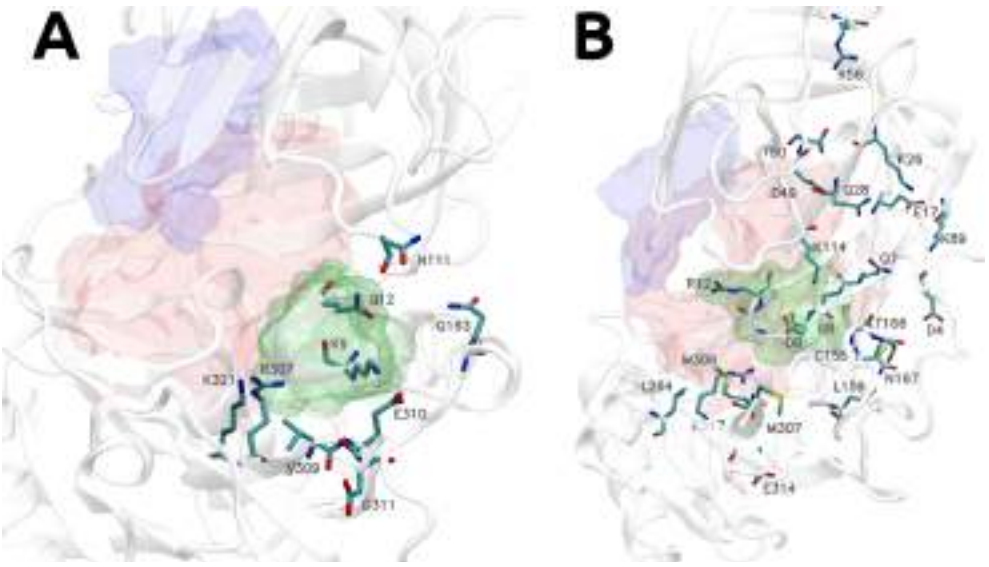


Figure 6.6: *BACE1's (A) and BACE2's (B) "near-10s loop region" key residues interacting with the members of the first five clusters of selective CMOs retrieved by NUCLEAR.*

By contrast, the "near-10s loop region" of BACE2 (Figure 6.6B) contains more and different residues with prevalent interactions: GLY8, ASP9, GLY11, ARG12, GLY112, LYS114, GLY165, ASN167, LEU264, MET306, MET307, GLU314 and ARG317. This region is shared by the two significantly distinct binding modes that CMOs adopts in this protein: that of clusters 1, 2, and 3 (that also covers the active site region) and the one seen in clusters 4 and 5 (positioned in the outer right side of the 10s loop, without any contact with the active site and thus not essential to the present study). Consequently, interacting residues are seldom present with the same prevalence across the five clusters. The ARG12, able to form hydrogen bonds across all clusters and salt bridges in C1 to C3, stands as one of the residues with more interactions. LYS114 is the only residue in the region to establish $\pi-$cation interactions with a few exemplars. From the two consecutive methionines, MET307 forms hydrogen bonds across all clusters, though they seem stronger for clusters 3, 4, and 5, where close contacts are detected with this residue.

In Figure 9.19, the interacting residues of this region are colored by their type. In BACE1, there are three basic residues together: an acidic dyad and a polar sub-region. There is no close acidic pair for BACE2, and a non-polar chain outlines one of the two basic residues. The distinctive composition of this particular area, coupled with the significant frequency of interactions established with the CMOs, strongly suggests their relevance for the design of BACE-X selective inhibitors (currently focused on the protein's active site).

Examining the active site region (refer to Figure 6.7), we observe distinct differences in the composition and significance of the interacting residues in each protein. Notably, the flap region (TYR71, THR72, GLN73) remains conserved between both proteins. The claw-like interaction observed in clusters 3 and 5 of BACE1 results from $\pi - \pi$ stacking with TYR71 and hydrogen bonds with THR72. These residues also exist in BACE2, although the $\pi - \pi$ stacking is observed solely in cluster 3. GLN73 is more prevalent in BACE2 clusters than in BACE2.

A critical distinction between BACE-X interaction patterns pertains to the catalytic aspartic dyad (ASP32-ASP228 in BACE1, ASP32-ASP225 in BACE2). A limited number of members from cluster 4 in BACE1 exhibit hydrogen bonds with ASP228 and close contacts with ASP32. This particular location assumes a more significant role in BACE2, where members of clusters 1, 2, and 3 can establish hydrogen bonds, salt bridges, or hydrophobic interactions with ASP32 or ASP225.

The unusual binding mode of cluster 4 in BACE1, supported by $\pi - \pi$ stacking (TYR68), hydrogen bonds (LYS75, GLU77, SER328), and salt bridges interactions (LYS75 and GLU77), does not have a counterpart in BACE2. This binding, while anomalous, aligns with other authors' recommendations of targeting the flap region, where the shape and flexibility differ between BACE-X enzymes[21,172].

Overall, despite the claimed similarity between BACE-X proteins, our analysis reveals that the highest-ranked CMOs interacting with them are positioned differently, not just within the active site but also in other protein sub-sites crucial for molecular anchoring.

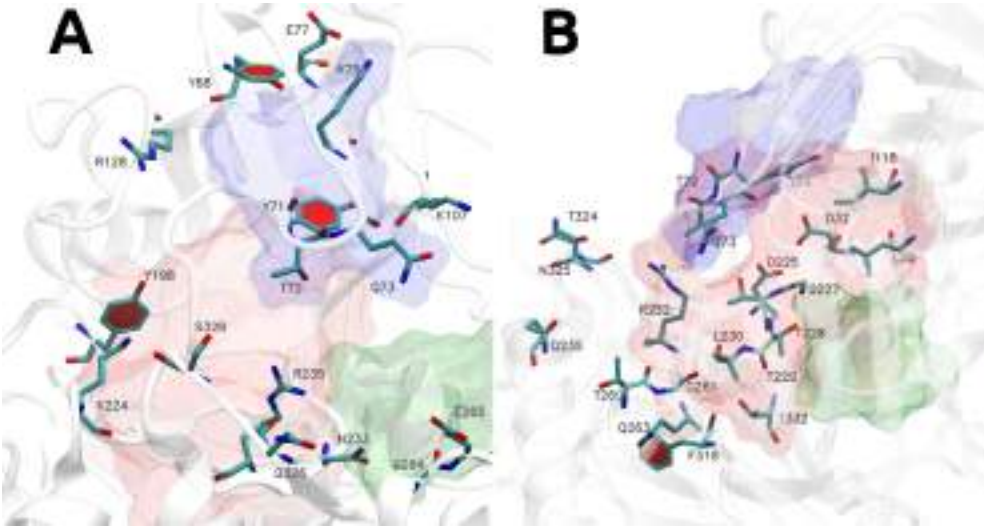


Figure 6.7: BACE1*'s top (A) and* BACE2*'s right lateral (B) view of the active site region's key residues interacting with the members of the first five clusters of selective* CMOs *retrieved by* NUCLEAR*.*

Table 6.1: *Descriptors of potential selective inhibitors of* BACE-X*. The Selectivity column refers to* $S_i(B1)$ *or* $S_i(B2)$ *for* BACE1 *or* BACE2*, respectively.*

Protein	Clust. ID	Clust. size	Chain	EINT	Rank	#contacts	TI	RMSD	Selectivity
				[kcal/mol]				*[Å]*	
BACE1	1	105	R2C-OAU-1MA-OAU-3MC-R2C	-177.46	13	24	0.82	1.00	1.00
	2	268	HWG-R2C-OAU-1MG-T6A-SIA	-170.97	11	29	0.90	0.97	1.00
	3	119	13P-70U-OAU-BUG-6IA-HWG	-165.99	22	31	0.88	0.92	1.00
	4	160	HNA-OAU-K2C-HCU-OAU-GUA	-165.77	3	23	0.88	1.00	1.00
	5	36	OAU-70U-OAU-BUG-MMA-HWG	-163.89	43	28	0.87	0.88	1.00
BACE2	1	16	3MC-YYG-K2C-3AU-PBG-OMG	-180.66	10	31	0.77	0.98	1.00
	2	11	5CU-PBG-K2C-3AU-PBG-OMG	-178.5	18	34	0.78	0.99	1.00
	3	36	ADE-BUG-5HU-5CU-MAU-BUG	-168.2	174	28	0.71	0.98	1.00
	4	8	R2C-BUG-MSU-R2C-ADE-HWG	-159.84	20	27	0.83	1.00	1.00
	5	34	R2C-BUG-THY-R2C-5HU-PBG	-159.32	22	27	0.93	0.93	1.00

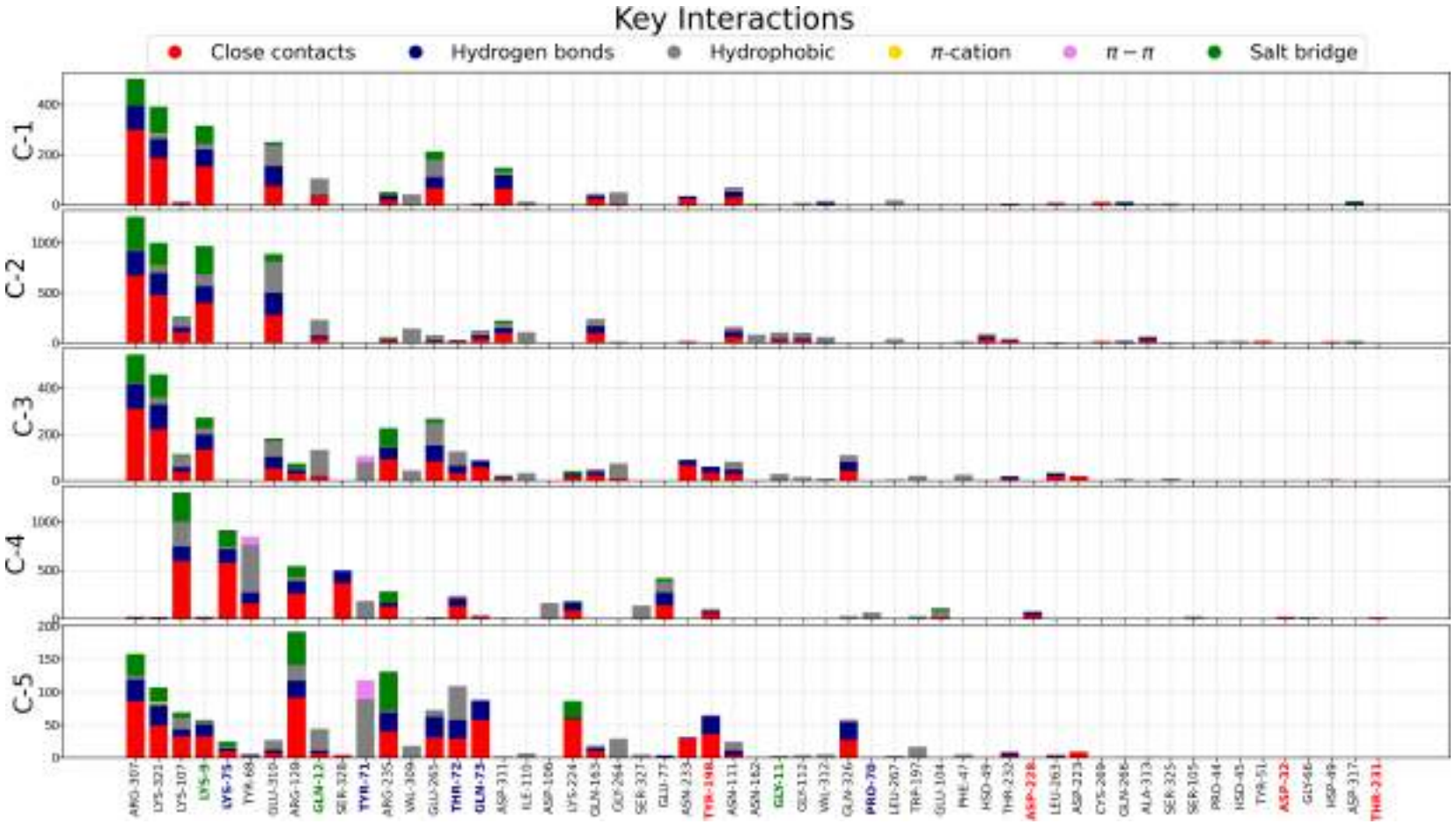


Figure 6.8: *Count of molecular interactions formed between all constituents of the first five clusters (C1 to C5) of the CMOs targeted against BACE1. Residues on the x-axis are colored red, blue, or green if they are part of the active site region, flap, or 10s loop, respectively.*

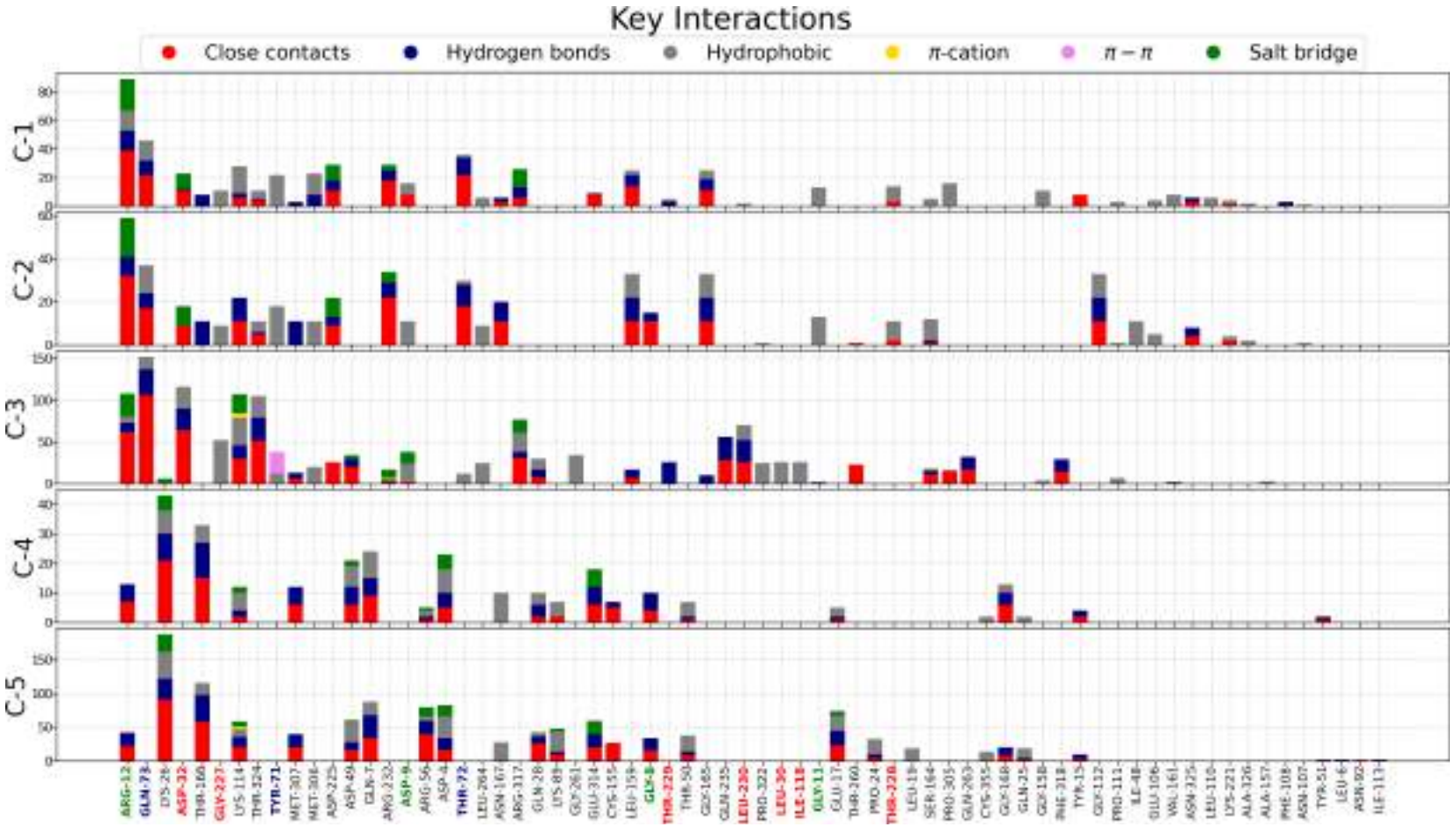


Figure 6.9: *Count of molecular interactions formed between all constituents of the first five clusters (C1 to C5) of the* CMOs *targeted against* BACE2*. Residues on the x-axis are colored red, blue, or green if they are part of the active site region, flap, or 10s loop, respectively.*

7 - CONCLUSIONS

1. The Multiple-Copy Simultaneous Search software was evaluated for docking nucleotides on a benchmark of 121 protein complexes. Different solvent and phosphate models were tested to optimize the success rate for identifying native poses (docking power) and the actual native nucleotide (screening power). As a result, the combined STDW model with the phosphate patch R310 appears to give the best performance, outperforming several scoring functions. The presence of water molecules in the preparation and optimization of the protein structure allows the minimized structure to deviate less from the experimental structure.

2. Four popular clustering algorithms were significantly optimized to enable their application at distinct phases of the Fragment-Based Drug Design. Through binary storage, binary translation of the primary operations, or complete reformulation of the algorithms, exact (BitClust, DP+) and modified (BitQT, RCDPeaks, MD-SCAN) versions of the original proposal were presented and thoroughly benchmarked. A methodological confusion between Quality Threshold and Daura's algorithms was exposed to the community.

3. An efficient computational linker for assembling Chemically Modified Oligonucleotides (fragments) onto oligochains (lead compounds) was developed. Our NUCLEotide AssembleR was able to return clash-free sequences following distinct constraints (in sequence or exploration region) and could identify (despite inherent limitations) several experimental binding modes in three case studies. Another necessary functionality of this software is the determination of hotspots at the target's surfaces for guiding the Fragment-Based Drug Design.

4. We designed an *in silico* fragment-based workflow that produces low-energy binding modes of Chemically Modified Oligonucleotides (obtained by the NUCLEotide AssembleR after Multiple-Copy Simultaneous Search docking) evincing structural selectivity against β-site Amyloid Precursor Protein Cleaving Enzyme 1 enzyme over the related β-site Amyloid Precursor Protein Cleaving Enzyme 2. The top-ranked $\mathrm{BACE1}$ binding modes are able to linearly traverse the active site region or concurrently interact with the top side of the flap and the active site, while similar binding modes are not detected for $\mathrm{BACE2}$.

8 - PERSPECTIVES

1. Integrating optimized clustering algorithms into the NUCLEAR software could enhance performance and scalability. Other approaches could be explored as alternatives to the current BitClust-inspired implementation to reduce computational complexity for extensive fragment collections.
2. Calculating physicochemical and drug-likeness molecular descriptors could enable rational filtering of non-viable oligonucleotide sequences identified by NUCLEAR. Descriptors related to solubility, cell permeability, and synthetic accessibility could be computed to screen for promising sequences worth experimental validation.
3. Performing MD simulations on the NUCLEAR-generated complexes would allow evaluating their geometrical stability and dynamics. These simulations could identify unstable or unfavorable interactions that may preclude function, informing on oligonucleotide sequence optimization and guiding synthetic efforts toward productive candidates. Longer timescale simulations may also reveal conformational changes relevant to the inhibition mechanism. Beyond geometrical stability, simulations could also probe oligonucleotide binding kinetics and affinity with receptors to establish viable complexes. Free energy calculations could quantify the relative strengths of binding.
4. Collaborations with synthetic chemists will be crucial to experimentally produce and characterize oligonucleotide candidates proposed by NUCLEAR. These validations will demonstrate real-world utility.

9 - ANNEXES

9.1 . MCSS-based predictions of binding and selectivity of nucleotides

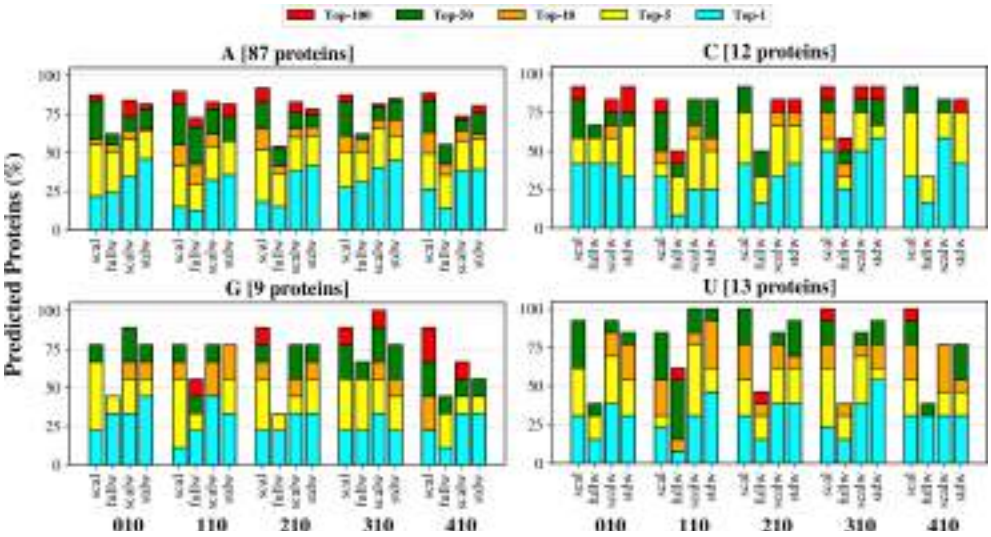


Figure 9.1: *Decomposition of docking powers per nucleotide type. The data are shown for the clustered distribution and each Top-i.*

9.2 . Reinventing the wheel of molecular clustering

9.2.1 . Details of MD used in benchmarks

9.2.1.1 6 kF

All details on generation of the 6 kF trajectory has been previously published by Shea and Levine[256].

9.2.1.2 30 kF

All details on generation of the 30 kF trajectory has been previously published by Melvin *et al.*[242].

9.2.1.3 50 kF

The initial coordinates of the repetitive unit of serotype 18C of *Streptococcus pneumoniae* were obtained with CarbBuilder[274]. The system with dimensions 32 $\times$ 32 $\times$ 43 Å contains 1202 water molecules and was neutralized using Na^+ and Cl^- ions at near-physiological

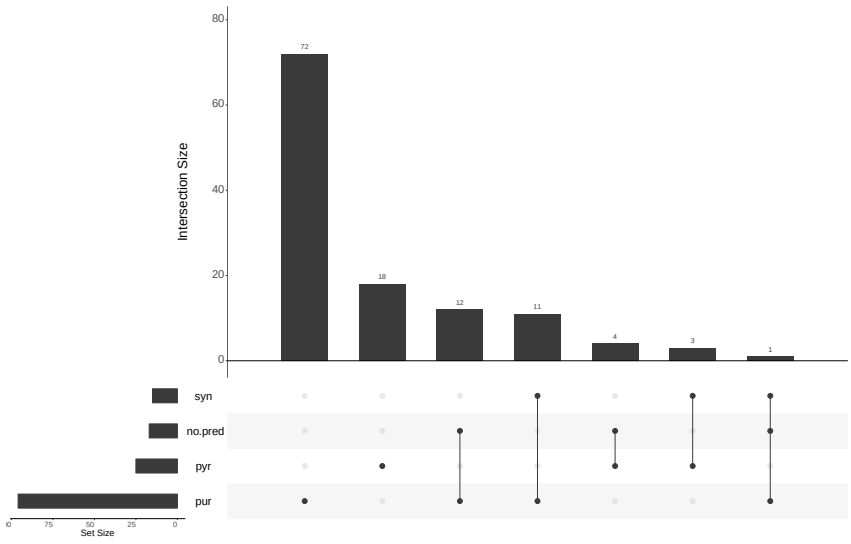


Figure 9.2: *Upset diagram of the impact of the conformational features on the Top-10 predictions. The intersections with only one member are not shown; syn: syn conformation of the nucleic acid base; pyr: pyrimidine; pur: purine.*

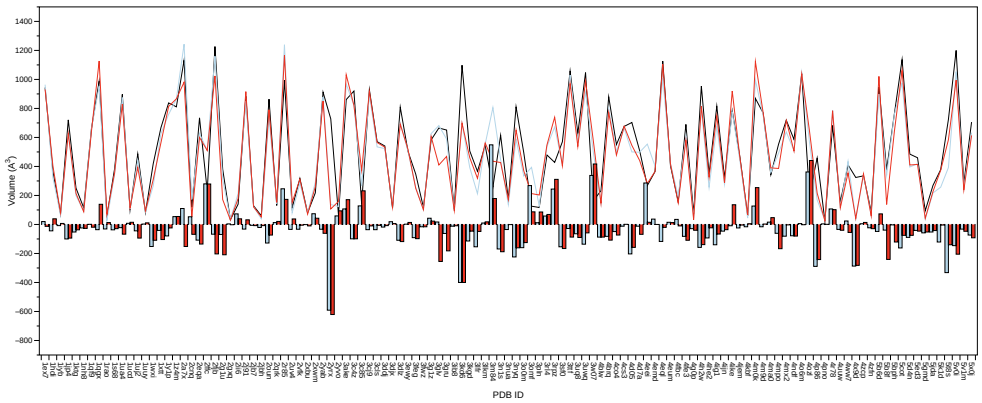


Figure 9.3: *Variations in the volume of the binding site. Black line: experimental structure; Blue line: optimized structure for the SCAL model; Red line: optimized structure for the STDW model. The histograms indicate a decreasing of the volume for the negative values and an increasing for the positive values. The calculation of volume does not take into account the water molecules.*

concentration (150 mM). The solvating water molecules were described with the TIP3P model ([275]).

The MD trajectory was computed with NAMD (v2.12) program[276], using the CHARMM36 force field[277,278] and periodic boundary conditions (PBC). Long-range interactions for the full system with PBC were handled by means of the particle-mesh Ewald method[279] with a grid resolution of less than 1 Å, while all other non-bonded interactions were computed with a cutoff of 12 Å. SHAKE constrains[280] were applied to water molecules bonds. A time step of 1 fs was used, saving all frames every 1 ps to obtain 50000 conformations

Features		Freq. Benchmark	Freq. good
binding site	nwat.low	62	60
	vol.low	69	70
	others	12	10
	metals	36	60
conformational	syn	12	0
	pur	79	80
	pyr	21	0
interaction	no.base.contacts	12	30
	no.salt.bridges	44	30
	no.stacking	49	70
	clash aa	22	20
	clash w	33	40

Table 9.1: *Frequencies of occurrences for molecular features in the Top-10 for non-optimal (good) predictions. Others: presence of additional nucleotidic (nucleic acid) fragment in the binding site; metals: presence of metal(s) in the binding site; nwat.low: presence of number of water molecules below the threshold value; vol.low: volume of the binding site below the threshold value; syn: syn conformation of the nucleic acid base; pyr: pyrimidine; pur: purine; no.base.contacts: absence of contacts with the nucleic acid base; clash_aa: clash(es) with amino-acid residues; clash_w: clash(es) with water molecules; no.salt.bridges: absence of salt-bridge; no.stacking: absence of stacking.*

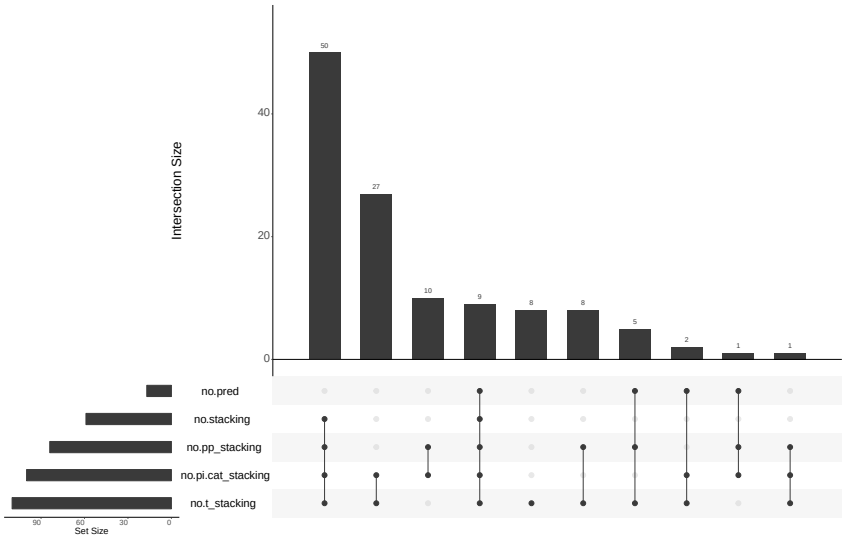


Figure 9.4: *Upset diagram of stacking contributions for the Top-10 predictions. no.pp_satcking: no π-π stacking; no.pi.cat_stacking: no π-cation stacking; no.t_stacking: no t stacking.*

contained in the 50 kF (kF=1000 Frames) trajectory. The production MD simulation was performed at a constant temperature of 300 K and a pressure of 1 bar (NPT ensemble) by using the Nosé-Hoover thermostat and barostat[281].

	stdw-R110	scal-R310	scal-R110
1rao	Y		
1wxi	Y		
1xtt	Y		
2g1u	Y		
2xbu	Y		
2xwm	N	Y	
3gru	N	N	N
3m84	Y		
3nua	N	N	Y
3omf	Y		
3sfo	N	Y	
4eei	N	Y	
4ijn	N	Y	
4zfn	Y		
5ed3	Y		
5jda	N	Y	
5voi	N	N	N

Table 9.2: *Impact of the nonbonded model and phosphate patch on the recovery effect of the Top-10 no-prediction subset. Y: recovered prediction using a different model and patch; N: no recovered prediction with the given model and patch.*

Volumes		Freq. Benchmark	Freq. nopred.
SCAL	UP	12	0
	DOWN	19	18
STDW	UP	13	0
	DOWN	21	35

Table 9.3: *Variations in the binding site's volume for the subset of protein-nucleotides complexes with no prediction in the Top-10. The volume of reference corresponds to that of the experimental structure; the modified volumes are calculated for both the SCAL and STDW models. Only the cases where the variation equals or exceeds 100Å^3 are considered. UP: increase of the binding site's volume. DOWN: decrease of the binding site's volume.*

9.2.1.4 100A kF

A structural model of Cyclophilin A (PDB ID 2N0T[282]) was embedded into an octahedral box. The system with dimensions 89 × 89 × 95 Å contains 22083 water molecules and was neutralized at near-physiological concentration (150 mM), using Na^+ and Cl^- ions. The water molecules were described with the TIP3P model[275]. At least 15 Å of space was left between the Cyclophilin structure and the simulation cell boundaries, keeping more than 50 Å of distance between protein copies of neighboring cells during MD runs.

The MD trajectory was computed with NAMD (v2.13) program[276], using the CHARMM36

Features		Freq. Benchmark	Freq. STDW(R310)
binding site	nwat.low	62	51
	vol.low	69	72
	others	12	6
	metals	36	30
conformational	syn	12	17
	pur	79	83
	pyr	21	17
interaction	no.base.contacts	12	11
	no.salt.bridges	44	62
	no.stacking	49	49
	clash aa	22	21
	clash w	33	40

Table 9.4: *Frequencies of occurrences for molecular features in the Top-10 for non-predicted cases of STDW-310 versus benchmark. Others: presence of additional nucleotidic (nucleic acid) fragment in the binding site; metals: presence of metal(s) in the binding site; nwat.low: presence of number of water molecules below the threshold value; vol.low: volume of the binding site below the threshold value; syn: syn conformation of the nucleic acid base; pyr: pyrimidine; pur: purine; no.base.contacts: absence of contacts with the nucleic acid base; clash_aa: clash(es) with amino-acid residues; clash_w: clash(es) with water molecules; no.salt.bridges: absence of salt-bridge; no.stacking: absence of stacking.*

force field[277,278] and periodic boundary conditions (PBC). Long-range interactions for the full system with PBC were handled by means of the particle-mesh Ewald method[279] with a grid resolution of less than 1 Å, while all other non-bonded interactions were computed with a cutoff of 12 Å. SHAKE constrains[280] were applied to water molecules bonds. A time step of 1.0 fs was used, saving all frames every 1 ps to obtain 100000 conformations contained in the 100A kF trajectory. The production MD simulation was performed at a constant temperature of 310 K and a pressure of 1.0 bar (NPT ensemble) by using the Nosé-Hoover thermostat and barostat[281].

9.2.1.5 100B kF

The 2.2 Å X-ray bovine-rhodopsin structure (Protein Data Bank code: 1U19)1,2 was embedded inside a palmitoyl-oleoyl-phosphatidylcholine (POPC) hydrated membrane. The protonation states of the titratable amino-acid residues were assigned according to the protocol described by[283]. The internal water molecules were conserved in the rhodopsin model. The rhodopsin/POPC/water system was neutralized at near-physiological concentration (150 mM) using Na^+ and Cl^- ions, where at least 15.0 Å of space was left between the rhodopsin structure and cell boundaries. The TIP3P model[275] was used for all water molecules. The two palmitoyl residues covalently linked to the residues Cys322 and Cys323 were kept in the three-dimensional structure, as well as the Cys110-Cys187 disulfide bridge.

Table 9.5: *Protein-nucleotide benchmark composition.*

PDB-ID	Nuc-ID	Resolution	Classification
1ex7	5GP	1.90	TRANSFERASE
1hdi	AMP	1.80	TRANSFERASE
1iyb	5GP	1.50	HYDROLASE
1jp4	AMP	1.69	HYDROLASE
1ktg	AMP	1.80	HYDROLASE
1nh8	AMP	1.80	TRANSFERASE
1qf9	C5P	1.70	TRANSFERASE
1qgx	AMP	1.60	HYDROLASE
1rao	AMP	1.56	TRANSFERASE
1s68	AMP	1.90	LIGASE
1ua4	AMP	1.90	TRANSFERASE
1ucd	U5P	1.30	HYDROLASE
1uj2	C5P	1.80	TRANSFERASE
1uuy	AMP	1.45	CHELATASE
1wxi	AMP	1.70	LIGASE
1xtt	U5P	1.80	TRANSFERASE
1y1p	AMP	1.60	OXIDOREDUCTASE
1z4m	U5P	1.70	HYDROLASE
2a7x	AMP	1.70	LIGASE
2cnq	AMP	1.00	LIGASE
2eqa	AMP	1.80	NA BINDING PROTEIN
2ffc	U5P	1.70	LYASE
2fjb	AMP	1.70	OXIDOREDUCTASE
2g1u	AMP	1.50	MEMBRANE PROTEIN
2gxq	AMP	1.20	HYDROLASE
2ii6	C5P	1.75	TRANSFERASE
2j91	AMP	1.80	LYASE
2jb7	AMP	1.65	UNKNOWN FUNCTION
2jbh	5GP	1.70	TRANSFERASE
2oun	AMP	1.56	HYDROLASE
2qrk	AMP	1.75	OXIDOREDUCTASE
2r85	AMP	1.70	UNKNOWN FUNCTION
2uv4	AMP	1.33	TRANSFERASE
2vfk	AMP	1.50	HYDROLASE
2xbu	5GP	1.80	TRANSFERASE
2xwm	C5P	1.80	TRANSFERASE
2yab	AMP	1.90	TRANSFERASE
2yrx	AMP	1.90	LIGASE
2yvo	AMP	1.67	HYDROLASE
3ake	C5P	1.50	TRANSFERASE
3c4z	AMP	1.84	TRANSFERASE
3c85	AMP	1.90	MEMBRANE PROTEIN
3cj9	AMP	1.80	HYDROLASE
3cls	AMP	1.65	ELECTRON TRANSPORT
3ddj	AMP	1.80	UNKNOWN FUNCTION
3djx	C5P	1.69	HYDROLASE
3dlz	AMP	1.85	TRANSFERASE
3ewy	U5P	1.10	LYASE
3feg	AMP	1.30	TRANSFERASE
3fwz	AMP	1.79	MEMBRANE PROTEIN
3g1z	AMP	1.95	LIGASE
3glv	AMP	1.99	BIOSYNTHETIC PROTEIN
3gru	AMP	1.60	TRANSFERASE
3ib8	AMP	1.80	HYDROLASE
3kd6	AMP	1.88	TRANSFERASE
3kgd	AMP	1.68	LIGASE
3lfr	AMP	1.53	MEMBRANE PROTEIN
3lkm	AMP	1.60	TRANSFERASE
3m84	AMP	1.70	LIGASE
3n1s	5GP	1.45	HYDROLASE
3nua	AMP	1.40	LIGASE
3nyq	AMP	1.43	LIGASE
3oom	AMP	1.90	HYDROLASE
3omf	AMP	1.80	METAL BINDING PROTEIN
3pln	U5P	1.50	OXIDOREDUCTASE
3rl4	5GP	1.29	HYDROLASE
3rpz	AMP	1.51	LYASE
3sfo	AMP	1.35	HYDROLASE
3ttf	AMP	1.92	TRANSFERASE
3uq8	AMP	1.70	LIGASE
3uwq	U5P	1.80	LYASE
3w07	U5P	1.03	LYASE
4blw	AMP	1.95	TRANSFERASE
4brq	AMP	1.45	HYDROLASE
4co4	AMP	1.50	SIGNALING PROTEIN
4cs3	AMP	1.50	LIGASE
4do5	AMP	1.65	LIGASE
4d7a	AMP	1.80	LIGASE
4eei	AMP	1.92	LYASE
4emd	C5P	1.75	TRANSFERASE
4eql	AMP	1.80	LIGASE
4eum	AMP	1.80	TRANSFERASE
4fbc	AMP	1.70	HYDROLASE
4fe3	U5P	1.74	HYDROLASE
4gop	U5P	1.80	GENE REGULATION
4h2w	AMP	1.95	LIGASE
4he2	AMP	1.60	HYDROLASE
4ig1	AMP	1.43	HYDROLASE
4ijn	AMP	1.70	TRANSFERASE
4ike	AMP	1.48	TRANSFERASE
4jem	C5P	1.55	HYDROLASE
4kbf	AMP	1.90	HYDROLASE
4mok	AMP	1.40	TRANSFERASE
4m9d	AMP	1.82	LIGASE
4mao	AMP	1.98	LYASE
4mpo	AMP	1.90	HYDROLASE
4mx2	AMP	1.90	LYASE
4ndf	AMP	1.94	NA BINDING PROTEIN
4o6m	C5P	1.90	TRANSFERASE
4ozl	AMP	1.49	SIGNALING PROTEIN
4p86	5GP	1.93	TRANSFERASE
4pno	U5P	0.97	NA BINDING PROTEIN
4r78	AMP	1.45	TRANSFERASE
4uuw	AMP	1.98	BIOSYNTHETIC PROTEIN
4ww7	AMP	1.67	TRANSFERASE
4x9d	U5P	1.50	NA BINDING PROTEIN
4zcp	C5P	1.98	TRANSFERASE
4zfn	5GP	1.90	TRANSFERASE
5b6d	C5P	1.65	TRANSFERASE
5b8f	C5P	1.45	LYASE
5bph	AMP	1.70	LIGASE
5cot	AMP	1.69	LIGASE
5d4n	AMP	1.60	SIGNALING PROTEIN
5ed3	AMP	1.31	HYDROLASE
5gmd	AMP	1.50	LYASE
5jda	AMP	1.40	TRANSFERASE
5k1d	5GP	1.94	HYDROLASE
5t8s	AMP	1.70	TRANSFERASE
5voi	AMP	1.90	LIGASE
5v1m	U5P	1.47	HYDROLASE
5xoj	AMP	1.43	TRANSFERASE

A 100-ns MD trajectory was computed with the NAMD (v2.12)[276] program, using the CHARMM36 force field[277,278]. The system was simulated at the constant temperature of 310 K and pressure of 1.0 bar by using the Nosé-Hoover thermostat and barostat[281]. A time-step of 1.0 fs was used for the production 100-ns run, and the frames were saved every 1 ps, which allowed to obtain a MD trajectory containing 100000 frames. A cutoff of

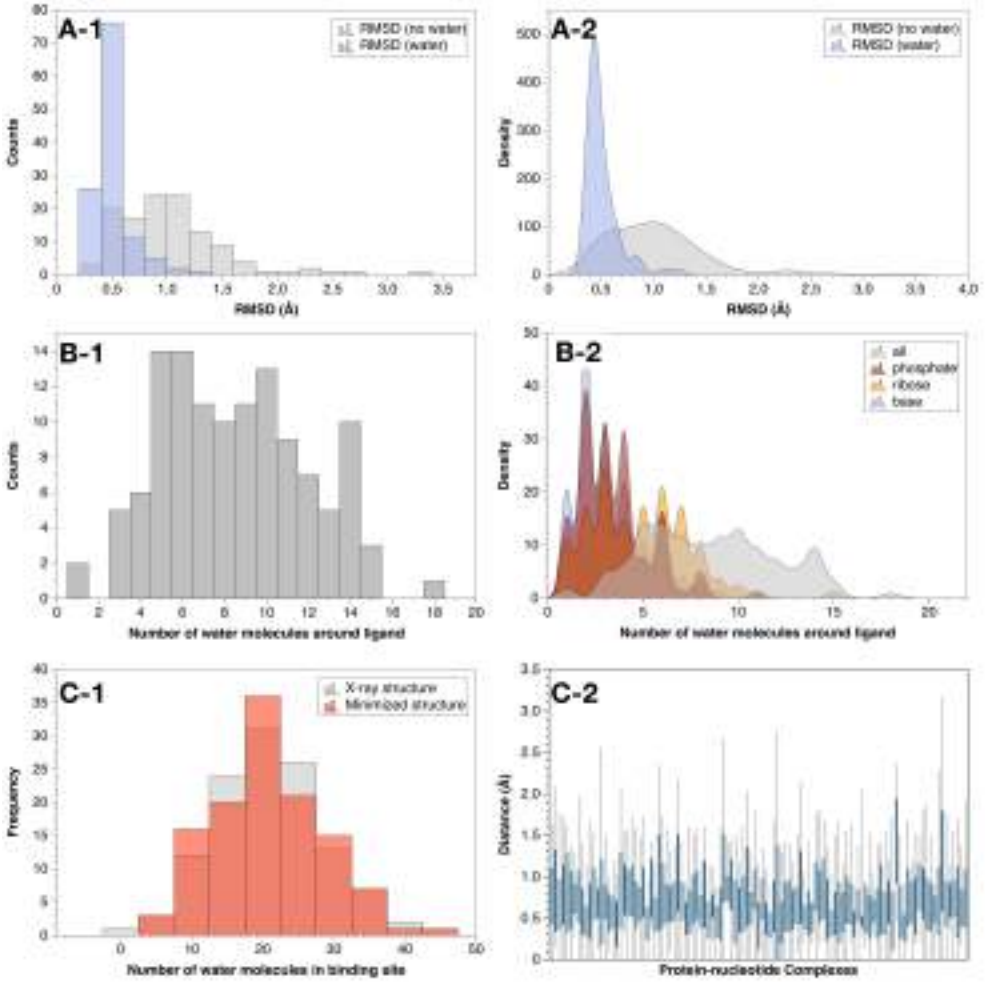


Figure 9.5: *Distributions of water molecules and impact on the binding sites. A-1.: Histogram of RMSD in presence/absence of water molecules; A-2.: Same as A-1 with a smooth histogram; B-1.: the number of water molecules around the ligand (distance cutoff of 4.0Å); B-2.: Same as B-1 with decomposition per nucleotide moiety; C-1.: Number of water molecules within the binding site as defined in MCSS by the box parameters (see Section 2.1.3); C-2.: displacements (Å) of water molecules from their crystallized positions.*

12 Å was used to compute the other non-bonded interactions. The system was simulated using periodic boundary conditions (PBC), which were considered in the calculations of long-range interactions for the full system, using the particle-mesh Ewald method[279] with a grid resolution less than 1.0 Å.

A first minimization-equilibration cycle was performed while both all rhodopsin atoms and internal water molecules were fixed, and the solvating water molecules were relaxed by a 10000-step conjugated gradient (CG) minimization, followed by 4-ns NVT MD simulation. In the second minimization-equilibration cycle, the protein backbone atoms were fixed and the system was again subjected to a 10000-step CG minimization, followed by 4-ns NPT simulation. Subsequently, a third 8-ns NPT equilibration cycle was performed

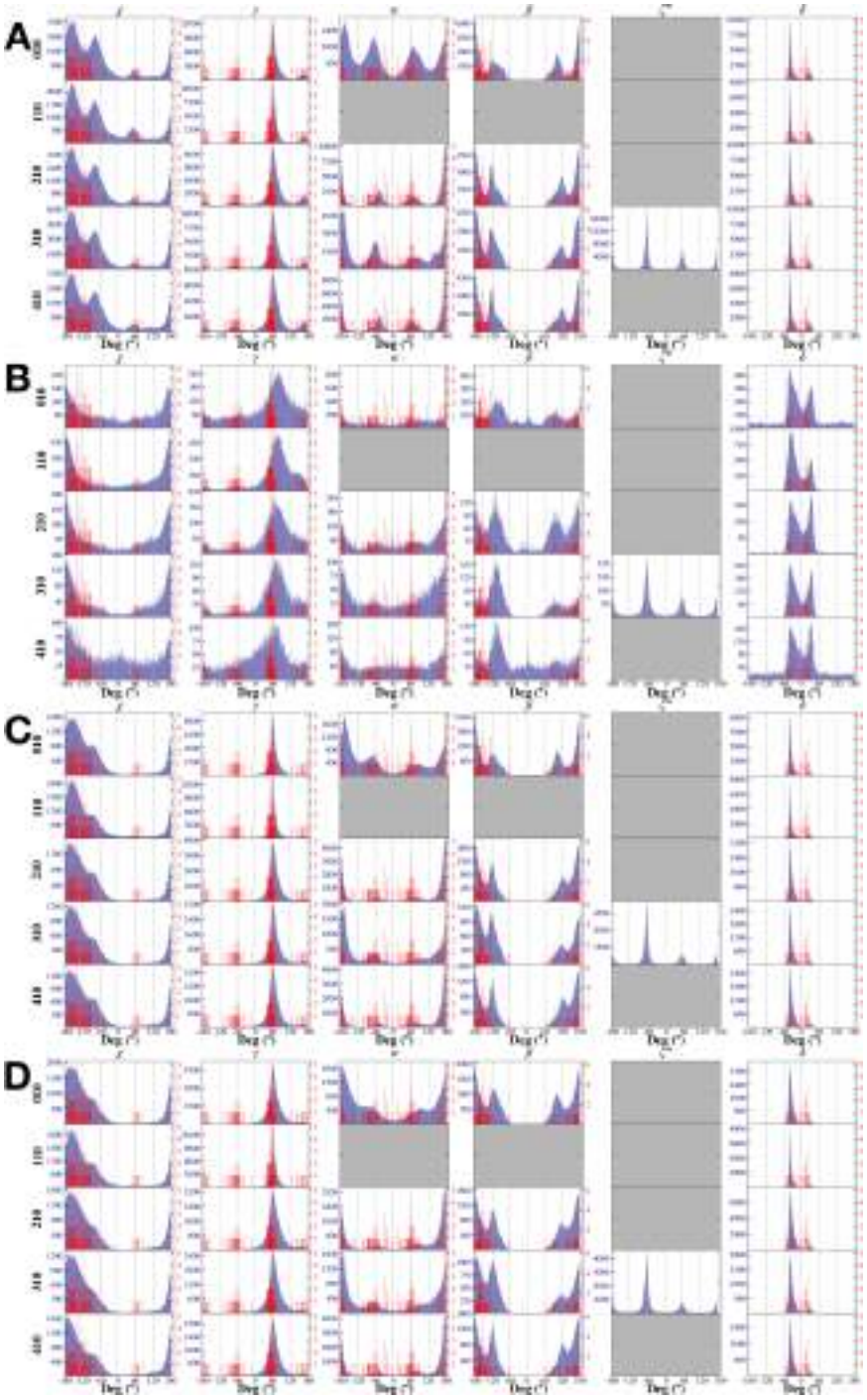


Figure 9.6: *Torsion angles. Non-bonded models and associated patches (R010 to R410): A. SCAL, B. FULLW, C. SCALW, D. STDW. In blue, the distribution of the torsion angles observed in the MCSS minima. In red, the distribution of the torsion angles observed in the bound ligands.*

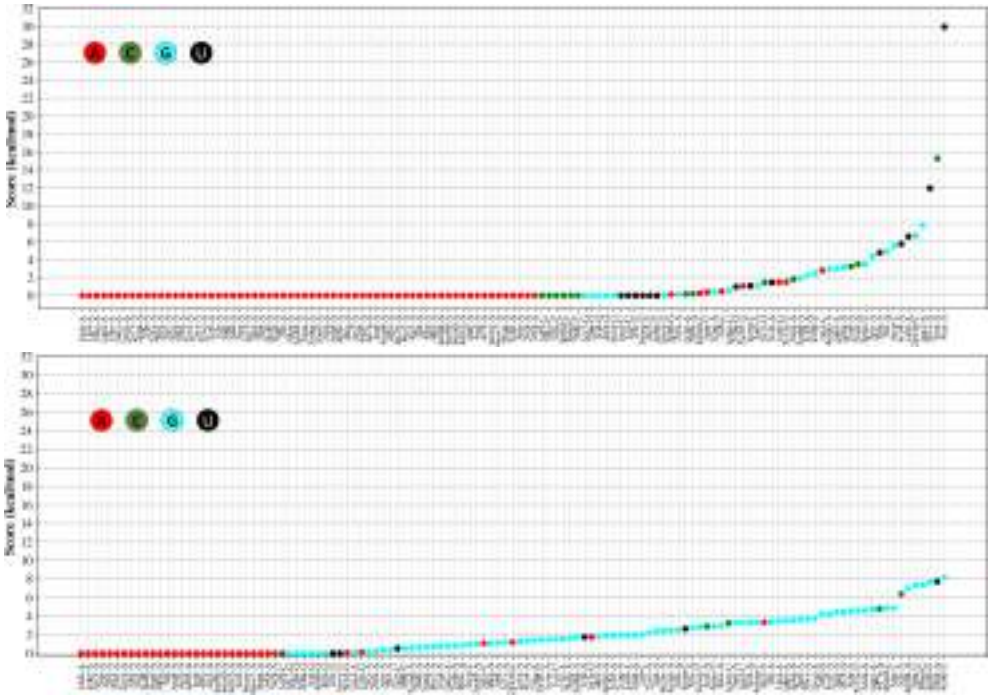


Figure 9.7: *Scoring differences (offset) between the best-ranked pose whatever the nucleotide type and the best-ranked pose for the nucleotide corresponding to the native ligand. Top: STDW model; bottom: SCAL model. The color code indicates the nucleotide type.*

with the protein free to move. Finally, a production 100-ns NPT run was performed without applying any restraint to obtain the 100K trajectory.

9.2.1.6 250 kF

The coordinates of the PHF8 tetramer were derived from the biological assembly of the Tau peptide (PDB ID 2ON9). All the MD settings were defined using the CHARMM-GUI interface[249,284,285] and the CHARMM36m force-field[286]. An octahedral water box was defined using a 10 Å edge distance. Na^+ and Cl^- ions were added using a physiological concentration (150 mM) and assigned via the Monte-Carlo protocol.

The MD simulation were carried through NAMD (v2.12)[287] using the TIP3P water model at 298.15K. The constant temperature was maintained using Langevin dynamics. All bonds involving hydrogen atoms were constrained with the SHAKE algorithm with a time step of 2 fs. The minimized structures of the tetramers were subjected to an equilibration of 1 ns in the NVT ensemble. The production was carried out in the NPT ensemble. The pressure of 1 atm was maintained with the Nosé-Hoover-Langevin piston algorithm. Electrostatic interactions were computed with the smooth particle-mesh Ewald sum[279] ; the real-space part of the Ewald sum and the Lennard-Jones interactions were truncated using a smooth switching function between 10 and 12 Å.

Features		Freq. Benchmark	Freq. no.pred
binding site	nwat.low	62	59
	vol.low	69	82
	others	12	6
	metals	36	24
conformational	syn	12	0
	pur	79	71
	pyr	21	23
interaction	no.base.contacts	12	12
	no.salt.bridges	44	59
	no.stacking	49	53
	clash aa	22	18
	clash w	33	41

Table 9.6: *Frequencies of occurrences for molecular features in the Top-10 non-predicted cases versus benchmark. Others: presence of additional nucleotidic (nucleic acid) fragment in the binding site; metals: presence of metal(s) in the binding site; nwat.low: presence of number of water molecules below the threshold value; vol.low: volume of the binding site below the threshold value; syn: syn conformation of the nucleic acid base; pyr: pyrimidine; pur: purine; no.base.contacts: absence of contacts with the nucleic acid base; clash_aa: clash(es) with amino-acid residues; clash_w: clash(es) with water molecules; no.salt.bridges: absence of salt-bridge; no.stacking: absence of stacking.*

9.2.1.7 500 kF

To generate the 500 kF trajectory, 100000 frames randomly combined 6 kF were concatenated five times.

9.2.1.8 1 MF

The atomic coordinates of a 1.8 Å resolution crystal structure of ubiquitin (PDB ID 1UBQ). The system was inserted in an octahedral box with initial dimensions 70 $\times$ 72 $\times$ 76 Å, containing 11815 water molecules described with the TIP3P[275] model. The system was neutralized at near-physiological concentration (150 mM) using Na^+ and Cl^- ions. At least 50 Å of distance was left between protein copies of neighboring cells during molecular dynamics (MD) runs, keeping more than 15.0 Å of space between the ubiquitin structural model and simulation cell boundaries.

The MD simulation was performed with NAMD (v2.13)[288,289], applying the CHARMM36[277,278] force field and SHAKE[280] constrains to water molecules bonds. Long-range interactions were calculated for the full system with periodic boundary conditions (PBC), using the particle-mesh Ewald method[279] and a grid resolution of less than 1.0 Å, as well as, a cutoff of 12 Å to compute all other non-bonded interactions. A 100 ns MD run was carried out at a constant temperature of 310 K and pressure of 1.0 atm by using the Nosé-Hoover thermostat and barostat[281] while a time-step of 1.0 fs was applied, saving all frames every

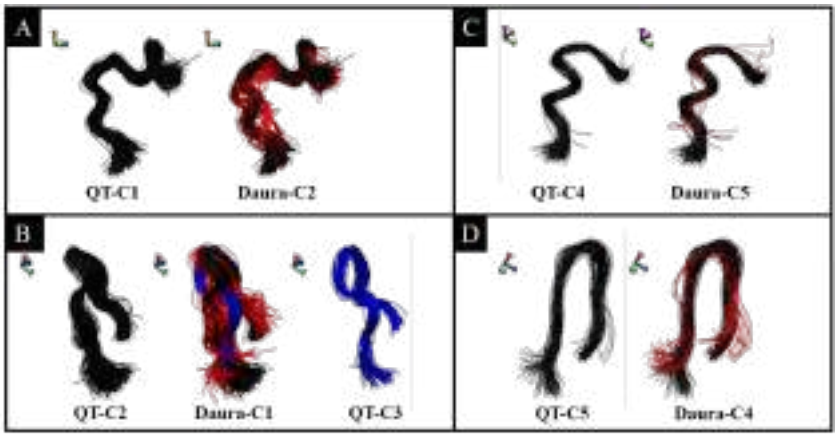


Figure 9.8: *Graphical ribbons representation of 6kF trajectory (backbone) for the first five clusters retrieved by* QT *and their related counterpart determined by Daura algorithm.*

0.1 ps to obtain 1000000 conformations.

9.2.2 . Reports inaccurately claiming to perform QT clustering

On a careful analysis of 30 scientific works claiming to use QT to retrieve clusters from MD trajectories, we identify two common pitfalls (labeled as P1 and P2) that appear in the current literature. In the first group of works (P1)[290–299], the authors explicitly describe the algorithm they were using as if it were QT, citing the original paper of Heyer *et al.*[214]. Only one of them correctly described the algorithm as Daura clustering[296] but still cited the incorrect reference, Heyer *et al.*[214].

In the second group of works (P2)[300–319], the authors do not thoroughly describe the algorithm used but do affirm performing QT when they execute Daura clustering. While both groups of authors might have wanted to select QT because of the high uniformity of returned clusters, those in P1, which explicitly describe the foundations of QT, are more affected because part of their research was built based on incorrect assumptions.

In Figure 9.8, a relationship is established between clusters retrieved by QT and Daura algorithm. Graphical representation corresponds to ribbons of the 6 kF trajectory's backbone. The numbering of clusters (CX) is consistent with that in Figure 4.1. Note that both algorithms return clusters in decreasing order of size, but this does not imply that conformations contained in equally numbered clusters are the same. Figure 9.8A shows that all elements of the first cluster returned by QT (QT-C1) are contained in the second cluster reported by Daura (Daura-C2). As it is easy to note, Daura-C2 contains, in addition, many elements (in red) that do not fit well in the uniform conformation depicted by QT-C1 (black).

The first cluster retrieved by Daura (Daura-C1, Figure 9.8B) is a critical example of the algorithm selection implications and the risks of confusing both. Retrieved conformations by Daura-C1 are dispersed. It groups two QT clusters, QT-C2 (black) and QT-C3 (blue), representing highly correlated sets of distinct conformations. It is worth noting that Daura-C1 could be erroneously qualified as the most representative cluster of the

simulation, given that it is the most populated one.

The fourth and fifth clusters retrieved by QT (QT-C4 and QT-C5, respectively) also present a uniform conformation and are entirely contained at Daura-C5 and Daura-C4, respectively (Figure 9.8C and D). Once again, clusters retrieved by Daura algorithm contain elements that deviate from the uniform pattern that QT clusters exhibit. Marked dispersion of groups returned by Daura method could alter those analyses based on intra-cluster average properties.

9.2.3 . DP+ pseudocode

Algorithm 5: Get ρ and η for a particular node i

function get_node_info($i, k, d_c, trajectory$)
    $i\_vector =$ ***calc_rmsd_vector***$(i, trajectory)$
    $i\_rho =$ ***count_elements***$(i\_vector < d_c)$
    $i\_partition =$ ***partial_sort_elements***(i_vector, k)
    $i\_eta\_elements =$ ***sort_elements***$(i\_partition[0:k])$
    $i\_eta\_rmsd = i\_vector[i\_eta\_elements]$
    $i\_eta =$ ***join***$(i\_eta\_elements, i\_eta\_rmsd)$
    return (i_rho, i, i_eta)

Algorithm 6: Compute the Oriented Tree of an MD trajectory

function compute_oriented_tree($k, d_c, trajectory$)
► **1. Initialize containers**
  $elements = \{1, 2, 3, ..., trajectory.size\}$
  $main\_heap =$ ***create_heap***()
  $auxiliary\_heap =$ ***create_heap***()
  $rho\_info = \{\}$
  $delta\_info = \{\}$
  $nearest\_neighbors = \{\}$
► **2. Find node i whose neighborhood will be analyzed**
  while $True$ **do**
    if $main\_heap \neq \emptyset$ **then**
      $i, i\_rho, i\_eta =$ ***pop_first_from***$(main\_heap)$
    else if $elements \neq \emptyset$ **then**
      $i =$ ***pop_any_from***$(elements)$
      $i\_rho, i, i\_eta =$ ***get_node_info***$(i, k, d_c, trajectory)$
    else
      break
► 3. **Try to find j inside η_i**
    while $True$ **do**
      if $i\_eta \neq \emptyset$ **then**
        $j, rmsd\_ij =$ ***next***(i_eta)
      else
        send$((i\_rho, i), auxiliary\_heap)$
        break
      if $j \in elements$ **then**
        $j\_rho, j, j\_eta =$ ***get_node_info***$(j, k, d_c, trajectory)$
        send$((j\_rho, j, j\_eta), main\_heap)$
        $rho\_info[j] = j\_rho$
        remove_from$(elements, j)$
      else
        $j\_rho = rho\_info[j]$
      if $j\_rho > i\_rho$ **then**
        $nearest\_neighbors[i] = j$
        $delta\_info[i] = rmsd\_ij$
        break
► **4. Processing the auxiliary heap**
  while $True$ **do**
    if $auxiliary\_heap \neq \emptyset$ **then**
      $i\_rho, i =$ ***pop_first_from***$(auxiliary\_heap)$
      $i\_vector =$ ***calc_rmsd_vector***$(i, trajectory)$
      $denser\_j =$ ***get_elements***$(rho\_info > i\_rho)$
      $j\_vector = i\_vector[denser\_j]$
      if $j\_vector \neq \emptyset$ **then**
        $j =$ ***get_min_element***(j_vector)
        $delta\_info[i] = i\_vector[j]$
        $nearest\_neighbors[i] = j$
      else
        $delta\_info[i] =$ ***get_max_value***$[i\_vector]$
        $nearest\_neighbors[i] = i$
    else
      break
return $(delta\_info, rho\_info, edges)$

9.2.4 . RCDPeaks refinements over DP

9.2.4.1 Automatic detection and screening of cluster centers

In the original DP, users must select the cluster centers from the decision graph before the DP algorithm can assign the remaining elements to each cluster (Figure 9.9A). This selection introduces a potentially biased, user-dependent step that prevents automatic runs. Several authors have used statistical mechanisms to bypass this step (see reference[228] for a review) by detecting clusters centers as ρ, δ or γ outliers (Equation 9.1).

$$\gamma_i = \rho_i * \delta_i \tag{9.1}$$

The gap-based centers selection method proposed by Flores and Garza[228] proceeds as follows: First, a subset P_1 containing elements whose ρ and δ values are higher than the average defined (discontinue lines in Figure 9.9B). P_1 is sorted in descending order of each γ_i score. The consecutive point distance (Equation 9.2) between all candidates, as well as the average point distance (Equation 9.3) are then computed. In this context, a gap is formally defined as a $d_i \geq \bar{d}$. The last gap in P_1 (formed by elements i and $i+1$) is considered a threshold and all elements before i are marked as cluster centers.

$$d_i = abs(\gamma_i - \gamma_{i+1}) \tag{9.2}$$

$$\bar{d} = \sum_{i \in P_i} \frac{d_i}{|P_i|} \tag{9.3}$$

The described methodology produced many cluster centers for the trajectories analyzed in this work. Instead of stopping the algorithm after the first loop, RCDPeaks makes another iteration on a new subset P_2, containing only elements whose ρ and δ values are higher than the average in P_1 (Figure 9.9C). This procedure effectively reduces the number of candidate clusters, which are intuitively a subset of the original P_1. Iteration continues until the one-member set P_n is found (Figure 9.9D). All sets from P_1 to P_n may be considered valid automatic guesses of cluster centers. Each of the P_n guesses made by RCDPeaks will be further processed in n distinct clustering jobs of the same oriented tree represented by the decision graph in Figure 9.9A. In the analyzed trajectories, n varies from 2 to 3.

Although RCDPeaks implements the Flores and Garza method, users still have the choice to set ρ and δ values manually. Also, as the most time-consuming part of RCDPeaks consists of computing those two magnitudes for each element, the software saves the decision graph. This inexpensive saving allows users to experiment independently with the result of different ρ and δ cutoffs for cases where the automatic guesses do not perform as expected.

Centers retrieved by either an automatic or a manual selection may lie within a d_c radius. The original DP unsuccessfully handles these cases by considering all centers as good choices. RCDPeaks avoids taking into account similar centers through a screening

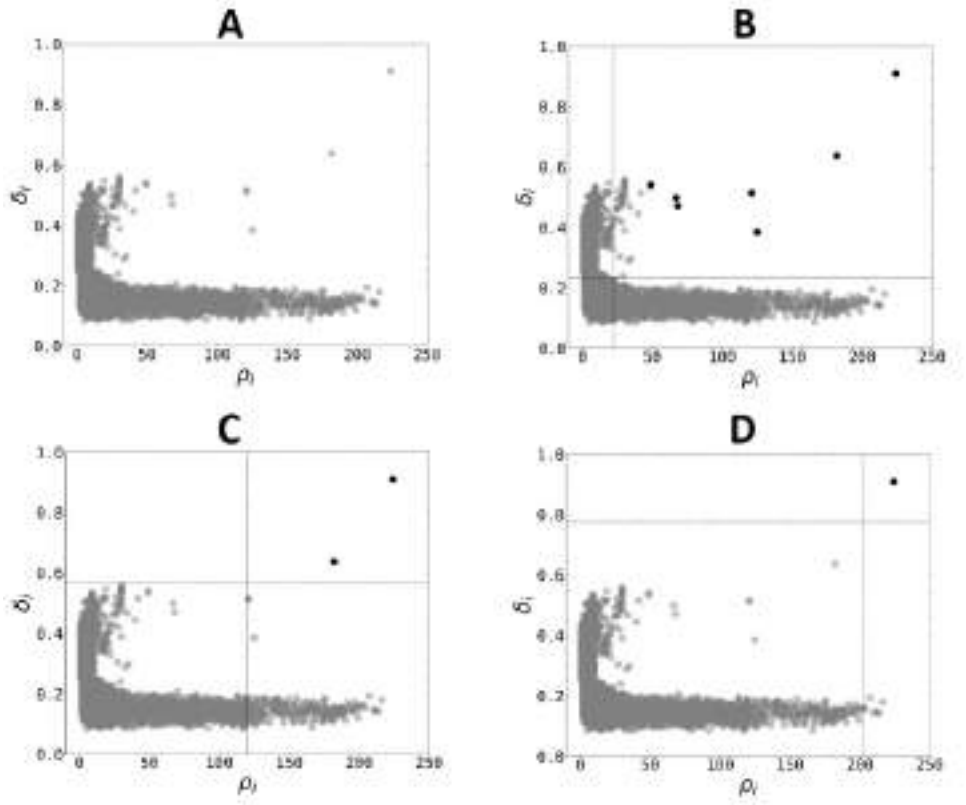


Figure 9.9: *Iterative gap-based method of Flores and Garza implemented in RCDPeaks for the automatic detection of cluster centers. A-) Decision graph. B-D-) Consecutive iterations of the method produce several automatic guesses of cluster centers.*

process that iteratively takes the center of highest γ_i from P_i as a reference and removes other centers within a d_c distance from further consideration.

9.2.4.2 Clusters core refining

MD clusters generated by the original version of DP usually contain structurally unrelated elements. Definitions of the core and halo zones (see Section 1.6.2.3) contribute to some extent to the separation of highly similar elements (core) from more loosely related ones (halo). However, the original cores obtained by the DP clustering may still display a high level of dissimilarity, as can be appreciated in Figure 9.10.

Since cluster centers have a preponderant significance in DP, it is reasonable to expect their geometrical resemblance to elements in their respective cores. RCDPeaks follows a simple procedure to extract a set of exemplar elements (a refined core), evincing a higher degree of collective similarity than what can be obtained from the original definition of core zones in DP. For each cluster C_i, its refined core will consist of those elements within a d_c distance from its cluster center. As shown in Figure 9.10C, this restrained set

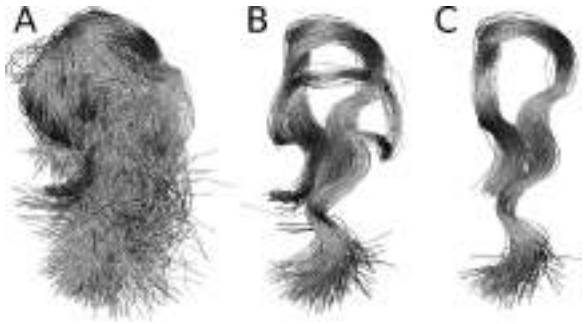


Figure 9.10: *Second cluster of trajectory 6 kF. A-) The raw cluster obtained by the original DP approach. B-) Cluster core obtained by the original DP approach. C-) refined cluster core obtained by RCDPeaks.*

exhibits considerable uniformity.

9.2.5 . Approaches for computing the quasi-MST in the HDBSCAN* variants

Computing a minimum spanning tree (MST) is an important step of the HDBSCAN methodology. However, determining the exact MST is a time-consuming task approachable from a heuristic perspective. In HDBSCAN* software, two such heuristics are employed: (i) the generic and (ii) the Prim alternatives.

9.2.5.1 Generic-based HDBSCAN*

The generic option of HDBSCAN* adopt the Single Linkage (SL) algorithm to obtain the approximate MST. SL is an agglomerative scheme that groups together elements in a bottom-up fashion. HDBSCAN* implements SL using three vectors called left (L), right (R), and current_distances (C). L and R contain the distances to be compared while C holds the results of such comparison. Concretely, for a trajectory T={1, 2, 3, ..., N}, the generic implementations of HDBSCAN* work as follows:

1. Initialization of L: L is defined as a vector of length N, filled with an infinite value at each index.
2. Initialization of R: an element i from T is selected, and R is filled with the distances from i to all not analyzed elements in T.
3. Initialization of C: the array C gets created and holds the minima of the element-wise comparison of R and L.
4. Edge formation and updating: the minimum distance in C ($d_{ij}, i \neq j$) is selected as the weight of a new edge between i and j. L is redefined as R and j becomes the next element to analyze.

Steps from 2 are repeated until all nodes in T gets analyzed.

9.2.5.2 Prim-based HDBSCAN*

Constructing an MST from a set of nodes through the Prims' algorithm can be done following the next iterative steps:

1. Find an edge between nodes i and j with the lowest distance.
2. Connect i and j if the resulting graph does not present cycles or self-connections.

The implemented Prims' algorithm in HDBSCAN* is slightly different from the original one. It starts selecting the first frame (i) as root node (instead of finding the lowest distance edge). An edge between i and its nearest neighbor j is created and j is taken as the next node to analyze. Then the lowest distance of the remaining points and j is selected as the weight of a new edge. Analyzing non-previously seen vertices, the algorithm is able to avoids cycles and self-connected nodes. However, it is not possible to obtain the MST because in each iteration, only the distances associated to a single node are taken into account.

9.2.6 . Impact of the vp-tree encoding on MDSCAN performance

Table 9.7: *Impact of the vp-tree encoding on MDSCAN run time, RAM consumption, and quasi-MST weight for each analyzed trajectory.*

Traj. Name	MDSCAN [with vptree]			MDSCAN [without vptree]		
	Run time *hh:mm:ss*	RAM peak *GB*	q-MST weight *mrd*	Run time *hh:mm:ss*	RAM peak *GB*	q-MST weight *mrd*
6 kF	0:00:06	0.18	1565.6521	0:00:04	0.10	1565.6497
30 kF	0:00:19	0.21	5286.8418	0:00:40	0.12	5286.8511
50 kF	0:01:37	0.26	2713.9600	0:02:00	0.16	2713.9534
100A kF	0:36:42	1.78	5516.5962	0:46:05	0.88	5516.4922
250 kF	0:37:46	1.20	17354.6699	1:16:55	0.67	17354.6445
500 kF	6:42:18	2.83	1258.8015	12:25:49	1.61	1238.6390
1 MF	21:01:06	7.67	39514.9727	37:20:55	4.10	39514.6328

9.2.7 . Cluster composition equivalence between MDSCAN and HDBSCAN alternatives

In the corresponding graphs of Figure 9.11, nodes depict clusters. The label of each node is composed of a letter and a cluster ID (ordering starts from 1, the most populated cluster for a particular letter). Each node's size correlates to its population, while its color corresponds to the average diameter of the represented cluster (the darker the node, the bigger its average diameter). We define as diameter the maximum pairwise distance between any cluster's frames. Edges exist if linked nodes have shared frames. Edges' color and size are consistent with the number of shared frames (the darker and broader the edge, the greater the number of common frames between linked nodes).

In the 6 kF trajectory (Figure 9.11A, Table 9.8), a total equivalence of the third and fourth clusters retrieved by all software is appreciated. A similar agreement is shown for cluster 5 of MDSCAN and the sixth one reported by the HDBSCAN* variants.

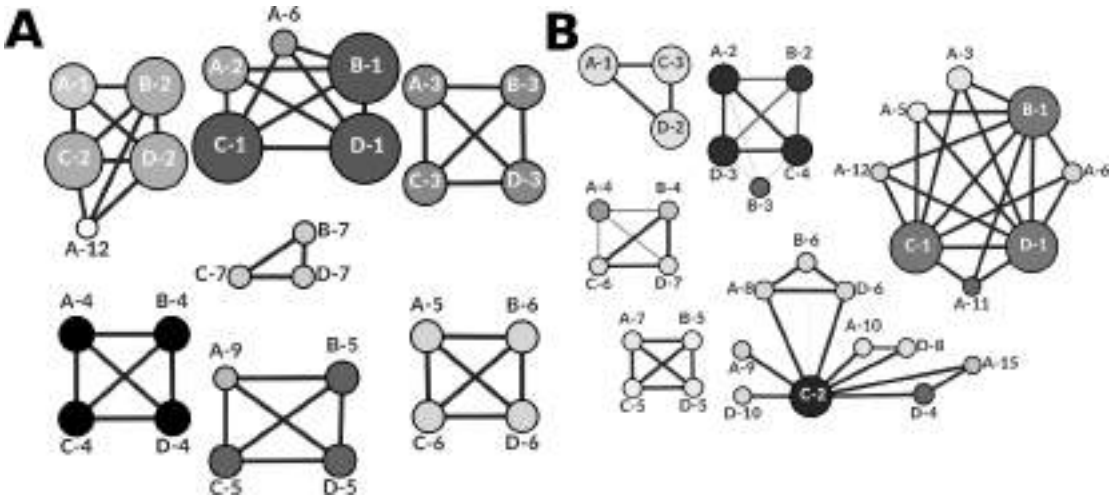


Figure 9.11: *Equivalence of clusters detected in trajectories 6 and 30 kF (*$\mathbf{A}$ *and* $\mathbf{B}$ *sections respectively) with software A- MDSCAN, B- the generic* RMSD*-based* HDBSCAN*, C- the generic Euclidean-based* HDBSCAN*, and D- the Prim Euclidean-based* HDBSCAN*. Each node represents a cluster. Nodes color corresponds to the average diameter of the cluster (the darker the node, the bigger its average diameter) while nodes' size correlates to their population. The color and size of edges map to the number of common frames between two clusters (the darker and wider the edge, the greater the number of common frames between the linked nodes.)*

Interestingly, the first and second clusters produced by all variants of HDBSCAN* are identical but MDSCAN splits each of them in two smaller groups of less diameter; A2 and A6 for the first cluster, and A1 and A12 for the second cluster. A9 is fully contained in the bigger and wider cluster 5 produced by HDBSCAN* alternatives. Finally, HDBSCAN* variants reported a seventh cluster not detected with MDSCAN, while MDSCAN identified another four clusters not recognized with any of the HDBSCAN* implementations: A7, A8, A10, and A11.

The relationship between the representative clusters reported for trajectory 30 kF (Figure 9.11B, Table 9.9) is not as pronounced as in the 6 kF case.

The only instance of total equivalence among the HDBSCAN* implementations is for their first cluster, which is split by MDSCAN in five smaller and tighter clusters (A3, A5, A6, A11, and A12). A high overlap (though not total) is also observed in several cases: (i) between the seventh cluster of MDSCAN and the fifth of HDBSCAN* variants, (ii) between B4, C6, and D7 (which contain some frames of A4), (iii) between A1, C3, and D2, and finally (iv) between A2, C4, and D3 (this partition is split by the generic RMSD-based, implementation of HDBSCAN* in B2 and B3). The big cluster C2 produced by the generic Euclidean-based implementation of HDBSCAN* contain frames that have been split into several smaller and tighter clusters by MDSCAN (A8, A9, A10, and A15), and by the Prim Euclidean-based HDBSCAN* (D4, D6, D8, and D10). Also, C2 shares some frames with the generic RMSD-based variant (B6).

Table 9.8: *Equivalence of representative clusters in the 6 kF trajectory*

Software		Cluster-ID	Size	Average Diameter ()	Intersecting clusters (# common frames)
MDSCAN		A1	213	2.36	B2(213), C2(213), D2(211)
		A2	210	2.61	B1(210), C1(210), D1(210)
		A3	200	2.81	B3(200), C3(200), D3(200)
		A4	145	3.41	B3(145), C3(145), D3(145)
		A5	131	2.27	B6(131), C6(131), D6(131)
		A6	99	2.67	B1(99), C1(99), D1(99)
		A7	80	4.44	-
		A8	79	4.91	-
		A9	78	2.48	B5(78), C5(78), D5(78)
		A10	74	5.33	-
		A11	70	3.08	-
		A12	62	2.07	B2(62), C2(62), D2(62)
HDBSCAN*	generic-RMSD	B1	363	3.10	A2(210), A6(99), C1(363), D1(363)
		B2	287	2.56	A1(213), A12(62), C2(287), D2(285)
		B3	200	2.81	A3(200), C3(200), D3(200)
		B4	145	3.41	A4(145), C4(145), D4(145)
		B5	135	3.05	A9(78), C5(135), D5(135)
		B6	131	2.27	A5(131), C6(131), D6(131)
		B7	73	2.33	C7(73), D7(73)
	generic-Eucl.	C1	363	3.10	A2(210), A6(99), B1(363), D1(363)
		C2	287	2.56	A1(213), A12(62), B2(287), D2(285)
		C3	200	2.81	A3(200), B3(200), D3(200)
		C4	145	3.41	A4(145), B4(145), D4(145)
		C5	135	3.05	A9(78), B5(135), D5(135)
		C6	131	2.27	A5(131), B6(131), D6(131)
		C7	73	2.33	B7(73), D7(73)
	Prim-Eucl.	D1	363	3.10	A2(210), A6(99), B1(363), C1(363)
		D2	285	2.55	A1(211), A12(62), B2(285), C2(285)
		D3	200	2.81	A3(200), B3(200), C3(200)
		D4	145	3.41	A4(145), B4(145), C4(145)
		D5	135	3.05	A9(78), B5(135), C5(135)
		D6	131	2.27	A5(131), B6(131), C6(131)
		D7	73	2.33	C7(73), D7(73)

9.3 . NUCLEAR: an efficient assembler for the FBDD of CMOs

9.3.1 . NUCLEAR performance in oligonucleotide searches

9.4 . *In-silico* design of selective CMO against BACE1

9.4.1 . BACE1 protein candidates selection

The PDB database was queried using the string *beta secretase 1 OR beta-secretase 1*, returning 415 structures. PDB files header contains a field of *related_entries* that could be useful to analyze, so we adjusted these 40 related structures for completeness to the already found ($415 + 40 = 455$). Fetched PDBs contain *chains* and *models* that must be split into individual files for further analyses. The splitting stage produced 884 individual

Table 9.9: *Equivalence of representative clusters composition in the 30 kF trajectory*

Software		Cluster-ID	Size	Average Diameter ()	Intersecting clusters (# common frames)
MDSCAN		A1	4251	2.8	C3(3986), D2(4090)
		A2	2646	5.76	B2(1939), B3(702), C4(2642), D3(2637)
		A3	1754	2.71	B1(1754), C1(1754), D1(1754)
		A4	999	4.11	B4(599), C6(550), D7(552)
		A5	870	2.52	B1(870), C1(870), D1(870)
		A6	735	2.98	B1(735), C1(735), D1(735)
		A7	680	2.65	B5(651), C5(644), D5(672)
		A8	641	2.95	B6(640), C2(641), D6(636)
		A9	506	3.04	C2(506)
		A10	471	2.71	C2(471), D8(412)
		A11	468	4.87	B1(468), C1(468), D1(468)
		A12	445	2.97	B1(445), C1(445), D1(445)
		A13	394	1.83	-
		A14	379	4.82	-
		A15	336	3.37	C2(336), D4(328)
HDBSCAN*	generic-RMSD	B1	6541	4.63	A3(1754), A5(870), A6(735), A11(468), A12(445), C1(6541), D1(6541)
		B2	1939	5.57	A2(1939), C4(1938), D3(1935)
		B3	702	4.87	A2(702), C4(702), D3(701)
		B4	698	3.29	A4(599), C6(550), D7(552)
		B5	651	2.5	A7(651), C5(644), D5(651)
		B6	640	2.95	A8(640), C2(640), D6(636)
	generic-Eucl.	C1	6541	4.63	A3(1754), A5(870), A6(735), A11(468), A12(445), B1(6541), D1(6541)
		C2	4632	5.89	A8(641), A9(506), A10(471), A15(336), B6(640), D4(893), D6(636), D8(507), D10(372)
		C3	3986	2.7	A1(3986), D2(3981)
		C4	2642	5.76	A2(2642), B2(1938), B3(702), D3(2637)
		C5	644	2.48	A7(644), B5(644), D5(644)
		C6	550	2.80	A4(550), B4(550), D7(549)
	Prim-Eucl.	D1	6541	4.63	A3(1754), A5(870), A6(735), A11(468), A12(445), B1(6541), C1(6541)
		D2	4090	2.73	A1(4090), C3(3981)
		D3	2638	5.76	A2(2637), B2(1935), B3(701), C4(2637)
		D4	893	4.89	A15(328), C2(893)
		D5	672	2.59	A7(672), B5(651), C5(644)
		D6	636	2.93	A8(636), B6(636), C2(636)
		D7	552	2.80	A4(552), B4(552), C6(549)
		D8	507	2.81	A10(412), C2(507)
		D9	401	4.08	-
		D10	372	2.89	C2(372)
		D11	372	4.44	-

structures filtered to preserve only those similar to a BACE1 reference (1SGZ-A).

A pairwise sequence alignment was performed between the 884 targets against the 1SGZ-A reference. The alignment score (ranging from 0 to 100) was calculated as follows, where $nresids\_aligned$ represents the number of residues in the target that aligns to the reference and *nresids_reference* is the total number of residues in the reference:

$$score = (nresids\_aligned/nresids\_reference) * 100 \tag{9.4}$$

The following fields for each structure were computed: (i) a unique identifier {PDB code}-{chain/model}, (ii) the score (by Equation 9.4), (iii) the sequence identity, (iv) the sequence overlap, (v) the crystallization pH, (vi) the resolution, (vii) R (distance from *CG-Asp32* to *OH-Tyr71*), and (i) psi (pseudo-dihedral *C-Trp76, N-Val69, CA-Thr72, CA-Gln73*).

Structures with sequence identity less or equal to 75% against 1SGZ-A were removed from further analyses. There were many presenting gaps from the remaining 693 structures similar to 1SGZ-A. We defined a *gap* as a protein's region with missing residues (discontinuities in the residue enumeration) and discarded affected structures.

The 171 structures without gaps were submitted to a clustering procedure aiming to select just a few geometrically distant models for docking stages. We conserved all

Table 9.10: NUCLEAR *performance for the global geometric (GG) oligonucleotide searches performed on proteins 2XNR, 5WWX, and 5ELH. The following timings are reported: (i) the time taken to parse the* MCSS *docking distributions (**t1**), (ii) the time taken to construct both binary matrices (**t2**), and (iii) the time taken to perform the actual search of chains (**t3**). Additionally, the total elapsed time and the peak memory usage for each job is reported.*

Protein	RMSD	$d(C5' - O3')$ [Å]	#Poses	#Chains	t1	t2	t3	Total time [seconds]	RAM Peak [MB]
2XNR	0	3	9075	15694	2	64	5	71	396
		4		677268	2	63	13	78	269
		5		6492116	2	62	82	146	265
		6		34213514	2	69	425	496	396
	1	3	5552	3849	46	24	4	74	396
		4		164885	44	23	6	73	224
		5		1595613	44	21	22	87	225
		6		8447693	47	24	103	174	396
	2	3	2696	338	17	3	0	20	396
		4		17638	17	3	4	24	219
		5		176381	19	4	7	30	219
		6		953535	20	5	17	42	396
5WWX	0	3	22818	37619	6	315	5	326	396
		4		1658706	6	309	27	342	517
		5		17435561	6	301	241	548	396
		6		98355759	6	349	1396	1751	396
	1	3	14664	9001	194	112	5	311	396
		4		412382	192	111	10	313	393
		5		4337966	191	110	61	362	523
		6		24619477	206	124	343	673	396
	2	3	7494	1130	79	22	1	102	396
		4		8334	80	27	5	112	301
		5		537549	82	26	11	119	301
		6		3068691	93	29	44	166	396
5ELH	0	3	8473	14965	2	81	5	88	396
		4		562254	2	99	22	123	272
		5		5008076	2	79	64	145	271
		6		25863579	2	92	328	422	396
	1	3	4881	3037	35	26	3	64	396
		4		114771	34	25	6	65	213
		5		1056292	35	25	17	77	215
		6		5508112	38	27	71	136	396
	2	3	2262	245	13	3	0	16	396
		4		11335	12	8	0	20	209
		5		110845	13	5	5	23	209
		6		596755	15	5	12	32	396

equivalent atoms between the 171-group (2798). The Active Site Region (ASR) was defined as the atoms within 8Å of the residues Asp32 and Asp228 plus the atoms involved in the flap region (residues from 67 to 77). Then we used the BitQT formalism (see Section 4.1) with a cutoff of 1Å on the RMSD matrix of the 171-group ASRs.

Nevertheless, the ASR is rigid, exhibiting only modest conformational variability in the flap zone. By using BitQT, five homogeneous clusters were obtained. However, the RMSD metric is inaccurate to capture the Tyr-71 orientation, an essential residue in the BACE1 active site and the most significant source of conformational diversity in the clustered structures.

In the particular case of the BACE1 ASR, a couple of order parameters character-

izing the flap conformation have been previously identified: R and psi. We plotted the R vs. psi values of the 693 individual structures similar to 1SGZ-A. As it can be appreciated in Figure 9.12A, the R/psi regions observed roughly correspond to the regions described in the MD explorations of holo/apo BACE1[181].

To produce Figure 9.12B, we used the density-based clustering algorithm HDBSCAN in the Cartesian space of R/psi values (both normalized to 0-1 range beforehand). Seven dense clusters were found.

Keeping the previous cluster labels, we restricted the analysis to the 171-structures group without gaps. As depicted in Figure 9.12C, six clusters were retrieved. In Table 9.11, we detail some information on the representative structure of each.

Table 9.11: *Representative structures after the HDBSCAN clustering procedure.*

Cluster	Selected	pH	Resolution	R / Psi	#waters	# waters@ASR
1	**1SGZ-A**	**6.5**	**1.80**	**11.29 / 10.77**	**234**	**13**
2	3SKG-D	6.2	2.88	7.52 / 12.71	20	2
3	**6UVV-A**	**7.4**	**1.63**	**7.6 / -6.28**	**588**	**20**
4	4FSL-A	6.2	2.50	6.64 /-19.39	215	13
5	6BFW-A	N/A	1.84	6.68 / 19.21	416	17
6	**5MCQ-A**	**4.5**	**1.82**	**6.60 / -2.15**	**551**	**14**
6	**4GID-A**	**6.5**	**2.00**	**6.44 / -9.15**	**316**	**9**

It should be highlighted that cluster 2 only has 1 member with poor resolution, cluster 4 has only 2 structures with poor resolution and similar to cluster 6, while cluster 5 structures are similar to cluster 6. So the final set of BACE1 conformations comprises clusters 1, 3, and 6 as candidates to start docking studies. These structures have good resolution and an appropriate number of water molecules at the ASR.

There is a region of BACE proteins called exosite surrounded by three loops where inhibitors also bind preferentially (loop C, D, and F comprising residues 254-257, 270-274, and 309-320, respectively)[320]. Apart from the three structures selected (1SGZ, 4GID, and 6UVV), which are conformationally different at the ASR, we chose 5MCQ, which belongs to cluster 6 (so having a similar arrangement of the ASR that 4GID) but whose exosite presents a different conformation.

9.4.2 . Modified nucleotides present at the MCSS library

9.4.3 . Key interactions of CMO-BACE-X complexes

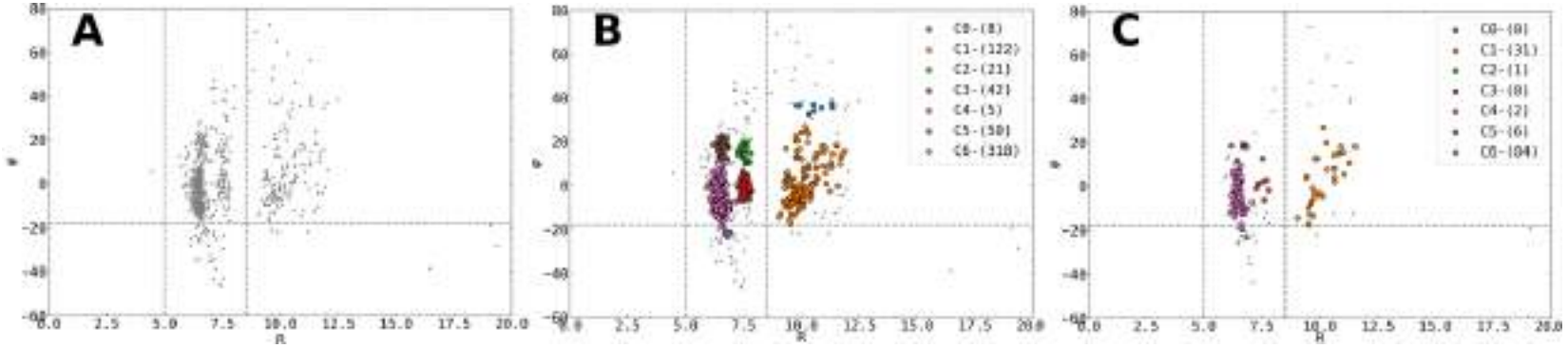


Figure 9.12: *Plot of distance R vs. pseudo-dihedral psi of the 693 individual structures similar to 1SGZ-A)*

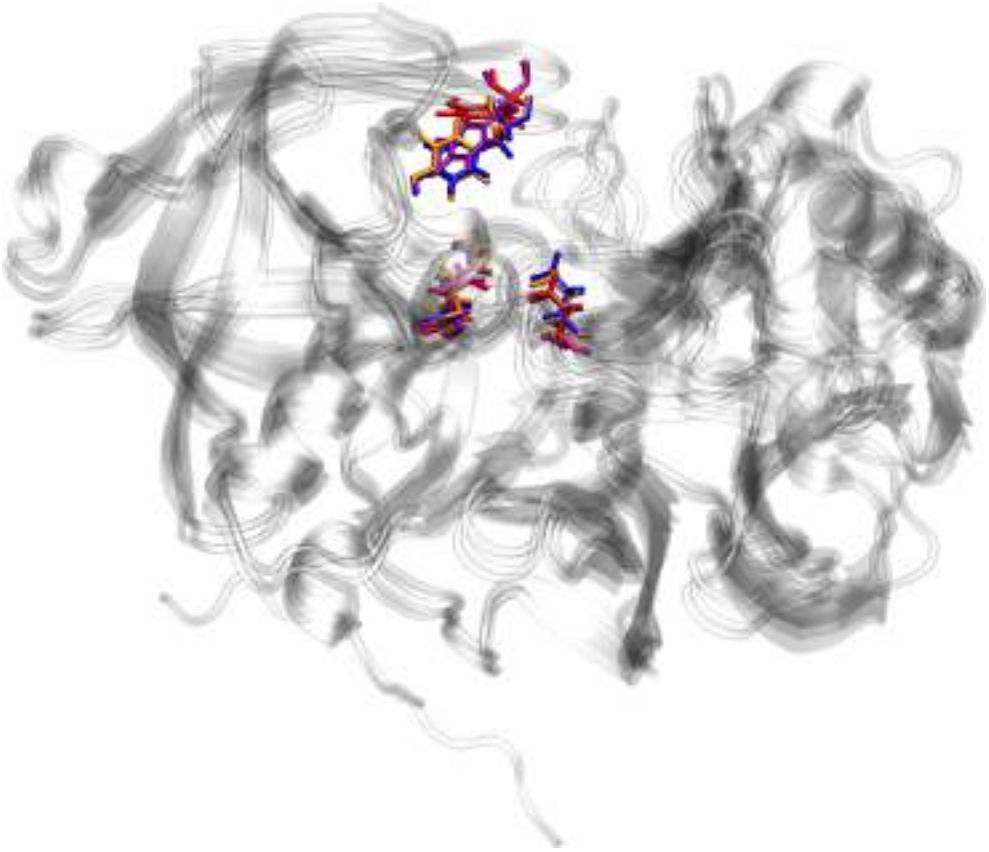


Figure 9.13: *Superposition of the four* BACE1 *selected conformations; 1SGZ (red), 4GID (blue), 5MCQ (orange), and 6UVV (purple).*

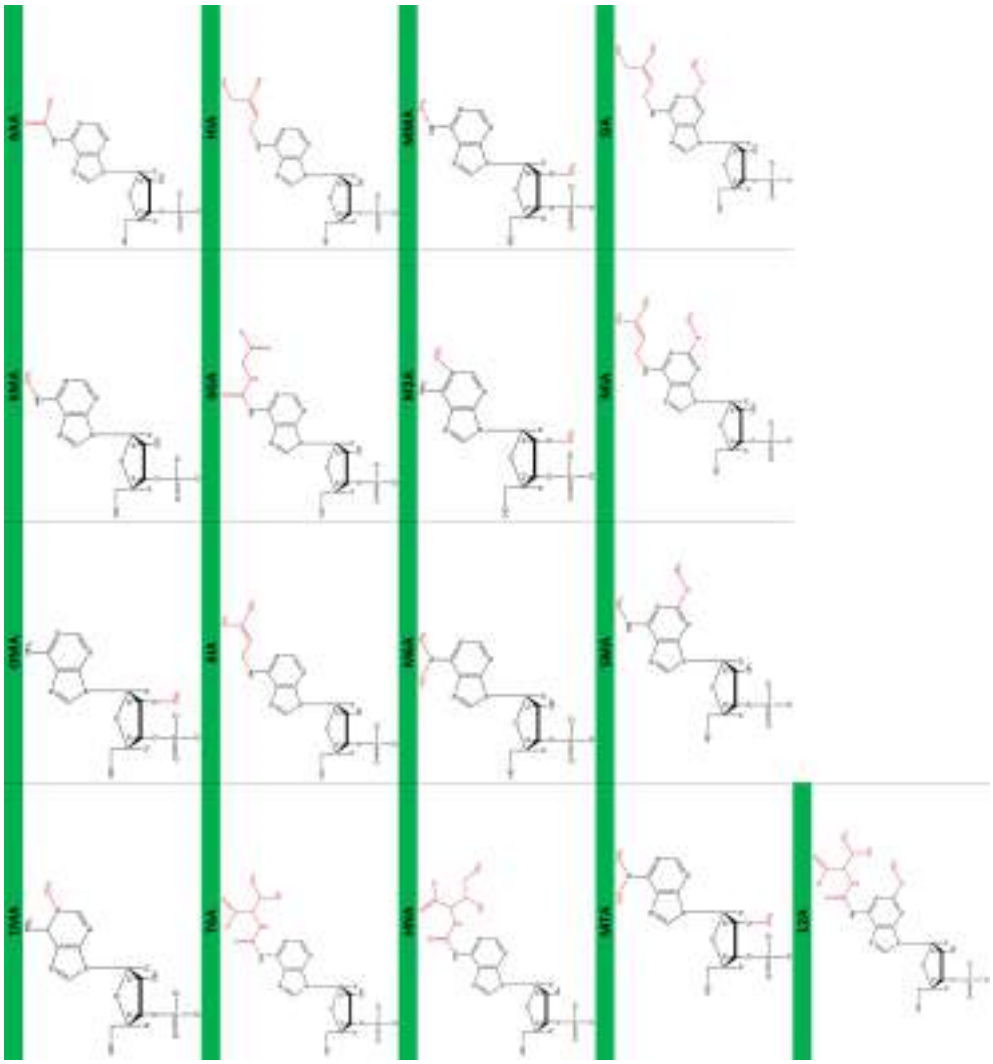


Figure 9.14: *A-derived modified nucleotides present at the* MCSS *library.*

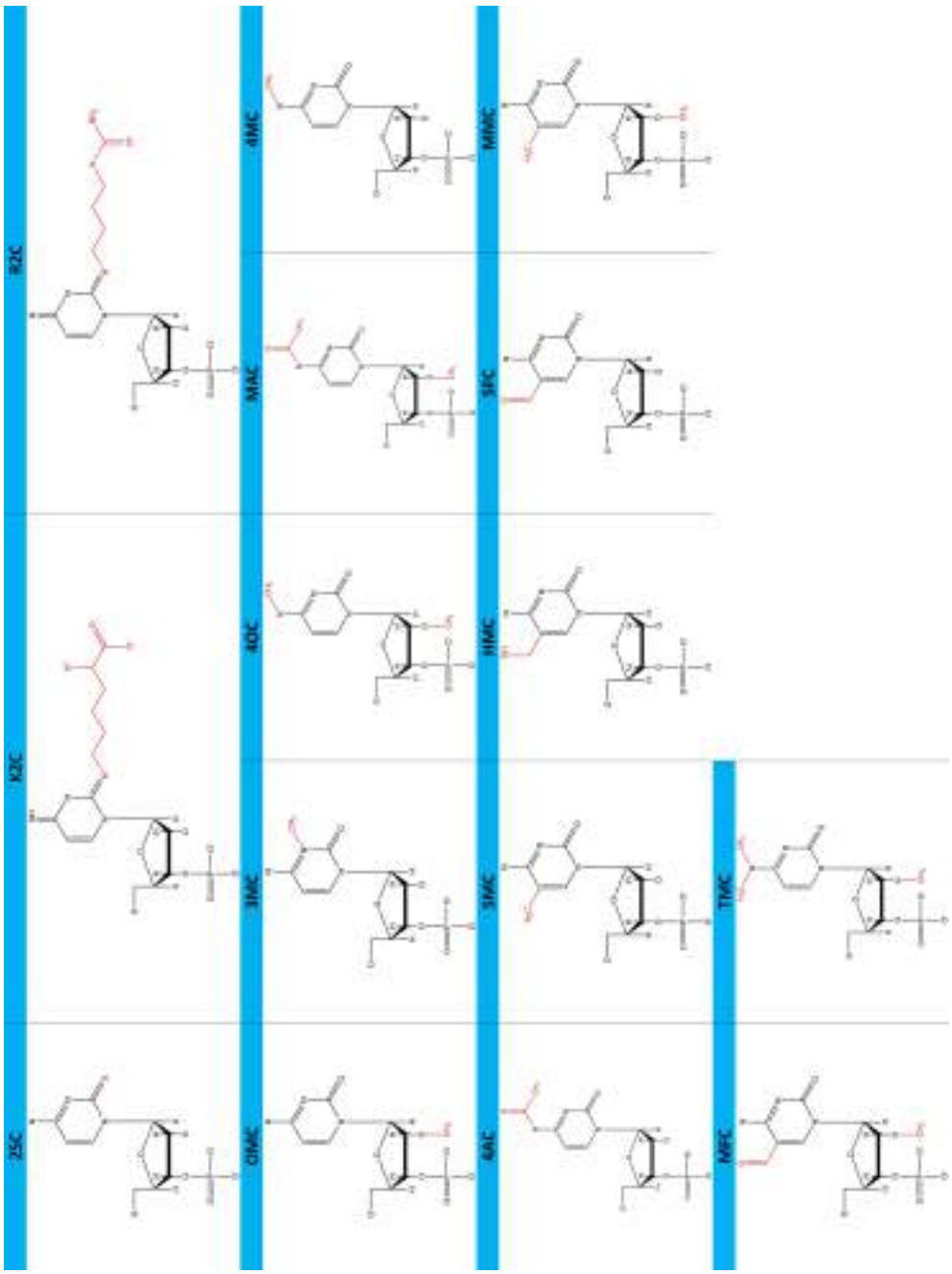


Figure 9.15: *C-derived modified nucleotides present at the* MCSS *library.*

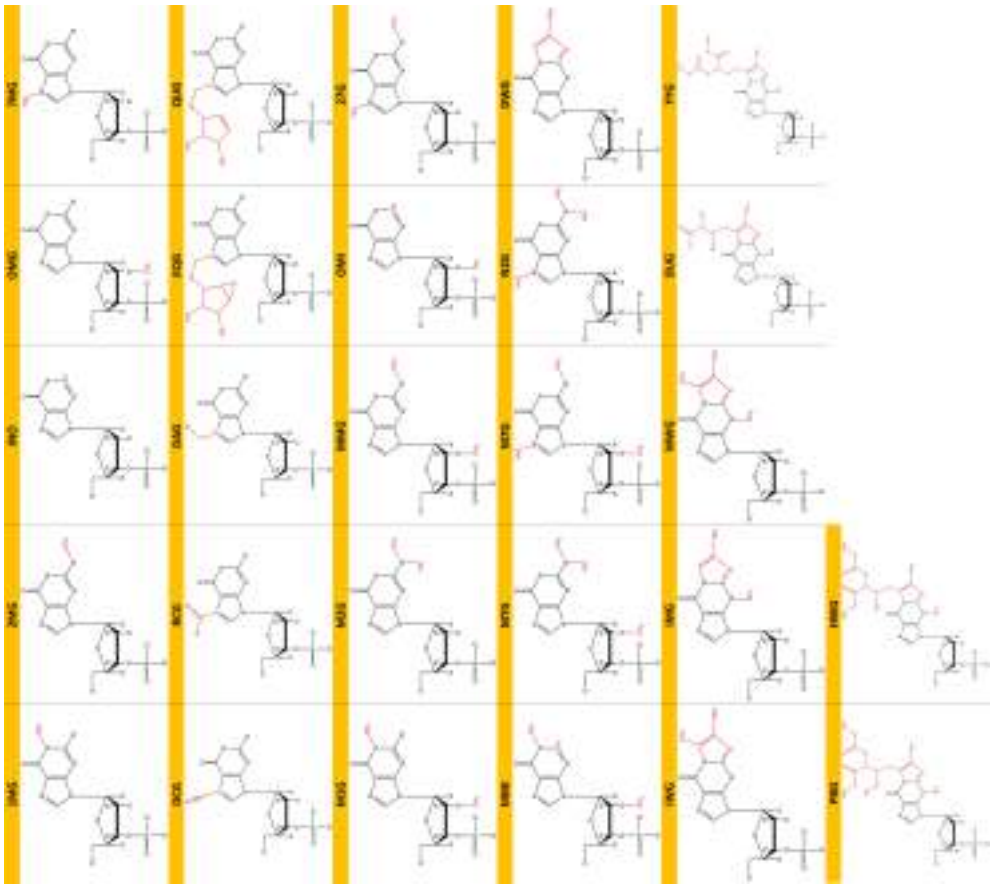


Figure 9.16: *G-derived modified nucleotides present at the* MCSS *library.*

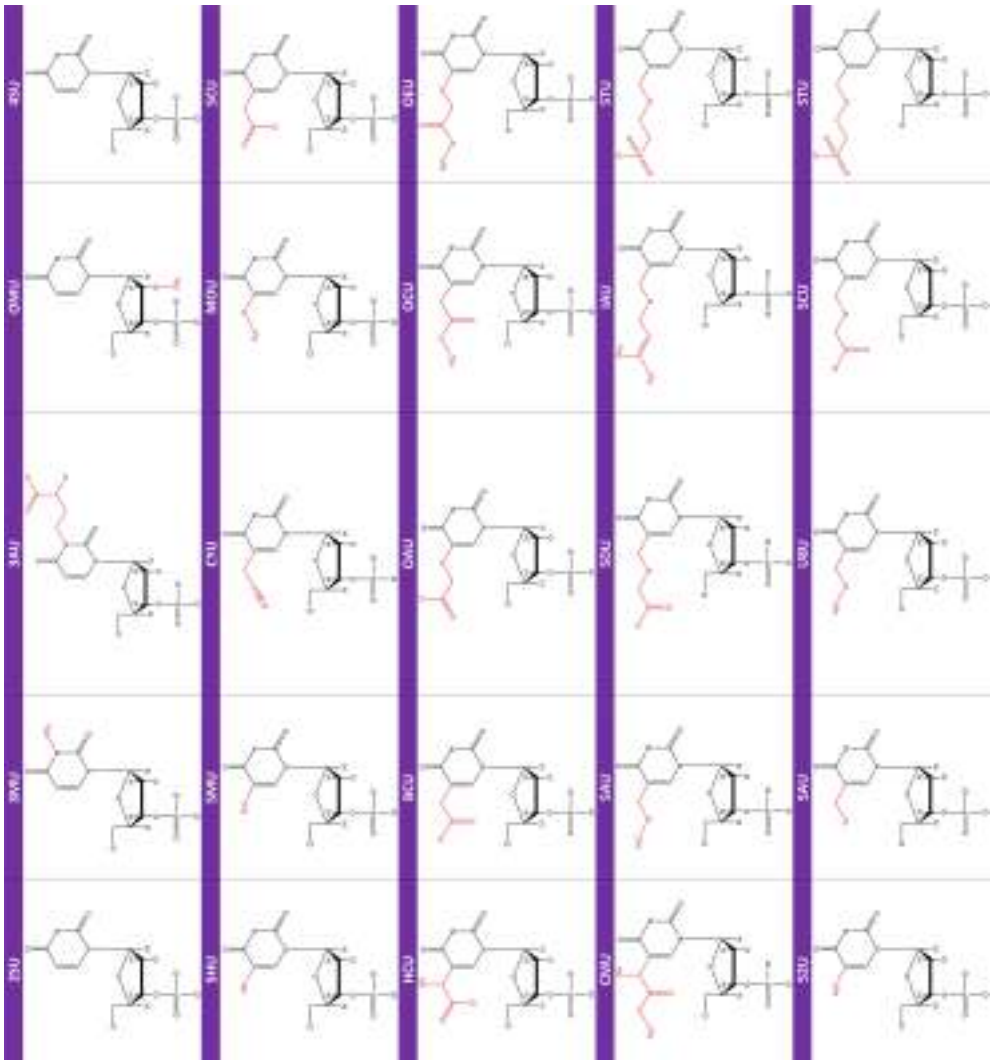


Figure 9.17: *U-derived modified nucleotides present at the* MCSS *library.*

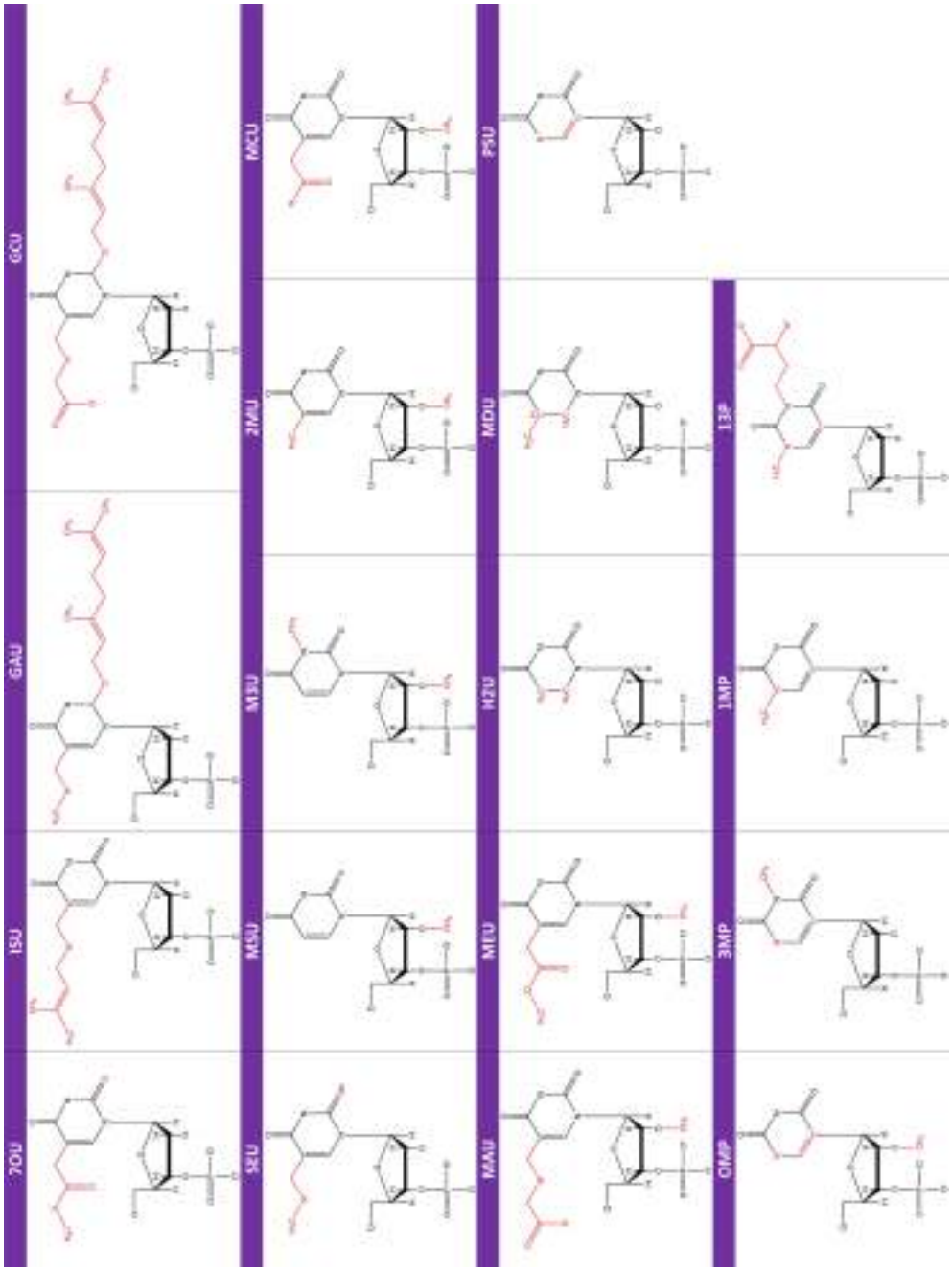


Figure 9.18: *U-derived modified nucleotides present at the* MCSS *library (continuation of Figure 9.17).*

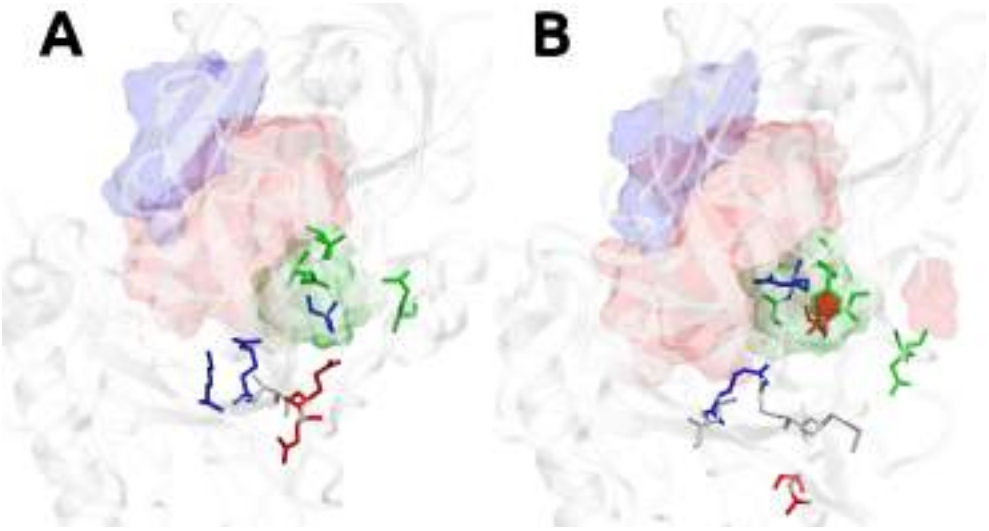


Figure 9.19: *"Near-1os loop region" of* BACE1 *(A) and* BACE2 *(B) colored by residue type.*

RÉSUMÉ ÉTENDU

INTRODUCTION

La découverte de médicaments est un processus complexe et long qui comporte plusieurs étapes, de l'identification d'une cible biologique à l'approbation d'un nouveau médicament par les organismes de réglementation. La tâche peut prendre plusieurs années et des milliards de dollars, avec un taux d'échec élevé à chaque étape[1]. La première de ces étapes consiste à identifier une cible biologique qui joue un rôle dans le processus de la maladie. Une fois identifiés, les chercheurs utilisent diverses approches pour découvrir des candidats-médicaments potentiels qui peuvent interagir avec la cible et moduler son activité[2].

De la famille de cibles médicamenteuses connues jusqu'à présent, les protéines sont les membres les plus courants[3]. Leur inhibition joue un rôle essentiel dans la découverte de médicaments, car elle fournit un moyen de supprimer leur participation à une maladie particulière. Cependant, le blocage d'une protéine spécifique est difficile et rencontre souvent l'effet dit *off-target*. Ce terme décrit les événements qui peuvent se produire lorsqu'un médicament se lie à des cibles (protéines ou autres molécules dans le corps) autres que celles pour lesquelles il était censé se lier, provoquant des effets secondaires inattendus et potentiellement nocifs[4].

La conception de médicaments par fragments (FBDD) est un moyen rationnel de concevoir la conception d'inhibiteurs de protéines. Il commence par une petite collection de molécules de faible masse moléculaire et de faible affinité appelées fragments, puis les intègre dans les médicaments de plus haut poids moléculaire[5]. Il y a plusieurs cas de réussite lorsque FBDD a été appliqué à la conception et à la découverte de médicaments, avec plus de 30 candidats à base de fragments entrant dans la clinique depuis le milieu des années 1990[5–9].

L'acide ribonucléique (RNA) et les molécules dérivées sont apparus comme des outils prometteurs pour l'inhibition sélective des protéines en raison de leur spécificité élevée, de leur faible immunogénicité et de leurs propriétés physico-chimiques ajustables[10]. Par exemple, le développement d'aptamères (oligonucléotides courts simple brin d'acide désoxyribonucléique (DNA) ou RNA comme agents thérapeutiques a fait l'objet d'intenses recherches, de nombreuses études faisant état de leur application réussie dans le traitement de diverses maladies[11]. Les avantages des aptamers comprennent leur facilité de génération, leur faible coût de fabrication et leur faible immunogénicité. Cependant, ces molécules doivent subir des modifications chimiques pour éviter leur susceptibilité inhérente à l'hydrolyse des nucléases et à une clairance rapide par filtration glomérulaire[12].

Les alliés éprouvés des campagnes expérimentales de découverte de médicaments sont les méthodes computationnelles ou *in silico*, qui réduisent les coûts de temps et de ressources grâce à des simulations virtuelles. Lorsque l'on recherche de nouveaux médicaments, une étape névralgique consiste à examiner d'immenses bases de données

représentatives de l'espace chimique du médicament pour trouver un remède moléculaire approprié à une maladie. Les méthodes d'amarrage informatique jouent un rôle essentiel, et de nombreuses alternatives sont disponibles pour les chercheurs[13]. Dans l'arène FBDD, le logiciel Multiple-Copy Simultaneous Search (MCSS)[14] se distingue comme une méthodologie virtuelle pionnière pour l'amarrage qui a été précédemment couplée à d'autres logiciels pour joindre des fragments dans des composés de départ sujets à optimisation[15–19].

Les algorithmes de clustering (dédiés au regroupement d'entités similaires en ensembles appelés clusters)[20] sont utilisés à différentes étapes du pipeline FBDD (bien que souvent de manière transparente pour les utilisateurs), principalement pour regrouper des fragments ou des composés similaires en fonction de leurs propriétés structurales ou physicochimiques. La conception de nouveaux algorithmes de clustering efficaces ou l'optimisation de ceux actuellement utilisés est obligatoire pour faire face à la taille croissante des ensembles moléculaires générés par les techniques de calcul.

La présente thèse porte sur la conception computationnelle basée sur des fragments d'oligonucléotides chimiquement modifiés (inspirés par le développement et le succès des aptamères) pour l'inhibition sélective des protéines, en utilisant β-site Amyloid Precursor Protein Cleaving Enzyme 1 (BACE1) comme étude de cas. BACE1 est une cible thérapeutique bien établie pour la maladie d'Alzheimer (AD) en raison de son rôle essentiel dans la production de peptides amyloïdes-bêta, qui sont les principaux constituants des plaques amyloïdes dans le cerveau des patients atteints de la maladie d'Alzheimer. L'un des principaux inconvénients du ciblage de BACE1 réside dans l'inhibition hors cible démontrée par une protéase apparentée, β-site Amyloid Precursor Protein Cleaving Enzyme 2 (BACE2)[21].

Bien que les thérapies RNA (y compris les alternatives d'aptamères) soient de plus en plus populaires, il n'existe pas de méthodologie pour la conception rationnelle d'oligonucléotides chimiquement modifiés sélectifs pour des applications médicales, ce qui constitue le **problème scientifique** ici adressé.

Le travail suivant a été conçu sur l'hypothèse globale **hypothèse** que les principes FBDD peuvent être appliqués efficacement à la conception rationnelle *in silico* d'oligonucléotides chimiquement modifiés avec une affinité et une sélectivité élevées pour les cibles protéiques.

L'objectif principal de cette thèse est de développer un cadre de calcul intégré pour la conception par fragments d'oligonucléotides chimiquement modifiés avec une affinité et une sélectivité élevées pour les cibles protéiques, en utilisant BACE1 comme preuve de concept pertinente. Les **objectifs spécifiques** qui ont guidé nos efforts sont les suivants :

1. Évaluer les pouvoirs d'arrimage et de criblage de la methodologie MCSS sur un benchmark représentatif de complexes nucléotidiques de protéines.

2. Optimiser des algorithmes de clustering populaires qui interviennent à des phases distinctes du FBDD.

3. Mettre en œuvre un schema informatique efficace pour assembler des nucléotides

(fragments) chimiquement modifiés sur des oligochaînes.

4. Valider un workflow de calcul pour proposer des oligonucléotides chimiquement modifiés comme inhibiteurs de protéines sélectives en utilisant BACE1 comme étude de cas.

Main results

PRÉDICTIONS DE LA LIAISON ET LA SÉLECTIVITÉ DES NUCLÉOTIDES AVEC MCSS

Fréquemment, les workflows virtuels de conception de médicaments par fragments présentent une limitation significative; le manque de performance des méthodes d'amarrage en raison de la nature approximative de leurs fonctions de score. Une hypothèse fondamentale de ce manuscrit est qu'en employant le logiciel MCSS, une performance accrue dans l'amarrage et la puissance de criblage est possible, faisant de cet outil un choix approprié pour les procédures de conception de médicaments à base de fragments impliquant des nucléotides.

Nous avons présenté un ensemble de données actualisé et représentatif de complexes protéine-nucléotide à haute résolution dans lesquels seuls les ligands mono-phosphate de nucléotide, en tant que ligands mono-résidu, sont inclus. La composition globale du référentiel, les descripteurs utilisés pour caractériser les sites de liaison à l'étude et la répartition des contacts sont analysés.

Nous avons accordé une attention particulière à l'effet des modèles de solvants et des patchs de phosphate sur le nombre de poses générées et la fraction de poses natives obtenues. Le nombre total de poses générées dépend principalement du modèle de solvant et du timbre de phosphate dans une moindre mesure.

La présence de molécules d'eau explicites réduit partiellement le volume moléculaire accessible pour les nucléotides dans la région de liaison. Ainsi, le nombre de poses générées avec le modèle SCAL est beaucoup plus grand que celui généré avec l'un des modèles de solvants hybrides : SCALW, FULLW et STDW (Figure 9.20). La comparaison des distributions brutes et en cluster montre également que le modèle SCAL présente la redondance la plus élevée dans les poses générées, démontrée par la plus grande différence entre les distributions brutes et en cluster pour chaque patch.

La fraction des poses natives sur l'ensemble de la distribution MCSS pour tous les modèles et patchs de solvants est similaire, sauf pour R310. Le patch R310 porte un groupe méthyle dans l'un des phosphates oxygène. Ce groupe confère la capacité d'établir plus de contacts hydrophobes que les autres patchs. Le modèle SCAL montre une fraction significativement plus faible de poses natives que les modèles solvatés malgré un nombre beaucoup plus important de poses générées (Figure 9.20).

Quant au nombre de poses, les distributions brutes et groupées sont plus dispersées en l'absence de molécules d'eau. Dans les modèles solvatés, les fractions de poses natives pour SCALW et STDW sont très similaires. D'autre part, le modèle FULLW a plus de cas où aucune pose native n'est trouvée, comme le montre le déplacement à zéro de la

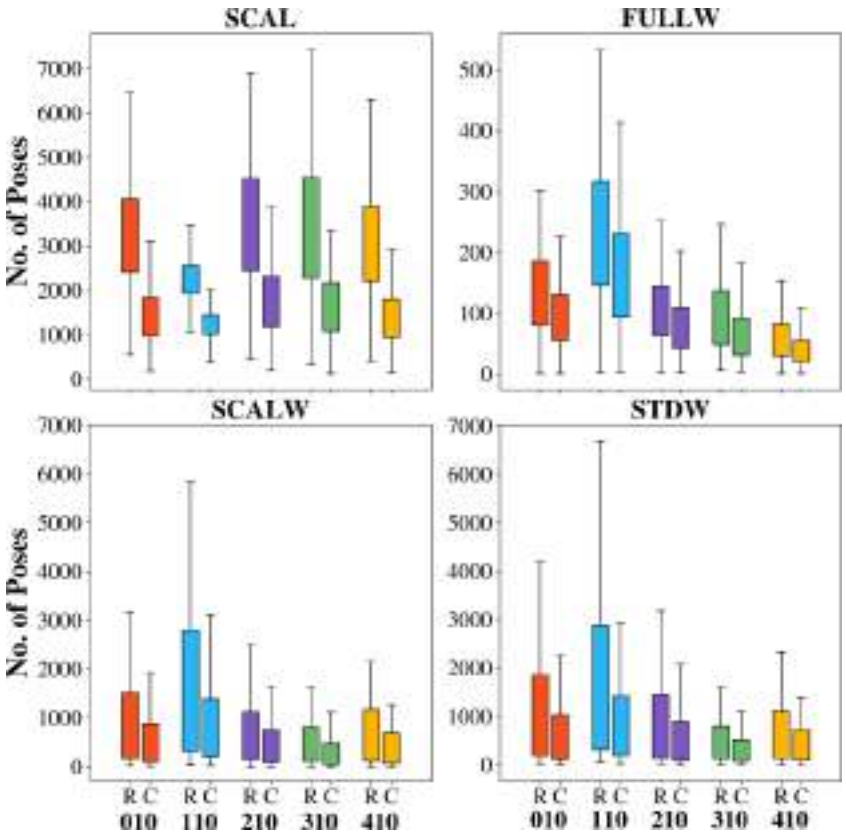


Figure 9.20: *Nombre de poses générées pour les 121 complexes protéine-nucléotide pour chaque nucléotide 5' patché (010, 110, 210, 310, 410). Les résultats pour les distributions brutes (**R**) et groupées (**C**) sont affichés.*

première section inter-quartile sur les boxplots (Figure 9.20).

Après cela, la puissance d'amarrage de MCSS, ainsi que sa puissance de criblage, ont été respectivement évaluées. La performance en puissance d'amarrage est évaluée sur tous les modèles et patchs en utilisant les mesures standard basées sur les poses natives trouvées dans les scores Top-1 à Top-100 avec les rangs intermédiaires: Top-5, Top-10 et Top-50. Les meilleures performances sont obtenues avec les modèles SCALW et STDW, quel que soit le patch utilisé (Figure 9.21). Le modèle STDW surpasse légèrement le modèle SCALW dans le Top-1 et le Top-10 pour tous les patchs (sauf pour R310, où la performance est équivalente pour le Top-10).

La meilleure performance est obtenue pour le patch R310. Il a un taux de réussite de 40% dans le Top 1, plus de 60% dans le Top 10 et plus de 80% dans le Top 100. Cependant, le gain de performance concernant les autres patchs est infime dans le Top-10 et le Top-50. Le clustering ne modifie pas les tendances générales observées dans les distributions brutes, mais il augmente légèrement la performance dans le Top-100 et, dans une moindre mesure, dans le Top-i. La discussion s'est terminée par la présentation des caractéristiques moléculaires associées au manque de prédictions.

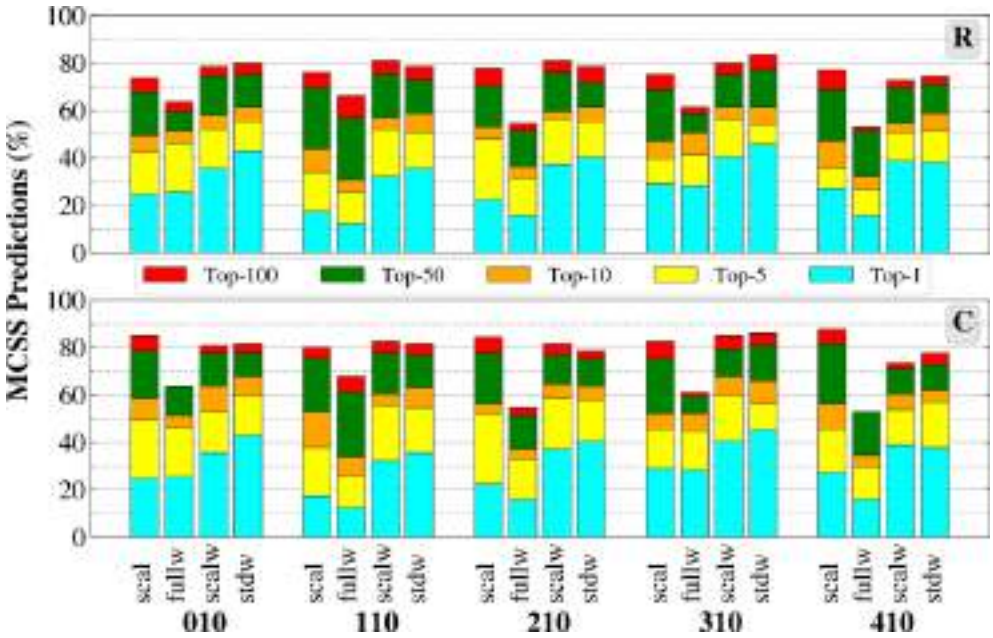


Figure 9.21: *Représentation de l'histogramme empilé des poses natives classées Top-i générées pour les 121 complexes protéine-nucléotide pour chaque patch nucléotidique. Les distributions brutes (en haut) et groupées (en bas) de textbfR: et de textbfC: sont affichées.*

RÉINVENTER LA ROUE DU REGROUPEMENT MOLÉCULAIRE

Nous avons présenté nos efforts pour diminuer les ressources spatiales de quatre algorithmes de clustering géométriques déjà appliqués aux ensembles moléculaires : (i) les algorithmes Quality Threshold (QT), (ii) le Daura, (iii) le Density Peaks (DP), et (iv) le Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN).

Nos implémentations ont été comparées aux méthodes alternatives les plus utilisées. Pour chacun d'entre eux, une nouvelle idée qui a eu un impact significatif sur leur consommation de mémoire a été développée. Un encodage binaire de la similarité moléculaire par paires (ainsi que la possibilité de traduire les étapes de clustering en opérations bit à bit) a été appliqué au clustering de Daura et du Quality Threshold. En outre, une correction méthodologique a été soulevée pour ces deux algorithmes qui ont été incorrectement et systématiquement utilisés de manière interchangeable.

QTPy a seulement été suggéré comme preuve de concept pour mettre en évidence les inexactitudes des autres alternatives de clustering qui prétendent sans faille effectuer QT. Il ne constitue pas une optimisation correcte, et sa complexité spatiale est $O(n^2)$. Néanmoins, nous avons utilisé des valeurs flottantes de demi-précision pour représenter RMSD, permettant ainsi à la matrice de similarité QTPy de consommer la moitié de l'espace requis par les alternatives qui utilisent des matrices de similarité de flotteurs de précision simple, telles que l'option gromos de GROMACS.

BitQT et **BitClust** adoptent la même approche pour raccourcir les exigences de mémoire des algorithmes Daura et QT en codant les distances entre paires RMSD en

bits. Par conséquent, malgré leur complexité spatiale quadratique, ils peuvent traiter des trajectoires considérablement plus grandes que les implémentations existantes.

L'algorithme **DP+** représente une tentative d'atténuer la complexité spatiale quadratique intrinsèque aux approches DP. Au lieu de construire une matrice complète, notre méthode fonctionne sur des vecteurs transitoires de taille N (où N est le nombre de conformations moleculaires) et deux tas qui peuvent étendre l'utilisation de la mémoire. Le tas secondaire comprend les tuples (i, rho), qui représentent l'indice et la valeur de densité de la trame.

Dans le pire des cas, ce tas s'étend aux entrées N avec une complexité spatiale de $N \cdot (O(1) + O(1)) \equiv O(N)$, démontrant ainsi une mise à l'échelle linéaire avec la longueur de la trajectoire. Le tas primaire comprend les éléments mentionnés ci-dessus et un sous-ensemble plus petit eta_i de taille $0.02 \cdot N$. Si nous traitons ceci comme un facteur constant c, même si le tas principal atteint sa taille maximale de N tuples, la complexité reste $N \cdot (O(1) + O(1) + O(c)) \equiv O(cN)$, toujours linéaire. Il convient de noter que de tels scénarios extrêmes sont peu probables avec une sélection prudente des paramètres.

L'objectif principal de **MDSCAN** était d'atténuer la complexité quadratique des alternatives HDBSCAN* disponibles lors de l'utilisation de mesures de grande dimension comme la RMSD. Remarquablement similaire à DP+, MDSCAN utilise les mêmes structures de données de tas et vecteurs transitoires déjà détaillés. La seule distinction réside dans le type d'information que ces tas contiennent; par conséquent, l'analyse de complexité spatiale du pire des cas présentée pour DP s'applique à MDSCAN. Cependant, MDSCAN traite le fichier de trajectoire différemment de tous les autres logiciels proposés ici, car un arbre de "vantage points" est construit, nécessitant une copie de la trajectoire. Ce traitement n'augmente pas la complexité spatiale de l'algorithme, bien qu'il nécessite plus d'espace par rapport à DP.

NUCLEAR: UN ASSEMBLEUR EFFICACE POUR LA CONCEPTION PAR FRAGMENTS D´OLIGONUCLÉOTIDES CHIMIQUEMENT MODIFIÉS

Bien que plusieurs outils de liaison de fragments soient disponibles dans la littérature, aucun ne convient à être inclus dans notre approche principalement pour les raisons suivantes : (i) ils ne peuvent pas fonctionner avec les formats de fichiers provenant des logiciels MCSS et CHARMM (PSF, DCD), (ii) ils ne sont pas conçus pour lier les oligonucléotides de C5' à O3' garantissant des solutions sans conflit, et surtout (iii) ils sont incapables de traiter de gros volumes de données.

En tenant compte des limites susmentionnées, nous avons présenté un nouveau logiciel appelé NUCLEotide AssembleR (NUCLEAR). NUCLEAR peut effectuer différents types de recherches d'oligonucléotides et récupérer des "hotspots" dans le récepteur à partir des distributions de fragments ancrés. Après avoir présenté les flux de travail disponibles dans NUCLEAR, nous avons détaillé le protocole de recherche des points chauds à la surface du récepteur.

Plus tard, les recherches d'oligonucléotides sous contrainte de séquence et spatiale sont décrites. Enfin, nous avons essayé la reproduction de trois structures cristallines en

utilisant NUCLEAR comme études de cas pour discuter du coût de calcul des principales étapes de l'algorithme, et pour approfondir ses limites.

CONCEPTION *IN-SILICO* D´OLIGONUCLÉOTIDES CHIMIQUEMENT MODIFIÉS SELECTIFS CONTRE BACE-X

Une fois que la nature des mono-nucléotides modifiés chimiquement et les conformations protéiques considérées sont présentées, nous illustrons comment attaquer le problème de l'assemblage des oligonucléotides dans deux situations; quand des informations pratiques sur les interactions protéines-receptor sont limitées ou indisponibles, et quand les connaissances sur ces interactions sont accessibles via des bases de données moléculaires.

Dans la dernière partie de la discussion, nous nous concentrons ensuite sur le discernement si les oligonucléotides produits ont le potentiel d'être sélectifs contre la protéine BACE1 sur BACE2 en implémentant plusieurs contraintes de sélection à leurs modes de liaison. La base moléculaire sous-jacente aux modes de liaison de la Figure 9.22 a été étudiée en examinant les contacts étroits, les liaisons hydrogène, les ponts salins, les interactions hydrophobes, $\pi-\pi$, T-stacking et $\pi-$cation entre l'CMOs et l'BACE-X. Cette analyse a été effectuée en utilisant les valeurs géométriques par défaut du programme BINANA et pour tous les membres des clusters inspectées.

Le mode de liaison inhabituel du groupe 4 chez BACE1, soutenu par l'empilement de $\pi-\pi$ (TYR68), les liaisons hydrogène (LYS75, GLU77, SER328) et les interactions des ponts de sel (LYS75 et GLU77), n'a pas de contrepartie chez BACE2. Cette liaison, bien qu'anormale, s'aligne sur les recommandations d'autres auteurs de cibler la région du lambeau, où la forme et la flexibilité diffèrent entre les enzymes BACE-X[21,172]. Dans l'ensemble, malgré la similitude revendiquée entre les protéines BACE-X, notre analyse révèle que les CMOs les mieux classés interagissant avec elles sont positionnés différemment, non seulement dans le site actif, mais aussi dans d'autres sous-sites protéiques cruciaux pour l'ancrage moléculaire.

Conclusions

1. Le logiciel Multiple-Copy Simultaneous Search a été évalué pour l'ancrage des nucléotides sur un benchmark de 121 complexes protéiques. Différents modèles de solvant et de phosphate ont été testés pour optimiser le taux de réussite pour identifier les poses natives (puissance d'amarrage) et le nucléotide natif réel (puissance de criblage). En conséquence, le modèle STDW combiné avec le patch phosphate R310 semble donner les meilleures performances, surpassant plusieurs fonctions de score. La présence de molécules d'eau dans la préparation et l'optimisation de la structure protéique permet à la structure minimisée de s'écarter moins de la structure expérimentale.

2. Quatre algorithmes de clustering populaires ont été significativement optimisés pour permettre leur application à des phases distinctes de l'Fragment-Based Drug Design. Grâce au stockage binaire, la traduction binaire des opérations primaires,

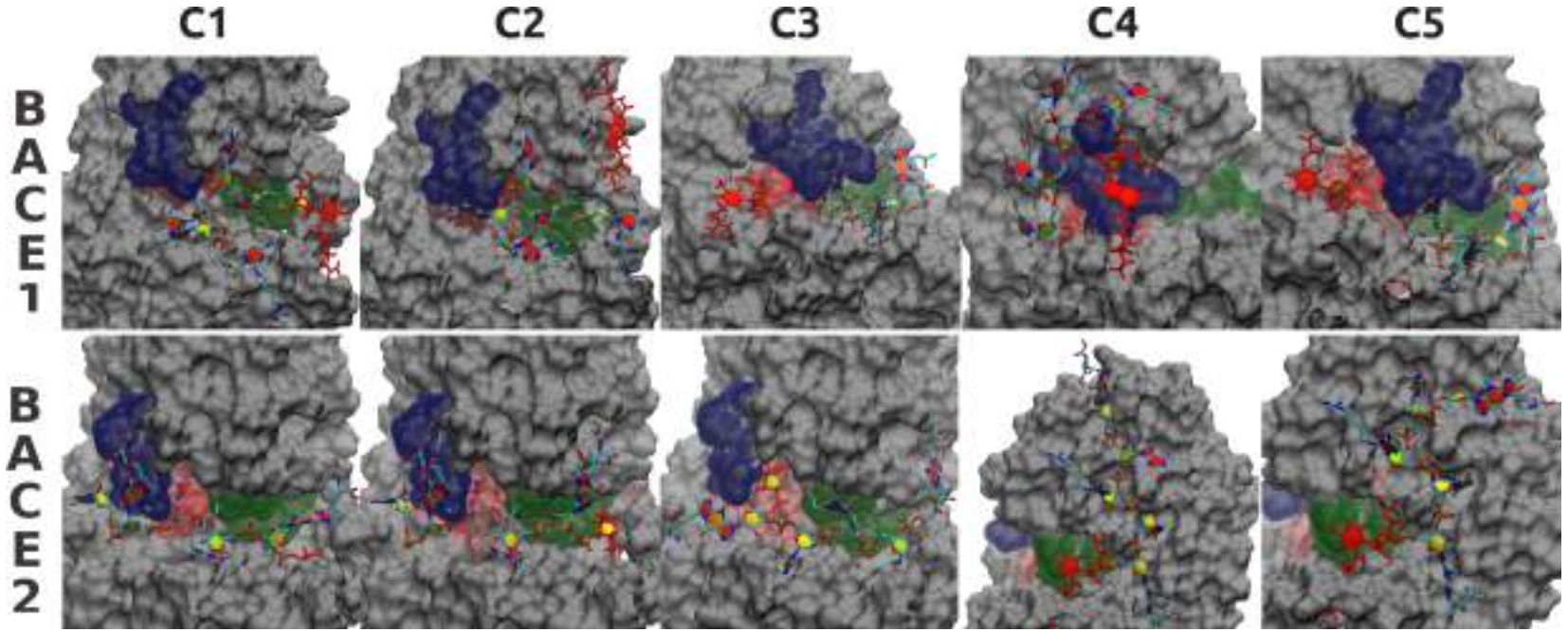


Figure 9.22: *Les cinq premiers clusters des inhibiteurs sélectifs potentiels de BACE-X. Les régions rouges, bleues et vertes correspondent respectivement au site actif, au "flap" et aux régions de boucle 10s. Les oligonucléotides sont représentés sans protons et les atomes de phosphore ont été mis en évidence pour une meilleure clarté visuelle. Le nucléotide rouge démarre l'oligomère dont la séquence est présentée dans le tableau 9.12.*

ou la reformulation complète des algorithmes, les versions exactes (BitClust, DP+) et modifiées (BitQT, RCDPeaks, MDSCAN) de la proposition originale ont été présentées et soigneusement comparées. Une confusion méthodologique entre Quality Threshold et les algorithmes de Daura a été exposée à la communauté.

3. Un liant informatique efficace pour l'assemblage d'Chemically Modified Oligonucleotides (fragments) sur des oligochaînes a été développé. Notre NUCLEotide AssembleR a été capable de renvoyer des séquences sans conflits geometriques suivant des contraintes distinctes (en séquence ou région d'exploration) et a pu identifier (malgré des limitations inhérentes) plusieurs modes de liaison expérimentaux dans trois études de cas. Une autre fonctionnalité nécessaire de ce logiciel est la détermination des "hotspots" à la surface de la cible pour guider l'Fragment-Based Drug Design.

4. Nous avons conçu un flux de travail basé sur des fragments *in silico* qui produit des modes de liaison à faible énergie d'Chemically Modified Oligonucleotides (obtenus par l'amarrage NUCLEotide AssembleR après Multiple-Copy Simultaneous Search) démontrant une sélectivité structurelle contre l'enzyme BACE1 sur les BACE2. Les modes de liaison BACE1 les mieux classés peuvent traverser linéairement la région active du site ou interagir simultanément avec le côté supérieur du volet et le site actif, tandis que des modes de liaison similaires ne sont pas détectés pour BACE2.

Table 9.12: *Descripteurs d'inhibiteurs sélectifs potentiels de BACE-X. La colonne Selectivity fait référence à $S_i(B1)$ ou $S_i(B2)$ pour BACE1 ou BACE2, respectivement.*

Protein	Clust. ID	Clust. size	Chain	EINT	Rank	#contacts	TI	RMSD	Selectivity
				[kcal/mol]				*[Å]*	
BACE1	1	105	R2C-OAU-1MA-OAU-3MC-R2C	-177.46	13	24	0.82	1.00	1.00
	2	268	HWG-R2C-OAU-1MG-T6A-SIA	-170.97	11	29	0.90	0.97	1.00
	3	119	13P-70U-OAU-BUG-6IA-HWG	-165.99	22	31	0.88	0.92	1.00
	4	160	HNA-OAU-K2C-HCU-OAU-GUA	-165.77	3	23	0.88	1.00	1.00
	5	36	OAU-70U-OAU-BUG-MMA-HWG	-163.89	43	28	0.87	0.88	1.00
BACE2	1	16	3MC-YYG-K2C-3AU-PBG-OMG	-180.66	10	31	0.77	0.98	1.00
	2	11	5CU-PBG-K2C-3AU-PBG-OMG	-178.5	18	34	0.78	0.99	1.00
	3	36	ADE-BUG-5HU-5CU-MAU-BUG	-168.2	174	28	0.71	0.98	1.00
	4	8	R2C-BUG-MSU-R2C-ADE-HWG	-159.84	20	27	0.83	1.00	1.00
	5	34	R2C-BUG-THY-R2C-5HU-PBG	-159.32	22	27	0.93	0.93	1.00

10 - Bibliography

[1] Berdigaliyev, N. and Aljofan, M. *An overview of drug discovery and development*. *Future Medicinal Chemistry*, 12(10):939–947, 2020. doi: 10.4155/fmc-2019-0307.

[2] Hughes, J. P., Rees, S. S., Kalindjian, S. B., and Philpott, K. L. *Principles of early drug discovery*. *British Journal of Pharmacology*, 162(6):1239–1249, 2011. doi: 10.1111/j.1476-5381.2010.01127.x.

[3] Mariyam, I. *A Short Note on Molecular Drug Targets and its Types Abstract Species-Specific Genes as Drug Targets*. pages 1–3, 2022.

[4] NIH. *Off-target effect*.

[5] Baker, M. *Fragment-based lead discovery grows up. Nature reviews. Drug discovery*, 12(1):5–7, 2013. doi: 10.1038/nrd3926.

[6] Shuker, S. B., Hajduk, P. J., Meadows, R. P., and Fesik, S. W. *Discovering high-affinity ligands for proteins: SAR by NMR*. *Science*, 274(5292):1531–1534, 1996. doi: 10.1126/science.274.5292.1531.

[7] Hajduk, P. J. and Greer, J. *A decade of fragment-based drug design: Strategic advances and lessons learned*. *Nature Reviews Drug Discovery*, 6(3):211–219, 2007. doi: 10.1038/nrd2220.

[8] Murray, C. W. and Rees, D. C. *The rise of fragment-based drug discovery*. *Nature Chemistry*, 1(3):187–192, 2009. doi: 10.1038/nchem.217.

[9] Price, A. J., Howard, S., and Cons, B. D. *Fragment-based drug discovery and its application to challenging drug targets*. *Essays in Biochemistry*, 61(5):475–484, 2017. doi: 10.1042/EBC20170029.

[10] Rinaldi, A. *RNA to the rescue* . *EMBO reports*, 21(7):e51013, 2020. doi: 10.15252/embr.202051013.

[11] Kaur, H., Bruno, J. G., Kumar, A., and Sharma, T. K. *Aptamers in the therapeutics and diagnostics pipelines*. *Theranostics*, 8(15):4016–4032, 2018. doi: 10.7150/thno.25958.

[12] Zhang, Y., Lai, B. S., and Juhas, M. *Recent advances in aptamer discovery and applications*. *Molecules*, 24(5):941, 2019. doi: 10.3390/molecules24050941.

[13] Kontoyianni, M. *Docking and virtual screening in drug discovery*. *Methods in Molecular Biology*, 1647:255–266, 2017. doi: 10.1007/978-1-4939-7201-2_18.

[14] Miranker, A. and Karplus, M. *Functionality maps of binding sites: A multiple copy simultaneous search method*. *Proteins: Structure, Function, and Bioinformatics*, 11(1):29–34, 1991. doi: 10.1002/prot.340110104.

[15] Eisen, M. B., Wiley, D. C., Karplus, M., and Hubbard, R. E. *HOOK: A program for finding novel molecular architectures that satisfy the chemical and steric requirements of a macromolecule binding site*. *Proteins: Structure, Function, and Bioinformatics*, 19(3):199–221, 1994. doi: 10.1002/prot.340190305.

[16] Stultz, C. M. and Karplus, M. *Dynamic ligand design and combinatorial optimization: Designing inhibitors to endothiapepsin*. *Proteins: Structure, Function and Genetics*, 40(2):258–289, 2000. doi: 10.1002/(SICI)1097-0134(20000801)40:2<258::AID-PROT80>3.0.CO;2-I.

[17] Takano, Y., Koizumi, M., Takarada, R., Kamimura, M. T., Czerminski, R., and Koike, T. *Computer-aided design of a factor Xa inhibitor by using MCSS functionality maps and a CAVEAT linker search*. *Journal of Molecular Graphics and Modelling*, 22(2):105–114, 2003. doi: 10.1016/S1093-3263(03)00140-2.

[18] So, S. S. and Karplus, M. *Evaluation of designed ligands by a multiple screening method: Application to glycogen phosphorylase inhibitors constructed with a variety of approaches*. *Journal of Computer-Aided Molecular Design*, 15(7):613–647, 2001. doi: 10.1023/A:1011945119287.

[19] Huang, Z., Zhang, M., Burton, S. D., Katsakhyan, L. N., and Ji, H. *Targeting the Tcf4 G 13 ANDE 17 Binding Site To Selectively Disrupt β-Catenin/T-Cell Factor Protein–Protein Interactions* . *ACS Chemical Biology*, 9(1):193–201, 2014. doi: 10.1021/cb400795x.

[20] Sammut, C. and Webb, G. I., editors. *Encyclopedia of Machine Learning*. Springer US, Boston, MA, 2010. doi: 10.1007/978-0-387-30164-8.

[21] Mirsafian, H., Ripen, A. M., Merican, A. F., and Mohamad, S. B. *Amino Acid Sequence and Structural Comparison of BACE1 and BACE2 Using Evolutionary Trace Method*. *Scientific World Journal*, 2014:1–6, 2014. doi: 10.1155/2014/482463.

[22] Blay, V., Tolani, B., Ho, S. P., and Arkin, M. R. *High-Throughput Screening: today's biochemical and cell-based approaches*. *Drug Discovery Today*, 25(10):1807–1821, 2020. doi: 10.1016/j.drudis.2020.07.024.

[23] Erlanson, D. A., Fesik, S. W., Hubbard, R. E., Jahnke, W., and Jhoti, H. *Twenty years on: The impact of fragments on drug discovery*. *Nature Reviews Drug Discovery*, 15(9):605–619, 2016. doi: 10.1038/nrd.2016.109.

[24] Reymond, J. L. *The Chemical Space Project*. *Accounts of Chemical Research*, 48(3):722–730, 2015. doi: 10.1021/ar500432k.

[25] Brown, D. G. and Boström, J. *Where Do Recent Small Molecule Clinical Development Candidates Come From?* *Journal of Medicinal Chemistry*, 61(21):9442–9468, 2018. doi: 10.1021/acs.jmedchem.8b00675.

[26] Yu, H. S., Modugula, K., Ichihara, O., Kramschuster, K., Keng, S., Abel, R., and Wang, L. *General Theory of Fragment Linking in Molecular Design: Why Fragment Linking Rarely Succeeds and How to Improve Outcomes*. *Journal of Chemical Theory and Computation*, 17(1):450–462, 2021. doi: 10.1021/acs.jctc.0c01004.

[27] Lin, X., Li, X., and Lin, X. *A review on applications of computational methods in drug screening and design*. *Molecules*, 25(6):1–17, 2020. doi: 10.3390/molecules25061375.

[28] Bian, Y. and Xie, X. Q. S. *Computational Fragment-Based Drug Design: Current Trends, Strategies, and Applications*. *AAPS Journal*, 20(3):59, 2018. doi: 10.1208/s12248-018-0216-7.

[29] Huang, N., Shoichet, B. K., and Irwin, J. J. *Benchmarking sets for molecular docking*. *Journal of Medicinal Chemistry*, 49(23):6789–6801, 2006. doi: 10.1021/jm0608356.

[30] Kumar, A., Voet, A., and Zhang, K. *Fragment Based Drug Design: From Experimental to Computational Approaches*. *Current Medicinal Chemistry*, 19(30):5128–5147, 2012. doi: 10.2174/092986712803530467.

[31] Klon, A. E. *Fragment-based methods in drug discovery*, volume 1289 of *Methods in Molecular Biology*. Springer New York, New York, NY, 2015. doi: 10.1007/978-1-4939-2486-8.

[32] Gaillard, T. *Evaluation of AutoDock and AutoDock Vina on the CASF-2013 Benchmark*. *Journal of Chemical Information and Modeling*, 58(8):1697–1706, 2018. doi: 10.1021/acs.jcim.8b00312.

[33] Jacquemard, C. and Kellenberger, E. *A bright future for fragment-based drug discovery: what does it hold?* *Expert Opinion on Drug Discovery*, 14(5):413–416, 2019. doi: 10.1080/17460441.2019.1583643.

[34] Wood, E. J. *An Introduction to Medicinal Chemistry (Third Edition)*, volume 6. Oxford University Press, United Kingdom, 4th edition, 2005. doi: 10.3108/beej.2005.06000007.

[35] Rasul, A., Riaz, A., Sarfraz, I., Khan, S. G., Hussain, G., Zara, R., Sadiqa, A., Bushra, G., Riaz, S., Iqbal, M. J., Hassan, M., and Khorsandi, K. *Target*

Identification Approaches in Drug Discovery. August, pages 41–59. 2022. doi: 10.1007/978-3-030-95895-4_3.

[36] Visscher, P. M., Wray, N. R., Zhang, Q., Sklar, P., McCarthy, M. I., Brown, M. A., and Yang, J. *10 Years of GWAS Discovery: Biology, Function, and Translation*. *American Journal of Human Genetics*, 101(1):5–22, 2017. doi: 10.1016/j.ajhg.2017.06.005.

[37] Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. *Quantitative monitoring of gene expression patterns with a complementary DNA microarray*. *Science*, 270(5235):467–470, 1995. doi: 10.1126/science.270.5235.467.

[38] Hood, B. L., Veenstra, T. D., and Conrads, T. P. *Mass spectrometry-based proteomics*. *International Congress Series*, 1266(C):375–380, 2004. doi: 10.1016/j.ics.2004.02.087.

[39] Kitano, H. *Systems biology: A brief overview*. *Science*, 295(5560):1662–1664, 2002. doi: 10.1126/science.1069492.

[40] Hu, Y., Zhao, T., Zhang, N., Zhang, Y., and Cheng, L. *A Review of Recent Advances and Research on Drug Target Identification Methods*. *Current Drug Metabolism*, 20(3):209–216, 2018. doi: 10.2174/1389200219666180925091851.

[41] Congreve, M., Carr, R., Murray, C., and Jhoti, H. *A'rule of three'for fragment-based lead discovery? Drug discovery today*, 8(19):876–877, 2003.

[42] Köster, H., Craan, T., Brass, S., Herhaus, C., Zentgraf, M., Neumann, L., Heine, A., and Klebe, G. *A small nonrule of 3 compatible fragment library provides high hit rate of endothiapepsin crystal structures with various fragment chemotypes*. *Journal of Medicinal Chemistry*, 54(22):7784–7796, 2011. doi: 10.1021/jm200642w.

[43] Shi, Y. and von Itzstein, M. *How size matters: Diversity for fragment library design*. *Molecules*, 24(15):2838, 2019. doi: 10.3390/molecules24152838.

[44] Bon, M., Bilsland, A., Bower, J., and McAulay, K. *Fragment-based drug discovery—the importance of high-quality molecule libraries*. *Molecular Oncology*, 16(21):3761–3777, 2022. doi: 10.1002/1878-0261.13277.

[45] Heidrich, J., Sperl, L. E., and Boeckler, F. M. *Embracing the diversity of halogen bonding motifs in fragment-based drug discovery-construction of a diversity-optimized halogen-enriched fragment library*. *Frontiers in Chemistry*, 7(FEB), 2019. doi: 10.3389/fchem.2019.00009.

[46] Garner, P., Cox, P. B., Rathnayake, U., Holloran, N., and Erdman, P. *Design and Synthesis of Pyrrolidine-based Fragments That Sample Three-dimensional Molecular Space*. *ACS Medicinal Chemistry Letters*, 10(5):811–815, 2019. doi: 10.1021/acsmedchemlett.9b00070.

[47] Liu, M. and Quinn, R. J. *Fragment-based screening with natural products for novel anti-parasitic disease drug discovery*. *Expert Opinion on Drug Discovery*, 14(12):1283–1295, 2019. doi: 10.1080/17460441.2019.1653849.

[48] Kutchukian, P. S., So, S. S., Fischer, C., and Waller, C. L. *Fragment library design: Using cheminformatics and expert chemists to fill gaps in existing fragment libraries*. In *Methods in Molecular Biology*, volume 1289, pages 43–53. Springer New York, 2015. doi: 10.1007/978-1-4939-2486-8_5.

[49] Li, Q. *Application of Fragment-Based Drug Discovery to Versatile Targets*. *Frontiers in Molecular Biosciences*, 7(August):1–13, 2020. doi: 10.3389/fmolb.2020.00180.

[50] Carbery, A., Skyner, R., Von Delft, F., and Deane, C. M. *Fragment Libraries Designed to Be Functionally Diverse Recover Protein Binding Information More Efficiently Than Standard Structurally Diverse Libraries*. *Journal of Medicinal Chemistry*, 65(16):11404–11413, 2022. doi: 10.1021/acs.jmedchem.2c01004.

[51] Thakore, S. D., Akhtar, J., Jain, R., Paudel, A., and Bansal, A. K. *Analytical and Computational Methods for the Determination of Drug-Polymer Solubility and Miscibility*. *Molecular Pharmaceutics*, 18(8):2835–2866, 2021. doi: 10.1021/acs.molpharmaceut.1c00141.

[52] Faller, B. and Ertl, P. *Computational approaches to determine drug solubility*. *Advanced Drug Delivery Reviews*, 59(7):533–545, 2007. doi: 10.1016/j.addr.2007.05.005.

[53] Kirsch, P., Hartman, A. M., Hirsch, A. K., and Empting, M. *Concepts and core principles of fragment-based drug design*. *Molecules*, 24(23), 2019. doi: 10.3390/molecules24234309.

[54] Kuntz, I. D., Blaney, J. M., Oatley, S. J., Langridge, R., and Ferrin, T. E. *A geometric approach to macromolecule-ligand interactions*. *Journal of Molecular Biology*, 161(2):269–288, 1982. doi: 10.1016/0022-2836(82)90153-X.

[55] Hann, M. M. and Keserr, G. M. *Finding the sweet spot: The role of nature and nurture in medicinal chemistry*. *Nature Reviews Drug Discovery*, 11(5):355–365, 2012. doi: 10.1038/nrd3701.

[56] De Esch, I. J., Erlanson, D. A., Jahnke, W., Johnson, C. N., and Walsh, L. *Fragment-to-Lead Medicinal Chemistry Publications in 2020*. *Journal of Medicinal Chemistry*, 65(1):84–99, 2022. doi: 10.1021/acs.jmedchem.1c01803.

[57] Hopkins, A. L., Keserü, G. M., Leeson, P. D., Rees, D. C., and Reynolds, C. H. *The role of ligand efficiency metrics in drug discovery*. *Nature Reviews Drug Discovery*, 13(2):105–121, 2014. doi: 10.1038/nrd4163.

[58] Bancet, A., Raingeval, C., Lomberget, T., Le Borgne, M., Guichou, J. F., and Krimm, I. *Fragment Linking Strategies for Structure-Based Drug Design*. *Journal of Medicinal Chemistry*, 63(20):11420–11435, 2020. doi: 10.1021/acs.jmedchem.0c00242.

[59] Zoete, V., Grosdidier, A., and Michielin, O. *Docking, virtual high throughput screening and in silico fragment-based drug design*. *Journal of Cellular and Molecular Medicine*, 13(2):238–248, 2009. doi: 10.1111/j.1582-4934.2008.00665.x.

[60] Schneider, G. and Fechner, U. *Computer-based de novo design of drug-like molecules*. *Nature Reviews Drug Discovery*, 4(8):649–663, 2005. doi: 10.1038/nrd1799.

[61] Kawai, K., Nagata, N., and Takahashi, Y. *De novo design of drug-like molecules by a fragment-based molecular evolutionary approach*. *Journal of Chemical Information and Modeling*, 54(1):49–56, 2014. doi: 10.1021/ci400418c.

[62] Lipinski, C. A. *Drug-like properties and the causes of poor solubility and poor permeability*. *Journal of Pharmacological and Toxicological Methods*, 44(1):235–249, 2000. doi: 10.1016/S1056-8719(00)00107-6.

[63] Gohlke, H. and Klebe, G. *Approaches to the description and prediction of the binding affinity of small-molecule ligands to macromolecular receptors*. *Angewandte Chemie - International Edition*, 41(15):2644–2676, 2002. doi: 10.1002/1521-3773(20020802)41:15<2644::AID-ANIE2644>3.0.CO;2-O.

[64] Kitchen, D. B., Decornez, H., Furr, J. R., and Bajorath, J. *Docking and scoring in virtual screening for drug discovery: Methods and applications*. *Nature Reviews Drug Discovery*, 3(11):935–949, 2004. doi: 10.1038/nrd1549.

[65] Dixon, J. S. *Evaluation of the CASP2 docking section*. *Proteins: Structure, Function and Genetics*, 29(SUPPL. 1):198–204, 1997. doi: 10.1002/(SICI)1097-0134(1997)1+<198::AID-PROT26>3.0.CO;2-I.

[66] Carlson, H. A. and McCammon, J. A. *Accommodating protein flexibility in computational drug design*. *Molecular Pharmacology*, 57(2):213–218, 2000.

[67] Su, M., Yang, Q., Du, Y., Feng, G., Liu, Z., Li, Y., and Wang, R. *Comparative Assessment of Scoring Functions: The CASF-2016 Update*. *Journal of Chemical Information and Modeling*, 59(2):895–913, 2019. doi: 10.1021/acs.jcim.8b00545.

[68] Kramer, B., Rarey, M., and Lengauer, T. *Evaluation of the FlexX incremental construction algorithm for protein- ligand docking*. *Proteins: Structure, Function and Genetics*, 37(2):228–241, 1999. doi: 10.1002/(SICI)1097-0134(19991101)37:2<228::AID-PROT8>3.0.CO;2-8.

[69] Yang, Y., Lightstone, F. C., and Wong, S. E. *Approaches to efficiently estimate solvation and explicit water energetics in ligand binding: The use of WaterMap*. *Expert Opinion on Drug Discovery*, 8(3):277–287, 2013. doi: 10.1517/17460441.2013.749853.

[70] Chen, F., Liu, H., Sun, H., Pan, P., Li, Y., Li, D., and Hou, T. *Assessing the performance of the MM/PBSA and MM/GBSA methods. 6. Capability to predict protein-protein binding free energies and re-rank binding poses generated by protein-protein docking*. *Physical Chemistry Chemical Physics*, 18(32):22129–22139, 2016. doi: 10.1039/c6cp03670h.

[71] Kulik, H. J. *Large-scale QM/MM free energy simulations of enzyme catalysis reveal the influence of charge transfer*. *Physical Chemistry Chemical Physics*, 20(31):20650–20660, 2018. doi: 10.1039/c8cp03871f.

[72] Orozco-Gonzalez, Y., Manathunga, M., Marín, M. D. C., Agathangelou, D., Jung, K. H., Melaccio, F., Ferré, N., Haacke, S., Coutinho, K., Canuto, S., and Olivucci, M. *An Average Solvent Electrostatic Configuration Protocol for QM/MM Free Energy Optimization: Implementation and Application to Rhodopsin Systems*. *Journal of Chemical Theory and Computation*, 13(12):6391–6404, 2017. doi: 10.1021/acs.jctc.7b00860.

[73] Chaskar, P., Zoete, V., and Röhrig, U. F. *Toward on-the-fly quantum mechanical/molecular mechanical (QM/MM) docking: Development and benchmark of a scoring function*. *Journal of Chemical Information and Modeling*, 54(11):3137–3152, 2014. doi: 10.1021/ci5004152.

[74] Pason, L. P. and Sotriffer, C. A. *Empirical Scoring Functions for Affinity Prediction of Protein-ligand Complexes*. *Molecular Informatics*, 35(11-12):541–548, 2016. doi: 10.1002/minf.201600048.

[75] Liu, J. and Wang, R. *Classification of current scoring functions*. *Journal of Chemical Information and Modeling*, 55(3):475–482, 2015. doi: 10.1021/ci500731a.

[76] Ain, Q. U., Aleksandrova, A., Roessler, F. D., and Ballester, P. J. *Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening*. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 5(6):405–424, 2015. doi: 10.1002/wcms.1225.

[77] Li, Y., Liu, Z., Li, J., Han, L., Liu, J., Zhao, Z., and Wang, R. *Comparative assessment of scoring functions on an updated benchmark: 1. compilation of the test set*. *Journal of Chemical Information and Modeling*, 54(6):1700–1716, 2014. doi: 10.1021/ci500080q.

[78] Quiroga, R. and Villarreal, M. A. *Vinardo: A scoring function based on autodock vina improves scoring, docking, and virtual screening*. *PLoS ONE*, 11(5):e0155183—–18, 2016. doi: 10.1371/journal.pone.0155183.

[79] Muegge, I. and Martin, Y. C. *A general and fast scoring function for protein-ligand interactions: A simplified potential approach*. *Journal of Medicinal Chemistry*, 42(5):791–804, 1999. doi: 10.1021/jm980536j.

[80] Gohlke, H., Hendlich, M., and Klebe, G. *Knowledge-based scoring function to predict protein-ligand interactions*. *Journal of Molecular Biology*, 295(2):337–356, 2000. doi: 10.1006/jmbi.1999.3371.

[81] Zheng, Z. and Merz, K. M. *Development of the knowledge-based and empirical combined scoring algorithm (KECSA) to score protein-ligand interactions*. *Journal of Chemical Information and Modeling*, 53(5):1073–1083, 2013. doi: 10.1021/ci300619x.

[82] Velec, H. F., Gohlke, H., and Klebe, G. *DrugScoreCSD-knowledge-based scoring function derived from small molecule crystal data with superior recognition rate of near-native ligand poses and better affinity prediction*. *Journal of Medicinal Chemistry*, 48(20):6296–6303, 2005. doi: 10.1021/jm050436v.

[83] Neudert, G. and Klebe, G. *DSX: A knowledge-based scoring function for the assessment of protein-ligand complexes*. *Journal of Chemical Information and Modeling*, 51(10):2731–2745, 2011. doi: 10.1021/ci200274q.

[84] Yang, C. Y., Wang, R., and Wang, S. *M-score: A knowledge-based potential scoring function accounting for protein atom mobility*. *Journal of Medicinal Chemistry*, 49(20):5903–5911, 2006. doi: 10.1021/jm050043w.

[85] Huang, S. Y. and Zou, X. *A knowledge-based scoring function for protein-RNA interactions derived from a statistical mechanics-based iterative method*. *Nucleic Acids Research*, 42(7):e55–e55, 2014. doi: 10.1093/nar/gku077.

[86] Shen, C., Ding, J., Wang, Z., Cao, D., Ding, X., and Hou, T. *From machine learning to deep learning: Advances in scoring functions for protein–ligand docking*. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 10(1):e1429, 2020. doi: 10.1002/wcms.1429.

[87] Khamis, M. A., Gomaa, W., and Ahmed, W. F. *Machine learning in computational docking*. *Artificial Intelligence in Medicine*, 63(3):135–152, 2015. doi: 10.1016/j.artmed.2015.02.002.

[88] Zhang, L., Ai, H. X., Li, S. M., Qi, M. Y., Zhao, J., Zhao, Q., and Liu, H. S. *Virtual screening approach to identifying influenza virus neuraminidase inhibitors*

using molecular docking combined with machine-learning-based scoring function. *Oncotarget*, 8(47):83142–83154, 2017. doi: 10.18632/oncotarget.20915.

[89] Cheng, T., Li, Q., Zhou, Z., Wang, Y., and Bryant, S. H. *Structure-based virtual screening for drug discovery: A problem-centric review*. *AAPS Journal*, 14(1):133–141, 2012. doi: 10.1208/s12248-012-9322-0.

[90] Ma, D. L., Chan, D. S. H., and Leung, C. H. *Drug repositioning by structure-based virtual screening*. *Chemical Society Reviews*, 42(5):2130–2141, 2013. doi: 10.1039/c2cs35357a.

[91] Wang, C. and Zhang, Y. *Improving scoring-docking-screening powers of protein–ligand scoring functions using random forest*. *Journal of Computational Chemistry*, 38(3):169–177, 2017. doi: 10.1002/jcc.24667.

[92] Charifson, P. S., Corkery, J. J., Murcko, M. A., and Walters, W. P. *Consensus scoring: A method for obtaining improved hit rates from docking databases of three-dimensional structures into proteins*. *Journal of Medicinal Chemistry*, 42(25):5100–5109, 1999. doi: 10.1021/jm990352k.

[93] Bissantz, C., Folkers, G., and Rognan, D. *Protein-based virtual screening of chemical databases. 1. Evaluation of different docking/scoring combinations*. *Journal of Medicinal Chemistry*, 43(25):4759–4767, 2000. doi: 10.1021/jm001044l.

[94] Chaput, L. and Mouawad, L. *Efficient conformational sampling and weak scoring in docking programs? Strategy of the wisdom of crowds*. *Journal of Cheminformatics*, 9(1):37, 2017. doi: 10.1186/s13321-017-0227-x.

[95] Ericksen, S. S., Wu, H., Zhang, H., Michael, L. A., Newton, M. A., Hoffmann, F. M., and Wildman, S. A. *Machine Learning Consensus Scoring Improves Performance Across Targets in Structure-Based Virtual Screening*. *Journal of Chemical Information and Modeling*, 57(7):1579–1590, 2017. doi: 10.1021/acs.jcim.7b00153.

[96] Terp, G. E., Johansen, B. N., Christensen, I. T., and Jørgensen, F. S. *A new concept for multidimensional selection of ligand conformations (multiselect) and multidimensional scoring (multiscore) of protein-ligand binding affinities*. *Journal of Medicinal Chemistry*, 44(14):2333–2343, 2001. doi: 10.1021/jm001090l.

[97] Betzi, S., Suhre, K., Chétrit, B., Guerlesquin, F., and Morelli, X. *GFscore: A general nonlinear consensus scoring function for high-throughput docking*. *Journal of Chemical Information and Modeling*, 46(4):1704–1712, 2006. doi: 10.1021/ci0600758.

[98] Bar-Haim, S., Aharon, A., Ben-Moshe, T., Marantz, Y., and Senderowitz, H. *SeleX-CS: A new consensus scoring algorithm for hit discovery and lead optimization*.

Journal of Chemical Information and Modeling, 49(3):623–633, 2009. doi: 10.1021/ci800335j.

[99] Plewczynski, D., Łaźniewski, M., Grotthuss, M. V., Rychlewski, L., and Ginalski, K. *VoteDock: Consensus docking method for prediction of protein-ligand interactions*. *Journal of Computational Chemistry*, 32(4):568–581, 2011. doi: 10.1002/jcc.21642.

[100] Elbert, R. and Karplus, M. *Enhanced Sampling in Molecular Dynamics: Use of the Time-Dependent Hartree Approximation for a Simulation of Carbon Monoxide Diffusion through Myoglobint*. *Journal of the American Chemical Society*, 112(25):9161–9175, 1990. doi: 10.1021/ja00181a020.

[101] Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S., and Karplus, M. *CHARMM: A program for macromolecular energy, minimization, and dynamics calculations*. *Journal of Computational Chemistry*, 4(2):187–217, 1983. doi: 10.1002/jcc.540040211.

[102] Caflisch, A., Miranker, A., and Karplus, M. *Multiple Copy Simultaneous Search and Construction of Ligands in Binding Sites: Application to Inhibitors of HIV-1 Aspartic Proteinase*. *Journal of Medicinal Chemistry*, 36(15):2142–2167, 1993. doi: 10.1021/jm00067a013.

[103] Joseph-McCarthy, D., Fedorov, A. A., and Almo, S. C. *Comparison of experimental and computational functional group mapping of an RNase A structure: Implications for computer-aided drug design*. *Protein Engineering*, 9(9):773–780, 1996. doi: 10.1093/protein/9.9.773.

[104] Joseph-McCarthy, D., Tsang, S. K., Filman, D. J., Hogle, J. M., and Karplus, M. *Use of MCSS to design small targeted libraries: Application to picornavirus ligands*. *Journal of the American Chemical Society*, 123(51):12758–12769, 2001. doi: 10.1021/ja003972f.

[105] Singh, J., Saldanha, J., and Thornton, J. M. *A novel method for the modelling of peptide ligands to their receptors*. *Protein Engineering, Design and Selection*, 4(3):251–261, 1991. doi: 10.1093/protein/4.3.251.

[106] Elkin, C. D., Zuccola, H. J., Hogle, J. M., and Joseph-McCarthy, D. *Computational design of D-peptide inhibitors of hepatitis delta antigen dimerization*. *Journal of Computer-Aided Molecular Design*, 14(8):705–718, 2000. doi: 10.1023/A:1008146015629.

[107] Zeng, J., Nheu, T., Zorzet, A., Catimel, B., Nice, E., Maruta, H., Burgess, A. W., and Treutlein, H. R. *Design of inhibitors of Ras-Raf interaction using a computational combinatorial algorithm*. *Protein Engineering*, 14(1):39–45, 2001. doi: 10.1093/protein/14.1.39.

[108] Leclerc, F. and Karplus, M. *MCSS-based predictions of RNA binding sites*. *Theoretical Chemistry Accounts*, 101(1-3):131–137, 1999. doi: 10.1007/s002140050419.

[109] Caflisch, A., Schramm, H. J., and Karplus, M. *Design of dimerization inhibitors of HIV-1 aspartic proteinase: A computer-based combinatorial approach*. *Journal of Computer-Aided Molecular Design*, 14(2):161–179, 2000. doi: 10.1023/A:1008146201260.

[110] Haider, M. K., Bertrand, H. O., and Hubbard, R. E. *Predicting fragment binding poses using a combined MCSS MM-GBSA approach*. *Journal of Chemical Information and Modeling*, 51(5):1092–1105, 2011. doi: 10.1021/ci100469n.

[111] Haider, K. and Huggins, D. J. *Combining solvent thermodynamic profiles with functionality maps of the Hsp90 binding site to predict the displacement of water molecules*. *Journal of Chemical Information and Modeling*, 53(10):2571–2586, 2013. doi: 10.1021/ci4003409.

[112] Onufriev, A. V. and Case, D. A. *Generalized Born Implicit Solvent Models for Biomolecules*. *Annual Review of Biophysics*, 48(1):275–296, 2019. doi: 10.1146/annurev-biophys-052118-115325.

[113] Tidor, B., Irikura, K. K., Brooks, B. R., and Karplus, M. *Dynamics of dna oligomers*. *Journal of Biomolecular Structure and Dynamics*, 1(1):231–252, 1983. doi: 10.1080/07391102.1983.10507437.

[114] MacKerell, A. D., Banavali, N., and Foloppe, N. *Development and current status of the CHARMM force field for nucleic acids*. *Biopolymers*, 56(4):257–265, 2000. doi: 10.1002/1097-0282(2000)56:4<257::AID-BIP10029>3.0.CO;2-W.

[115] Sullenger, B. A. and Nair, S. *From the RNAworld to the clinic*. *Science*, 352(6292):1417–1420, 2016. doi: 10.1126/science.aad8709.

[116] Wang, F., Zuroske, T., and Watts, J. K. *RNA therapeutics on the rise*. *Nature reviews. Drug discovery*, 19(7):441–442, 2020. doi: 10.1038/d41573-020-00078-0.

[117] Zhu, S., Rooney, S., and Michlewski, G. *RNA-targeted therapies and high-throughput screening methods*. *International Journal of Molecular Sciences*, 21(8):2996, 2020. doi: 10.3390/ijms21082996.

[118] Siddiqui, M. A. A. and Keating, G. M. *Pegaptanib: In exudative age-related macular degeneration*. *Drugs*, 65(11):1571–1577, 2005. doi: 10.2165/00003495-200565110-00010.

[119] Ng, E. W. and Adamis, A. P. *Anti-VEGF aptamer (pegaptanib) therapy for ocular vascular diseases*. *Annals of the New York Academy of Sciences*, 1082(1):151–171, 2006. doi: 10.1196/annals.1348.062.

[120] Pardi, N., Hogan, M. J., Porter, F. W., and Weissman, D. *mRNA vaccines-a new era in vaccinology*. *Nature Reviews Drug Discovery*, 17(4):261–279, 2018. doi: 10.1038/nrd.2017.243.

[121] Walsh, E. E., Frenck, R. W., Falsey, A. R., Kitchin, N., Absalon, J., Gurtman, A., Lockhart, S., Neuzil, K., Mulligan, M. J., Bailey, R., Swanson, K. A., Li, P., Koury, K., Kalina, W., Cooper, D., Fontes-Garfias, C., Shi, P.-Y., Türeci, Ö., Tompkins, K. R., Lyke, K. E., Raabe, V., Dormitzer, P. R., Jansen, K. U., Şahin, U., and Gruber, W. C. *Safety and Immunogenicity of Two RNA-Based Covid-19 Vaccine Candidates*. *New England Journal of Medicine*, 383(25):2439–2450, 2020. doi: 10.1056/nejmoa2027906.

[122] Zhou, J. and Rossi, J. *Aptamers as targeted therapeutics: Current potential and challenges*. *Nature Reviews Drug Discovery*, 16(3):181–202, 2017. doi: 10.1038/nrd.2016.199.

[123] Katz, B. and Goldbaum, M. *Macugen (pegaptanib sodium), a novel ocular therapeutic that targets vascular endothelial growth factor (VEGF)*. *International Ophthalmology Clinics*, 46(4):141–154, 2006. doi: 10.1097/01.iio.0000212130.91136.31.

[124] Yazdian-Robati, R., Bayat, P., Oroojalian, F., Zargari, M., Ramezani, M., Taghdisi, S. M., and Abnous, K. *Therapeutic applications of AS1411 aptamer, an update review*. *International Journal of Biological Macromolecules*, 155:1420–1431, 2020. doi: 10.1016/j.ijbiomac.2019.11.118.

[125] Hoellenriegel, J., Zboralski, D., Maasch, C., Rosin, N. Y., Wierda, W. G., Keating, M. J., Kruschinski, A., and Burger, J. A. *The Spiegelmer NOX-A12, a novel CXCL12 inhibitor, interferes with chronic lymphocytic leukemia cell motility and causes chemosensitization*. *Blood*, 123(7):1032–1039, 2014. doi: 10.1182/blood-2013-03-493924.

[126] Oberthür, D., Achenbach, J., Gabdulkhakov, A., Buchner, K., Maasch, C., Falke, S., Rehders, D., Klussmann, S., and Betzel, C. *Crystal structure of a mirror-image L-RNA aptamer (Spiegelmer) in complex with the natural L-protein target CCL2*. *Nature Communications*, 6(1):6923, 2015. doi: 10.1038/ncomms7923.

[127] Schwoebel, F., van Eijk, L. T., Zboralski, D., Sell, S., Buchner, K., Maasch, C., Purschke, W. G., Humphrey, M., Zöllner, S., Eulberg, D., Morich, F., Pickkers, P., and Klussmann, S. *The effects of the anti-hepcidin Spiegelmer NOX-H94 on inflammation-induced anemia in cynomolgus monkeys*. *Blood*, 121(12):2311–2315, 2013. doi: 10.1182/blood-2012-09-456756.

[128] DUERSCHMIED, D., MERHI, Y., TANGUAY, J., HUTABARAT, R., GILBERT, J., WAGNER, D., SCHAUB, R., DIENER, J., and DANIEL LAGASSÉ, H. *Inhibition*

of von Willebrand factor-mediated platelet activation and thrombosis by the anti-von Willebrand factor A1-domain aptamer ARC1779. *Journal of Thrombosis and Haemostasis*, 7(7):1155–1162, 2009. doi: 10.1111/j.1538-7836.2009.03459.x.

[129] Srivastava, S., Abraham, P. R., and Mukhopadhyay, S. *Aptamers: An Emerging Tool for Diagnosis and Therapeutics in Tuberculosis*. *Frontiers in Cellular and Infection Microbiology*, 11, 2021. doi: 10.3389/fcimb.2021.656421.

[130] Tuerk, C. and Gold, L. *Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase*. *Science*, 249(4968):505–510, 1990. doi: 10.1126/science.2200121.

[131] Ellington, A. D. and Szostak, J. W. *In vitro selection of RNA molecules that bind specific ligands*. *Nature*, 346(6287):818–822, 1990. doi: 10.1038/346818a0.

[132] O'Connell, D., Koenig, A., Jennings, S., Hicke, B., Han, H. L., Fitzwater, T., Chang, Y. F., Varki, N., Parma, D., and Varki, A. *Calcium-dependent oligonucleotide antagonists specific for L-selectin*. *Proceedings of the National Academy of Sciences of the United States of America*, 93(12):5883–5887, 1996. doi: 10.1073/pnas.93.12.5883.

[133] Wang, K., Wang, M., Ma, T., Li, W., and Zhang, H. *Review on the Selection of Aptamers and Application in Paper-Based Sensors*. *Biosensors*, 13(1):39, 2023. doi: 10.3390/bios13010039.

[134] Ni, S., Zhuo, Z., Pan, Y., Yu, Y., Li, F., Liu, J., Wang, L., Wu, X., Li, D., Wan, Y., Zhang, L., Yang, Z., Zhang, B. T., Lu, A., and Zhang, G. *Recent Progress in Aptamer Discoveries and Modifications for Therapeutic Applications*. *ACS Applied Materials and Interfaces*, 13(8):9500–9519, 2021. doi: 10.1021/acsami.0c05750.

[135] Padilla, R. and Sousa, R. *Efficient synthesis of nucleic acids heavily modified with non-canonical ribose 2'-groups using a mutant T7 RNA polymerase (RNAP)*. *Nucleic Acids Research*, 27(6):1561–1563, 1999. doi: 10.1093/nar/27.6.1561.

[136] Ruckman, J., Green, L. S., Beeson, J., Waugh, S., Gillette, W. L., Henninger, D. D., Claesson-Welsh, L., and Janjić, N. *2'-fluoropyrimidine RNA-based aptamers to the 165-amino acid form of vascular endothelial growth factor (VEGF165): Inhibition of receptor binding and VEGF-induced vascular permeability through interactions requiring the exon 7-encoded domain*. *Journal of Biological Chemistry*, 273(32):20556–20567, 1998. doi: 10.1074/jbc.273.32.20556.

[137] Barciszewski, J., Medgaard, M., Koch, T., Kurreck, J., and Erdmann, V. A. *Locked nucleic acid aptamers*. In *Methods in Molecular Biology*, volume 535, pages 165–186. Humana Press, 2009. doi: 10.1007/978-1-59745-557-2_10.

[138] Green, L. S., Jellinek, D., Bell, C., Beebe, L. A., Feistner, B. D., Gill, S. C., Jucker, F. M., and Janjić, N. *Nuclease-resistant nucleic acid ligands to vascular permeability factor/vascular endothelial growth factor*. *Chemistry and Biology*, 2(10):683–695, 1995. doi: 10.1016/1074-5521(95)90032-2.

[139] Jhaveri, S., Olwin, B., and Ellington, A. D. *In vitro selection of phosphorothiolated aptamers*. *Bioorganic and Medicinal Chemistry Letters*, 8(17):2285–2290, 1998. doi: 10.1016/S0960-894X(98)00414-4.

[140] King, D. J., Ventura, D. A., Brasier, A. R., and Gorenstein, D. G. *Novel combinatorial selection of phosphorothioate oligonucleotide aptamers*. *Biochemistry*, 37(47):16489–16493, 1998. doi: 10.1021/bi981780f.

[141] Yang, X. and Gorenstein, D. *Progress in Thioaptamer Development*. *Current Drug Targets*, 5(8):705–715, 2005. doi: 10.2174/1389450043345074.

[142] Abeydeera, N. D., Egli, M., Cox, N., Mercier, K., Conde, J. N., Pallan, P. S., Mizurini, D. M., Sierant, M., Hibti, F. E., Hassell, T., Wang, T., Liu, F. W., Liu, H. M., Martinez, C., Sood, A. K., Lybrand, T. P., Frydman, C., Monteiro, R. Q., Gomer, R. H., Nawrot, B., and Yang, X. *Evoking picomolar binding in RNA by a single phosphorodithioate linkage*. *Nucleic Acids Research*, 44(17):8052–8064, 2016. doi: 10.1093/nar/gkw725.

[143] Kimoto, M., Yamashige, R., Matsunaga, K. I., Yokoyama, S., and Hirao, I. *Generation of high-affinity DNA aptamers using an expanded genetic alphabet*. *Nature Biotechnology*, 31(5):453–457, 2013. doi: 10.1038/nbt.2556.

[144] Vater, A. and Klussmann, S. *Turning mirror-image oligonucleotides into drugs: The evolution of Spiegelmer® therapeutics*. *Drug Discovery Today*, 20(1):147–155, 2015. doi: 10.1016/j.drudis.2014.09.004.

[145] Ortigão, J. F. R., Rösch, H., Selter, H., Fröhlich, A., Lorenz, A., Montenarh, M., and Seliger, H. *Antisense Effect of Oligodeoxynucleotides with Inverted Terminal Internucleotidic Linkages: A Minimal Modification Protecting against Nucleolytic Degradation*. *Antisense Research and Development*, 2(2):129–146, 1992. doi: 10.1089/ard.1992.2.129.

[146] Di Giusto, D. A., Wlassoff, W. A., Gooding, J. J., Messerle, B. A., and King, G. C. *Proximity extension of circular DNA aptamers with real-time protein detection*. *Nucleic Acids Research*, 33(6):1–7, 2005. doi: 10.1093/nar/gni063.

[147] Guerchet, M., Prince, M., and Prina, M. *Numbers of people with dementia worldwide: An update to the estimates in the World Alzheimer Report 2015*. *International, Alzheimer's Disease*, 2020.

[148] Wimo, A., Guerchet, M., Ali, G. C., Wu, Y. T., Prina, A. M., Winblad, B., Jönsson, L., Liu, Z., and Prince, M. *The worldwide costs of dementia 2015 and comparisons with 2010. Alzheimer's and Dementia*, 13(1):1–7, 2017. doi: 10.1016/j.jalz.2016.07.150.

[149] Nichols, E., Steinmetz, J. D., Vollset, S. E., Fukutaki, K., Chalek, J., Abd-Allah, F., Abdoli, A., Abualhasan, A., Abu-Gharbieh, E., Akram, T. T., Al Hamad, H., Alahdab, F., Alanezi, F. M., Alipour, V., Almustanyir, S., Amu, H., Ansari, I., Arabloo, J., Ashraf, T., Astell-Burt, T., Ayano, G., Ayuso-Mateos, J. L., Baig, A. A., Barnett, A., Barrow, A., Baune, B. T., Béjot, Y., Bezabhe, W. M., Bezabih, Y. M., Bhagavathula, A. S., Bhaskar, S., Bhattacharyya, K., Bijani, A., Biswas, A., Bolla, S. R., Boloor, A., Brayne, C., Brenner, H., Burkart, K., Burns, R. A., Cámera, L. A., Cao, C., Carvalho, F., Castro-de Araujo, L. F., Catalá-López, F., Cerin, E., Chavan, P. P., Cherbuin, N., Chu, D. T., Costa, V. M., Couto, R. A., Dadras, O., Dai, X., Dandona, L., Dandona, R., De la Cruz-Góngora, V., Dhamnetiya, D., Dias da Silva, D., Diaz, D., Douiri, A., Edvardsson, D., Ekholuenetale, M., El Sayed, I., El-Jaafary, S. I., Eskandari, K., Eskandarieh, S., Esmaeilnejad, S., Fares, J., Faro, A., Farooque, U., Feigin, V. L., Feng, X., Fereshtehnejad, S. M., Fernandes, E., Ferrara, P., Filip, I., Fillit, H., Fischer, F., Gaidhane, S., Galluzzo, L., Ghashghaee, A., Ghith, N., Gialluisi, A., Gilani, S. A., Glavan, I. R., Gnedovskaya, E. V., Golechha, M., Gupta, R., Gupta, V. B., Gupta, V. K., Haider, M. R., Hall, B. J., Hamidi, S., Hanif, A., Hankey, G. J., Haque, S., Hartono, R. K., Hasaballah, A. I., Hasan, M. T., Hassan, A., Hay, S. I., Hayat, K., Hegazy, M. I., Heidari, G., Heidari-Soureshjani, R., Herteliu, C., Househ, M., Hussain, R., Hwang, B. F., Iacoviello, L., Iavicoli, I., Ilesanmi, O. S., Ilic, I. M., Ilic, M. D., Irvani, S. S. N., Iso, H., Iwagami, M., Jabbarinejad, R., Jacob, L., Jain, V., Jayapal, S. K., Jayawardena, R., Jha, R. P., Jonas, J. B., Joseph, N., Kalani, R., Kandel, A., Kandel, H., Karch, A., Kasa, A. S., Kassie, G. M., Keshavarz, P., Khan, M. A., Khatib, M. N., Khoja, T. A. M., Khubchandani, J., Kim, M. S., Kim, Y. J., Kisa, A., Kisa, S., Kivimäki, M., Koroshetz, W. J., Koyanagi, A., Kumar, G. A., Kumar, M., Lak, H. M., Leonardi, M., Li, B., Lim, S. S., Liu, X., Liu, Y., Logroscino, G., Lorkowski, S., Lucchetti, G., Lutzky Saute, R., Magnani, F. G., Malik, A. A., Massano, J., Mehndiratta, M. M., Menezes, R. G., Meretoja, A., Mohajer, B., Mohamed Ibrahim, N., Mohammad, Y., Mohammed, A., Mokdad, A. H., Mondello, S., Moni, M. A. A., Moniruzzaman, M., Mossie, T. B., Nagel, G., Naveed, M., Nayak, V. C., Neupane Kandel, S., Nguyen, T. H., Oancea, B., Otstavnov, N., Otstavnov, S. S., Owolabi, M. O., Panda-Jonas, S., Pashazadeh Kan, F., Pasovic, M., Patel, U. K., Pathak, M., Peres, M. F., Perianayagam, A., Peterson, C. B., Phillips, M. R., Pinheiro, M., Piradov, M. A., Pond, C. D., Potashman, M. H., Pottoo, F. H., Prada, S. I., Radfar, A., Raggi, A., Rahim, F., Rahman, M., Ram, P., Ranasinghe, P., Rawaf, D. L., Rawaf, S., Rezaei, N., Rezapour, A., Robinson, S. R., Romoli, M., Roshandel, G., Sahathevan, R., Sahebkar, A., Sahraian, M. A., Sathian, B., Sattin, D., Sawhney, M., Saylan, M., Schiavolin, S., Seylani, A., Sha, F., Shaikh, M. A., Shaji, K. S.,

Shannawaz, M., Shetty, J. K., Shigematsu, M., Shin, J. I., Shiri, R., Silva, D. A. S., Silva, J. P., Silva, R., Singh, J. A., Skryabin, V. Y., Skryabina, A. A., Smith, A. E., Soshnikov, S., Spurlock, E. E., Stein, D. J., Sun, J., Tabarés-Seisdedos, R., Thakur, B., Timalsina, B., Tovani-Palone, M. R., Tran, B. X., Tsegaye, G. W., Valadan Tahbaz, S., Valdez, P. R., Venketasubramanian, N., Vlassov, V., Vu, G. T., Vu, L. G., Wang, Y. P., Wimo, A., Winkler, A. S., Yadav, L., Yahyazadeh Jabbari, S. H., Yamagishi, K., Yang, L., Yano, Y., Yonemoto, N., Yu, C., Yunusa, I., Zadey, S., Zastrozhin, M. S., Zastrozhina, A., Zhang, Z. J., Murray, C. J., and Vos, T. *Estimation of the global prevalence of dementia in 2019 and forecasted prevalence in 2050: an analysis for the Global Burden of Disease Study 2019*. *The Lancet Public Health*, 7(2):e105–e125, 2022. doi: 10.1016/S2468-2667(21)00249-8.

[150] *2019 Alzheimer's disease facts and figures*. *Alzheimer's & Dementia*, 15(3):321–387, 2019. doi: 10.1016/j.jalz.2019.01.010.

[151] Cummings, J., Lee, G., Nahed, P., Kambar, M. E. Z. N., Zhong, K., Fonseca, J., and Taghva, K. *Alzheimer's disease drug development pipeline: 2022*. *Alzheimer's and Dementia: Translational Research and Clinical Interventions*, 8(1), 2022. doi: 10.1002/trc2.12295.

[152] Bjørklund, G., Aaseth, J., Dadar, M., and Chirumbolo, S. *Molecular targets in Alzheimer's disease*. *Molecular neurobiology*, 56:7032–7044, 2019.

[153] Mullard, A. *Alzheimer amyloid hypothesis lives on*. *Nature Reviews Drug Discovery*, 16(1):3–5, 2016. doi: 10.1038/nrd.2016.281.

[154] Colvin, M. T., Silvers, R., Ni, Q. Z., Can, T. V., Sergeyev, I., Rosay, M., Donovan, K. J., Michael, B., Wall, J., Linse, S., and Griffin, R. G. *Atomic Resolution Structure of Monomorphic Aβ42 Amyloid Fibrils*. *Journal of the American Chemical Society*, 138(30):9663–9674, 2016. doi: 10.1021/jacs.6b05129.

[155] Das, B. and Yan, R. *A Close Look at BACE1 Inhibitors for Alzheimer's Disease Treatment*. *CNS Drugs*, 33(3):251–263, 2019. doi: 10.1007/s40263-019-00613-7.

[156] Blennow, K. and Zetterberg, H. *Semagacestat's fall: Where next for AD therapies?* *Nature Medicine*, 19(10):1214–1215, 2013. doi: 10.1038/nm.3365.

[157] Das, B. and Yan, R. *A Close Look at BACE1 Inhibitors for Alzheimer's Disease Treatment*. *CNS Drugs*, 33(3):251–263, 2019. doi: 10.1007/s40263-019-00613-7.

[158] Zetterberg, H., Andreasson, U., Hansson, O., Wu, G., Sankaranarayanan, S., Andersson, M. E., Buchhave, P., Londos, E., Umek, R. M., Minthon, L., Simon, A. J., and Blennow, K. *Elevated cerebrospinal fluid BACE1 activity in incipient alzheimer disease*. *Archives of Neurology*, 65(8):1102–1107, 2008. doi: 10.1001/archneur.65.8.1102.

[159] Fukumoto, H., Rosene, D. L., Moss, M. B., Raju, S., Hyman, B. T., and Irizarry, M. C. *β-Secretase Activity Increases with Aging in Human, Monkey, and Mouse Brain*. *American Journal of Pathology*, 164(2):719–725, 2004. doi: 10.1016/S0002-9440(10)63159-8.

[160] Li, R., Lindholm, K., Yang, L. B., Yue, X., Citron, M., Yan, R., Beach, T., Sue, L., Sebbagh, M., Cai, H., Wong, P., Price, D., and Shen, Y. *Amyloid β peptide load is correlated with increased β-secretase activity in sporadic Alzheimer's disease patients*. *Proceedings of the National Academy of Sciences of the United States of America*, 101(10):3632–3637, 2004. doi: 10.1073/pnas.0205689101.

[161] Haass, C., Lemere, C. A., Capell, A., Citron, M., Seubert, P., Schenk, D., Lannfelt, L., and Selkoe, D. J. *The Swedish mutation causes early-onset Alzheimer's disease by β-secretase cleavage within the secretory pathway*. *Nature Medicine*, 1(12):1291–1296, 1995. doi: 10.1038/nm1295-1291.

[162] Jonsson, T., Atwal, J. K., Steinberg, S., Snaedal, J., Jonsson, P. V., Bjornsson, S., Stefansson, H., Sulem, P., Gudbjartsson, D., Maloney, J., Hoyte, K., Gustafson, A., Liu, Y., Lu, Y., Bhangale, T., Graham, R. R., Huttenlocher, J., Bjornsdottir, G., Andreassen, O. A., Jonsson, E. G., Palotie, A., Behrens, T. W., Magnusson, O. T., Kong, A., Thorsteinsdottir, U., Watts, R. J., and Stefansson, K. *A mutation in APP protects against Alzheimer's disease and age-related cognitive decline*. *Nature*, 488(7409):96, 2012. doi: 10.1038/nature11283.

[163] Martiskainen, H., Herukka, S. K., Stančáková, A., Paananen, J., Soininen, H., Kuusisto, J., Laakso, M., and Hiltunen, M. *Decreased plasma β-amyloid in the Alzheimer's disease APP A673T variant carriers*. *Annals of Neurology*, 82(1):128–132, 2017. doi: 10.1002/ana.24969.

[164] Luo, Y., Sunderland, T., Roth, G. S., and Wolozin, B. *Physiological levels of β-amyloid peptide promote PC12 cell proliferation*. *Neuroscience Letters*, 217(2-3):125–128, 1996. doi: 10.1016/0304-3940(96)13087-1.

[165] Chen, Y. and Dong, C. *Aβ40 promotes neuronal cell fate in neural progenitor cells*. *Cell Death and Differentiation*, 16(3):386–394, 2009. doi: 10.1038/cdd.2008.94.

[166] Forman, M., Palcza, J., Tseng, J., Leempoels, J., Ramael, S., Han, D., Jhee, S., Ereshefsky, L., Tanen, M., Laterza, O., Dockendorf, M., Krishna, G., Ma, L., Wagner, J., and Troyer, M. *P4-196: The novel BACE inhibitor MK-8931 dramatically lowers cerebrospinal fluid Aβ peptides in healthy subjects following single- and multiple-dose administration*. *Alzheimer's & Dementia*, 8(4S_Part_-19), 2012. doi: 10.1016/j.jalz.2012.05.1900.

[167] Zhu, K., Peters, F., Filser, S., and Herms, J. *Consequences of Pharmacological BACE Inhibition on Synaptic Structure and Function*. *Biological Psychiatry*, 84(7):478–487, 2018. doi: 10.1016/j.biopsych.2018.04.022.

[168] Satir, T. M., Agholme, L., Karlsson, A., Karlsson, M., Karila, P., Illes, S., Bergström, P., and Zetterberg, H. *Partial reduction of amyloid β production by β-secretase inhibitors does not decrease synaptic transmission*. *Alzheimer's Research and Therapy*, 12(1):1–9, 2020. doi: 10.1186/s13195-020-00635-0.

[169] McDade, E., Voytyuk, I., Aisen, P., Bateman, R. J., Carrillo, M. C., De Strooper, B., Haass, C., Reiman, E. M., Sperling, R., Tariot, P. N., Yan, R., Masters, C. L., Vassar, R., and Lichtenthaler, S. F. *The case for low-level BACE1 inhibition for the prevention of Alzheimer disease*. *Nature Reviews Neurology*, 17(11):703–714, 2021. doi: 10.1038/s41582-021-00545-1.

[170] Singh, N., Das, B., Zhou, J., Hu, X., and Yan, R. *Targeted BACE-1 inhibition in microglia enhances amyloid clearance and improved cognitive performance*, 2022. doi: 10.1126/sciadv.abo3610.

[171] Singh, N., Benoit, M. R., Zhou, J., Das, B., Davila-Velderrain, J., Kellis, M., Tsai, L. H., Hu, X., and Yan, R. *BACE-1 inhibition facilitates the transition from homeostatic microglia to DAM-1*. *Science Advances*, 8(24):eabo1286, 2022. doi: 10.1126/sciadv.abo1286.

[172] Fujimoto, K., Matsuoka, E., Asada, N., Tadano, G., Yamamoto, T., Nakahara, K., Fuchino, K., Ito, H., Kanegawa, N., Moechars, D., Gijsen, H. J., and Kusakabe, K. I. *Structure-Based Design of Selective β-Site Amyloid Precursor Protein Cleaving Enzyme 1 (BACE1) Inhibitors: Targeting the Flap to Gain Selectivity over BACE2*. *Journal of Medicinal Chemistry*, 62(10):5080–5095, 2019. doi: 10.1021/acs.jmedchem.9b00309.

[173] Bennett, B. D., Babu-Khan, S., Loeloff, R., Louis, J. C., Curran, E., Citron, M., and Vassar, R. *Expression analysis of BACE2 in brain and peripheral tissues*. *Journal of Biological Chemistry*, 275(27):20647–20651, 2000. doi: 10.1074/jbc.M002688200.

[174] Voytyuk, I., Mueller, S. A., Herber, J., Snellinx, A., Moechars, D., van Loo, G., Lichtenthaler, S. F., and De Strooper, B. *BACE2 distribution in major brain cell types and identification of novel substrates*. *Life Science Alliance*, 1(1):e201800026, 2018. doi: 10.26508/lsa.201800026.

[175] Yen, Y. C., Kammeyer, A. M., Tirlangi, J., Ghosh, A. K., and Mesecar, A. D. *A Structure-Based Discovery Platform for BACE2 and the Development of Selective BACE Inhibitors*. *ACS Chemical Neuroscience*, 12(4):581–588, 2021. doi: 10.1021/acschemneuro.0c00629.

[176] Fujimoto, K., Yoshida, S., Tadano, G., Asada, N., Fuchino, K., Suzuki, S., Matsuoka, E., Yamamoto, T., Yamamoto, S., Ando, S., Kanegawa, N., Tonomura,

Y., Ito, H., Moechars, D., Rombouts, F. J. R., Gijsen, H. J. M., and Kusakabe, K.-i. *Structure-Based Approaches to Improving Selectivity through Utilizing Explicit Water Molecules: Discovery of Selective β-Secretase (BACE1) Inhibitors over BACE2*. *Journal of Medicinal Chemistry*, 64(6):3075–3085, 2021. doi: 10.1021/acs.jmedchem.0c01858.

[177] Shimizu, H., Tosaki, A., Kaneko, K., Hisano, T., Sakurai, T., and Nukina, N. *Crystal Structure of an Active Form of BACE1, an Enzyme Responsible for Amyloid β Protein Production*. *Molecular and Cellular Biology*, 28(11):3663–3671, 2008. doi: 10.1128/MCB.02185-07.

[178] Grüninger-Leitch, F., Schlatter, D., Küng, E., Nelböck, P., and Döbeli, H. *Substrate and Inhibitor Profile of BACE (β-Secretase) and Comparison with Other Mammalian Aspartic Proteases*. *Journal of Biological Chemistry*, 277(7):4687–4693, 2002. doi: 10.1074/jbc.M109266200.

[179] Yildiz, M. *Docking studies suggest the important role of interactions among the catalytic dyad and inhibitors for designing Bace1 specific inhibitors*. *Journal of Molecular Structure*, 1208:127893, 2020. doi: 10.1016/j.molstruc.2020.127893.

[180] Moussa-Pacha, N. M., Abdin, S. M., Omar, H. A., Alniss, H., and Al-Tel, T. H. *BACE1 inhibitors: Current status and future directions in treating Alzheimer's disease*. *Medicinal Research Reviews*, 40(1):339–384, 2020. doi: 10.1002/med.21622.

[181] Ellis, C. R. and Shen, J. *PH-dependent population shift regulates BACE1 activity and inhibition*. *Journal of the American Chemical Society*, 137(30):9543–9546, 2015. doi: 10.1021/jacs.5b05891.

[182] Manoharan, P., Chennoju, K., and Ghoshal, N. *Computational analysis of BACE1-ligand complex crystal structures and linear discriminant analysis for identification of BACE1 inhibitors with anti P-glycoprotein binding property*. *Journal of Biomolecular Structure and Dynamics*, 36(1):262–276, 2018. doi: 10.1080/07391102.2016.1276477.

[183] Arora, S. and Barak, B. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[184] Fleck, M. M. *Building Blocks for Theoretical Computer Science*. University of Illinois, 2017.

[185] Drowell, E. *Big-O cheat sheet*.

[186] Sargsyan, K., Grauffel, C., and Lim, C. *How Molecular Size Impacts RMSD Applications in Molecular Dynamics Simulations*. *Journal of Chemical Theory and Computation*, 13(4):1518–1524, 2017. doi: 10.1021/acs.jctc.7b00028.

[187] Carugo, O. *How root-mean-square distance (r.m.s.d.) values depend on the resolution of protein structures that are compared*. *Journal of Applied Crystallography*, 36(1):125–128, 2003. doi: 10.1107/S0021889802020502.

[188] Damm, K. L. and Carlson, H. A. *Gaussian-weighted RMSD superposition of proteins: A structural comparison for flexible proteins and predicted protein structures*. *Biophysical Journal*, 90(12):4558–4573, 2006. doi: 10.1529/biophysj.105.066654.

[189] Wang, X. and Dong, J. *A Normalized Weighted RMSD for Measuring Protein Structure Superposition*. In *2012 2nd International Conference on Biomedical Engineering and Technology*, volume 34, pages 68–72. 2012.

[190] Rand, W. M. *Objective criteria for the evaluation of clustering methods*. *Journal of the American Statistical Association*, 66(336):846–850, 1971. doi: 10.1080/01621459.1971.10482356.

[191] Hubert, L. and Arabie, P. *Comparing partitions*. *Journal of Classification*, 2(1):193–218, 1985. doi: 10.1007/BF01908075.

[192] Steinley, D. *Properties of the Hubert-Arabie adjusted Rand index*. *Psychological Methods*, 9(3):386–396, 2004. doi: 10.1037/1082-989X.9.3.386.

[193] Bajusz, D., Rácz, A., and Héberger, K. *Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?* *Journal of Cheminformatics*, 7(1):1–13, 2015. doi: 10.1186/s13321-015-0069-3.

[194] Aggarwal, S. and Kumar, N. *Data structures*. In *Advances in Computers*, volume 121, pages 43–81. Elsevier, 2021.

[195] Butterfield, A. and Ngondi, G. E., editors. *A Dictionary of Computer Science*. Oxford University Press, 2016. doi: 10.1093/acref/9780199688975.001.0001.

[196] Christensson, P. *Heap Definition*, 2012.

[197] Bentley, J. L. *Multidimensional Binary Search Trees Used for Associative Searching*. *Communications of the ACM*, 18(9):509–517, 1975. doi: 10.1145/361002.361007.

[198] Baeldung. *kd-tree Definition*.

[199] Yianilos, P. N. *Data structures and algorithms for nearest neighbor search in general metric spaces*. *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 311–321, 1993.

[200] Jain, A. K., Murty, M. N., and Flynn, P. J. *Data clustering: A review*. *ACM Computing Surveys*, 31(3):264–323, 1999. doi: 10.1145/331499.331504.

[201] Frigui, H. *Clustering: Algorithms and applications*. In *2008 1st International Workshops on Image Processing Theory, Tools and Applications, IPTA 2008*, pages 1–11. IEEE, 2008. doi: 10.1109/IPTA.2008.4743793.

[202] Duan, S., Fokoue, A., Srinivas, K., and Byrne, B. *A Clustering-based Approach to Ontology*. *Proceedings of the 10th International Semantic Web Conference*, pages 146–161, 2011.

[203] Lesieutre, B. C., Rogers, K. M., Overbye, T. J., and Borden, A. R. *A sensitivity approach to detection of local market power potential*. *IEEE Transactions on Power Systems*, 26(4):1980–1988, 2011. doi: 10.1109/TPWRS.2011.2105893.

[204] Xia, X. and Xie, Z. *AMADA: Analysis of microarray data*. *Bioinformatics*, 17(6):569–570, 2001. doi: 10.1093/bioinformatics/17.6.569.

[205] Jiang, D., Pei, J., and Zhang, A. *An interactive approach to mining gene expression data*. *IEEE Transactions on Knowledge and Data Engineering*, 17(10):1363–1378, 2005. doi: 10.1109/TKDE.2005.159.

[206] Shao, J., Tanner, S. W., Thompson, N., and Cheatham, T. E. *Clustering molecular dynamics trajectories: 1. Characterizing the performance of different clustering algorithms*. *Journal of Chemical Theory and Computation*, 3(6):2312–2334, 2007. doi: 10.1021/ct700119m.

[207] Peng, J. H., Wang, W., Yu, Y. Q., Gu, H. L., and Huang, X. *Clustering algorithms to analyze molecular dynamics simulation trajectories for complex chemical and biological systems*. *Chinese Journal of Chemical Physics*, 31(4):404–420, 2018. doi: 10.1063/1674-0068/31/cjcp1806147.

[208] Schuffenhauer, A., Ruedisser, S., Marzinzik, A., Jahnke, W., Selzer, P., and Jacoby, E. *Library Design for Fragment Based Screening*. *Current Topics in Medicinal Chemistry*, 5(8):751–762, 2005. doi: 10.2174/1568026054637700.

[209] Schulz, M. N., Landström, J., Bright, K., and Hubbard, R. E. *Design of a Fragment Library that maximally represents available chemical space*. *Journal of Computer-Aided Molecular Design*, 25(7):611–620, 2011. doi: 10.1007/s10822-011-9461-x.

[210] Zerbe, B. S., Hall, D. R., Vajda, S., Whitty, A., and Kozakov, D. *Relationship between hot spot residues and ligand binding hot spots in protein-protein interfaces*. *Journal of Chemical Information and Modeling*, 52(8):2236–2244, 2012. doi: 10.1021/ci300175u.

[211] Hall, D. R., Kozakov, D., Whitty, A., and Vajda, S. *Lessons from Hot Spot Analysis for Fragment-Based Drug Discovery*. *Trends in Pharmacological Sciences*, 36(11):724–736, 2015. doi: 10.1016/j.tips.2015.08.003.

[212] von Luxburg, U., Williamson BobWilliamson, R. C., and Guyon, I. *Clustering: Science or Art? JMLR: Workshop and Conference Proceedings*, 27:6579, 2012.

[213] Abu-Jamous, B., Fa, R., and Nandi, A. K. *Integrative cluster analysis in bioinformatics*. 2015.

[214] Heyer, L. J., Kruglyak, S., and Yooseph, S. *Exploring expression data identification and analysis of coexpressed genes*. *Genome Research*, 9(11):1106–1115, 1999. doi: 10.1101/gr.9.11.1106.

[215] Yaakob, S. N., Lim, C. P., and Jain, L. *A novel Euclidean quality threshold ARTMAP network and its application to pattern classification*. *Neural Computing and Applications*, 19(2):227–236, 2010. doi: 10.1007/s00521-009-0293-8.

[216] Dutta, S. and Overbye, T. J. *A clustering based wind farm collector system cable layout design*. *2011 IEEE Power and Energy Conference at Illinois, PECI 2011*, pages 4–9, 2011. doi: 10.1109/PECI.2011.5740480.

[217] Olson, M. T., Epstein, J. A., Sackett, D. L., and Yergey, A. L. *Production of reliable MALDI spectra with quality threshold clustering of replicates*. *Journal of the American Society for Mass Spectrometry*, 22(6):969–975, 2011. doi: 10.1007/s13361-011-0097-9.

[218] Yaakob, S. N. and Jain, L. *An insect classification analysis based on shape features using quality threshold ARTMAP and moment invariant*. *Applied Intelligence*, 37(1):12–30, 2012. doi: 10.1007/s10489-011-0310-3.

[219] Danalis, A., McCurdy, C., and Vetter, J. S. *Efficient quality threshold clustering for parallel architectures*. In *Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium, IPDPS 2012*, pages 1068–1079. IEEE, 2012. doi: 10.1109/IPDPS.2012.99.

[220] Procacci, P., Darden, T. A., Paci, E., and Marchi, M. *ORAC: A molecular dynamics program to simulate complex molecular systems with realistic electrostatic interactions*. *Journal of Computational Chemistry*, 18(15):1848–1862, 1997. doi: 10.1002/(SICI)1096-987X(19971130)18:15<1848::AID-JCC2>3.0.CO;2-O.

[221] Humphrey, W., Dalke, A., and Schulten, K. *VMD: Visual molecular dynamics*. *Journal of Molecular Graphics*, 14(1):33–38, 1996. doi: 10.1016/0263-7855(96)00018-5.

[222] Seeber, M., Cecchini, M., Rao, F., Settanni, G., and Caflisch, A. *Wordom: A program for efficient analysis of molecular dynamics simulations*. *Bioinformatics*, 23(19):2625–2627, 2007. doi: 10.1093/bioinformatics/btm378.

[223] Melvin, R. and Salsbury, F. R. *Python Implementation of Quality Threshold Clustering for Molecular Dynamics*, 2016. doi: 10.6084/m9.figshare.3813930.v2.

[224] Daura, X., Gademann, K., Jaun, B., Seebach, D., van Gunsteren, W. F., and Mark, A. E. *Peptide Folding: When Simulation Meets Experiment*. *Angewandte Chemie International Edition*, 38(1/2):236–240, 1999. doi: 10.1002/(sici)1521-3773(19990115)38:1/2<236::aid-anie236>3.3.co;2-d.

[225] Rodriguez, A. and Laio, A. *Clustering by fast search and find of density peaks*. *Science*, 344(6191):1492–1496, 2014. doi: 10.1126/science.1242072.

[226] Träger, S., Tamò, G., Aydin, D., Fonti, G., Audagnotto, M., and Dal Peraro, M. *CLoNe: Automated clustering based on local density neighborhoods for application to biomolecular structural ensembles*. *Bioinformatics*, 37(7):921–928, 2021. doi: 10.1093/bioinformatics/btaa742.

[227] Seyedi, S. A., Lotfi, A., Moradi, P., and Qader, N. N. *Dynamic graph-based label propagation for density peaks clustering*. *Expert Systems with Applications*, 115:314–328, 2019. doi: 10.1016/j.eswa.2018.07.075.

[228] Flores, K. G. and Garza, S. E. *Density peaks clustering with gap-based automatic center detection*. *Knowledge-Based Systems*, 206:106350, 2020. doi: 10.1016/j.knosys.2020.106350.

[229] Wang, X. F. and Xu, Y. *Fast clustering using adaptive density peak detection*. *Statistical Methods in Medical Research*, 26(6):2800–2811, 2017. doi: 10.1177/0962280215609948.

[230] Du, M., Ding, S., and Jia, H. *Study on density peaks clustering based on k-nearest neighbors and principal component analysis*. *Knowledge-Based Systems*, 99:135–145, 2016. doi: 10.1016/j.knosys.2016.02.001.

[231] Du, M., Ding, S., and Xue, Y. *A novel density peaks clustering algorithm for mixed data*. *Pattern Recognition Letters*, 97:46–53, 2017. doi: 10.1016/j.patrec.2017.07.001.

[232] Majdara, A. and Nooshabadi, S. *Accelerated density-based clustering using Bayesian sequential partitioning*. In *Proceedings - IEEE International Symposium on Circuits and Systems*, volume 2020-Octob, pages 1–5. IEEE, 2020. doi: 10.1109/iscas45731.2020.9181237.

[233] Liang, Z. and Chen, P. *Delta-density based clustering with a divide-and-conquer strategy: 3DC clustering*. *Pattern Recognition Letters*, 73:52–59, 2016. doi: 10.1016/j.patrec.2016.01.009.

[234] Wang, G., Bu, C., and Luo, Y. *Modified FDP cluster algorithm and its application in protein conformation clustering analysis*. *Digital Signal Processing: A Review Journal*, 92:97–108, 2019. doi: 10.1016/j.dsp.2019.04.011.

[235] Roe, D. R. and Cheatham, T. E. *PTRAJ and CPPTRAJ: Software for processing and analysis of molecular dynamics trajectory data*. *Journal of Chemical Theory and Computation*, 9(7):3084–3095, 2013. doi: 10.1021/ct400341p.

[236] Campello, R. J. G. B., Moulavi, D., and Sander, J. *Density-Based Clustering Based on Hierarchical Density Estimates*. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7819 LNAI, pages 160–172. 2013. doi: 10.1007/978-3-642-37456-2_14.

[237] Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. *DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN*. *ACM Transactions on Database Systems*, 42(3):1–21, 2017. doi: 10.1145/3068335.

[238] Hinneburg, A. and Keim, D. A. *A General Approach to Clustering in Large Databases with Noise*. *Knowledge and Information Systems*, 5(4):387–415, 2003. doi: 10.1007/s10115-003-0086-9.

[239] Sun, H., Huang, J., Hanr, J., Deng, H., Zhaor, P., and Feng, B. *gSkeletonClu: Density-based network clustering via structure-connected tree division or agglomeration*. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 481–490. IEEE, 2010. doi: 10.1109/ICDM.2010.69.

[240] Pei, T., Jasra, A., Hand, D. J., Zhu, A. X., and Zhou, C. *DECODE: A new method for discovering clusters of different densities in spatial data*. *Data Mining and Knowledge Discovery*, 18(3):337–369, 2009. doi: 10.1007/s10618-008-0120-3.

[241] Stuetzle, W. and Nugent, R. *A generalized single linkage method for estimating the cluster tree of a density*. *Journal of Computational and Graphical Statistics*, 19(2):397–418, 2010. doi: 10.1198/jcgs.2009.07049.

[242] Melvin, R. L., Godwin, R. C., Xiao, J., Thompson, W. G., Berenhaut, K. S., and Salsbury, F. R. *Uncovering Large-Scale Conformational Change in Molecular Dynamics without Prior Knowledge*. *Journal of Chemical Theory and Computation*, 12(12):6130–6146, 2016. doi: 10.1021/acs.jctc.6b00757.

[243] Melvin, R. L., Xiao, J., Godwin, R. C., Berenhaut, K. S., and Salsbury, F. R. *Visualizing correlated motion with HDBSCAN clustering*. *Protein Science*, 27(1):62–75, 2018. doi: 10.1002/pro.3268.

[244] McInnes, L. and Healy, J. *Accelerated Hierarchical Density Based Clustering*. In *IEEE International Conference on Data Mining Workshops, ICDMW*, volume 2017-Novem, pages 33–42. IEEE, 2017. doi: 10.1109/ICDMW.2017.12.

[245] Berendsen, H. J., van der Spoel, D., and van Drunen, R. *GROMACS: A message-passing parallel molecular dynamics implementation*. *Computer Physics Communications*, 91(1-3):43–56, 1995. doi: 10.1016/0010-4655(95)00042-E.

[246] Tubiana, T., Carvaillo, J. C., Boulard, Y., and Bressanelli, S. *TTClust: A Versatile Molecular Simulation Trajectory Clustering Program with Graphical Summaries*. *Journal of Chemical Information and Modeling*, 58(11):2178–2182, 2018. doi: 10.1021/acs.jcim.8b00512.

[247] Durrant, J. D. and McCammon, J. A. *BINANA: A novel algorithm for ligand-binding characterization*. *Journal of Molecular Graphics and Modelling*, 29(6):888–893, 2011. doi: 10.1016/j.jmgm.2011.01.004.

[248] Zhang, Y. and Skolnick, J. *TM-align: A protein structure alignment algorithm based on the TM-score*. *Nucleic Acids Research*, 33(7):2302–2309, 2005. doi: 10.1093/nar/gki524.

[249] Jo, S., Kim, T., Iyer, V. G., and Im, W. *CHARMM-GUI: A web-based graphical user interface for CHARMM*. *Journal of Computational Chemistry*, 29(11):1859–1865, 2008. doi: 10.1002/jcc.20945.

[250] Im, W., Lee, M. S., and Brooks, C. L. *Generalized Born Model with a Simple Smoothing Function*. *Journal of Computational Chemistry*, 24(14):1691–1702, 2003. doi: 10.1002/jcc.10321.

[251] Chocholoušová, J. and Feig, M. *Balancing an accurate representation of the molecular surface in generalized born formalisms with integrator stability in molecular dynamics simulations*. *Journal of Computational Chemistry*, 27(6):719–729, 2006. doi: 10.1002/jcc.20387.

[252] Trott, O. and Olson, A. J. *AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading*. *Journal of Computational Chemistry*, 31(2):NA–NA, 2009. doi: 10.1002/jcc.21334.

[253] Wang, C. and Zhang, Y. *Improving scoring-docking-screening powers of protein–ligand scoring functions using random forest*. *Journal of Computational Chemistry*, 38(3):169–177, 2017. doi: 10.1002/jcc.24667.

[254] Smith, R. H., Dar, A. C., and Schlessinger, A. *PyVOL: a PyMOL plugin for visualization, comparison, and volume calculation of drug-binding sites*. *bioRxiv*, page 816702, 2019. doi: 10.1101/816702.

[255] 2.4.4.5, O. T. *{O}penEye {S}cientific {S}oftware, {S}anta {F}e, {NM}*.

[256] Shea, J. E. and Levine, Z. A. *Studying the early stages of protein aggregation using replica exchange molecular dynamics simulations*. In *Methods in Molecular Biology*, volume 1345, pages 225–250. 2016. doi: 10.1007/978-1-4939-2978-8_15.

[257] McGibbon, R. T., Beauchamp, K. A., Harrigan, M. P., Klein, C., Swails, J. M., Hernández, C. X., Schwantes, C. R., Wang, L. P., Lane, T. J., and Pande, V. S. *MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories*. *Biophysical Journal*, 109(8):1528–1532, 2015. doi: 10.1016/j.bpj.2015.08.015.

[258] Guardiani, C., Signorini, G. F., Livi, R., Papini, A. M., and Procacci, P. *Conformational landscape of N-glycosylated peptides detecting autoantibodies in multiple sclerosis, revealed by hamiltonian replica exchange*. *Journal of Physical Chemistry B*, 116(18):5458–5467, 2012. doi: 10.1021/jp301442n.

[259] Olsson, M. H., SØndergaard, C. R., Rostkowski, M., and Jensen, J. H. *PROPKA3: Consistent treatment of internal and surface residues in empirical p K a predictions*. *Journal of Chemical Theory and Computation*, 7(2):525–537, 2011. doi: 10.1021/ct100578z.

[260] Søndergaard, C. R., Olsson, M. H., Rostkowski, M., and Jensen, J. H. *Improved treatment of ligands and coupling effects in empirical calculation and rationalization of p K a values*. *Journal of Chemical Theory and Computation*, 7(7):2284–2295, 2011. doi: 10.1021/ct200133y.

[261] Viji, S. N., Balaji, N., and Gautham, N. *Molecular docking studies of protein-nucleotide complexes using MOLSDOCK (mutually orthogonal Latin squares DOCK)*. *Journal of Molecular Modeling*, 18(8):3705–3722, 2012. doi: 10.1007/s00894-012-1369-4.

[262] Jhoti, H., Williams, G., Rees, D. C., and Murray, C. W. *The 'rule of three' for fragment-based drug discovery: Where are we now? Nature Reviews Drug Discovery*, 12(8):644, 2013. doi: 10.1038/nrd3926-c1.

[263] Schneider, B., Morávek, Z., and Berman, H. M. *RNA conformational classes*. *Nucleic Acids Research*, 32(5):1666–1677, 2004. doi: 10.1093/nar/gkh333.

[264] Icazatti, A. A., Loyola, J. M., Szleifer, I., Vila, J. A., and Martin, O. A. *Classification of RNA backbone conformations into rotamers using 13C' chemical shifts: Exploring how far we can go*. *PeerJ*, 2019(10):e7904, 2019. doi: 10.7717/peerj.7904.

[265] Scharl, T. and Leisch, F. *The Stochastic QT Clust Algorithm: Evaluation of Stability and Variance on Time Course Microarray Data*. *Proceedings in Computational Statistics*, pages 1015–1022, 2006.

[266] Loforte, F. *Efficient Variations of the Quality Threshold Clustering Algorithm*. (43), 2015.

[267] Wu, Q. and Hao, J. K. *A review on algorithms for maximum clique problems*. *European Journal of Operational Research*, 242(3):693–709, 2015. doi: 10.1016/j.ejor.2014.09.064.

[268] Gerstberger, S., Hafner, M., and Tuschl, T. *A census of human RNA-binding proteins*. *Nature Reviews Genetics*, 15(12):829–845, 2014. doi: 10.1038/nrg3813.

[269] Lunde, B. M., Hörner, M., and Meinhart, A. *Structural insights into cis element recognition of non-polyadenylated RNAs by the Nab3-RRM*. *Nucleic Acids Research*, 39(1):337–346, 2011. doi: 10.1093/nar/gkq751.

[270] Murn, J., Teplova, M., Zarnack, K., Shi, Y., and Patel, D. J. *Recognition of distinct RNA motifs by the clustered CCCH zinc fingers of neuronal protein Unkempt*. *Nature Structural & Molecular Biology*, 23(1):16–23, 2016. doi: 10.1038/nsmb.3140.

[271] Yang, L., Wang, C., Li, F., Zhang, J., Nayab, A., Wu, J., Shi, Y., and Gong, Q. *The human RNA-binding protein and E3 ligase MEX-3C binds the MEX-3–recognition element (MRE) motif with high affinity*. *Journal of Biological Chemistry*, 292(39):16221–16234, 2017. doi: 10.1074/jbc.M117.797746.

[272] Deo, R. C., Bonanno, J. B., Sonenberg, N., and Burley, S. K. *Recognition of polyadenylate RNA by the poly(A)-binding protein*. *Cell*, 98(6):835–845, 1999. doi: 10.1016/S0092-8674(00)81517-2.

[273] Shen, W., De Hoyos, C. L., Migawa, M. T., Vickers, T. A., Sun, H., Low, A., Bell, T. A., Rahdar, M., Mukhopadhyay, S., Hart, C. E., Bell, M., Riney, S., Murray, S. F., Greenlee, S., Crooke, R. M., hai Liang, X., Seth, P. P., and Crooke, S. T. *Chemical modification of PS-ASO therapeutics reduces cellular protein-binding and improves the therapeutic index*. *Nature Biotechnology*, 37(6):640–650, 2019. doi: 10.1038/s41587-019-0106-2.

[274] Kuttel, M., Mao, Y., Widmalm, G., and Lundborg, M. *CarbBuilder: An adjustable tool for building 3D molecular structures of carbohydrates for molecular simulation*. *Proceedings - 2011 7th IEEE International Conference on eScience, eScience 2011*, pages 395–402, 2011. doi: 10.1109/eScience.2011.61.

[275] Jorgensen, W. L., Chandrasekhar, J., Madura, J. D., Impey, R. W., and Klein, M. L. *Comparison of simple potential functions for simulating liquid water*. *The Journal of Chemical Physics*, 79(2):926–935, 1983. doi: 10.1063/1.445869.

[276] Phillips, J. C., Schulten, K., Bhatele, A., Mei, C., Sun, Y., Bohm, E. J., and Kale, L. V. *Scalable molecular dynamics with NAMD*. *Parallel Science and*

Engineering Applications: The Charm++ Approach, 26(16):60–76, 2016. doi: 10.1201/b16251-15.

[277] MacKerell, A. D., Bashford, D., Bellott, M., Dunbrack, R. L., Evanseck, J. D., Field, M. J., Fischer, S., Gao, J., Guo, H., Ha, S., Joseph-McCarthy, D., Kuchnir, L., Kuczera, K., Lau, F. T., Mattos, C., Michnick, S., Ngo, T., Nguyen, D. T., Prodhom, B., Reiher, W. E., Roux, B., Schlenkrich, M., Smith, J. C., Stote, R., Straub, J., Watanabe, M., Wiórkiewicz-Kuczera, J., Yin, D., and Karplus, M. *All-atom empirical potential for molecular modeling and dynamics studies of proteins*. *Journal of Physical Chemistry B*, 102(18):3586–3616, 1998. doi: 10.1021/jp973084f.

[278] Mackerell, A. D., Feig, M., and Brooks, C. L. *Extending the treatment of backbone energetics in protein force fields: Limitations of gas-phase quantum mechanics in reproducing protein conformational distributions in molecular dynamics simulation*. *Journal of Computational Chemistry*, 25(11):1400–1415, 2004. doi: 10.1002/jcc.20065.

[279] Essmann, U., Perera, L., Berkowitz, M. L., Darden, T., Lee, H., and Pedersen, L. G. *A smooth particle mesh Ewald method*. *The Journal of Chemical Physics*, 103(19):8577–8593, 1995. doi: 10.1063/1.470117.

[280] Ryckaert, J. P., Ciccotti, G., and Berendsen, H. J. *Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes*. *Journal of Computational Physics*, 23(3):327–341, 1977. doi: 10.1016/0021-9991(77)90098-5.

[281] Nosé, S. *A unified formulation of the constant temperature molecular dynamics methods*. *The Journal of Chemical Physics*, 81(1):511–519, 1984. doi: 10.1063/1.447334.

[282] Chi, C. N., Vögeli, B., Bibow, S., Strotz, D., Orts, J., Güntert, P., and Riek, R. *A Structural Ensemble for the Enzyme Cyclophilin Reveals an Orchestrated Mode of Action at Atomic Resolution*. *Angewandte Chemie - International Edition*, 54(40):11657–11661, 2015. doi: 10.1002/anie.201503698.

[283] Okada, T., Sugihara, M., Bondar, A. N., Elstner, M., Entel, P., and Buss, V. *The retinal conformation and its environment in rhodopsin in light of a new 2.2 Å crystal structure*. *Journal of Molecular Biology*, 342(2):571–583, 2004. doi: 10.1016/j.jmb.2004.07.044.

[284] Jo, S., Cheng, X., Lee, J., Kim, S., Park, S. J., Patel, D. S., Beaven, A. H., Lee, K. I., Rui, H., Park, S., Lee, H. S., Roux, B., MacKerell, A. D., Klauda, J. B., Qi, Y., and Im, W. *CHARMM-GUI 10 years for biomolecular modeling and simulation*. *Journal of Computational Chemistry*, 38(15):1114–1124, 2017. doi: 10.1002/jcc.24660.

[285] Kim, S., Lee, J., Jo, S., Brooks, C. L., Lee, H. S., and Im, W. *CHARMM-GUI ligand reader and modeler for CHARMM force field generation of small molecules*, 2017. doi: 10.1002/jcc.24829.

[286] Huang, J., Rauscher, S., Nawrocki, G., Ran, T., Feig, M., De Groot, B. L., Grubmüller, H., and MacKerell, A. D. *CHARMM36m: An improved force field for folded and intrinsically disordered proteins*. *Nature Methods*, 14(1):71–73, 2016. doi: 10.1038/nmeth.4067.

[287] Khajeh, K., Aminfar, H., Masuda, Y., and Mohammadpourfard, M. *Implementation of magnetic field force in molecular dynamics algorithm: NAMD source code version 2.12*. *Journal of Molecular Modeling*, 26(5):1–9, 2020. doi: 10.1007/s00894-020-4349-0.

[288] Phillips, J. C., Schulten, K., Bhatele, A., Mei, C., Sun, Y., Bohm, E. J., and Kale, L. V. *Scalable molecular dynamics with NAMD*. *Parallel Science and Engineering Applications: The Charm++ Approach*, 26(16):60–76, 2016. doi: 10.1201/b16251-15.

[289] Phillips, J. C., Hardy, D. J., Maia, J. D., Stone, J. E., Ribeiro, J. V., Bernardi, R. C., Buch, R., Fiorin, G., Hénin, J., Jiang, W., McGreevy, R., Melo, M. C., Radak, B. K., Skeel, R. D., Singharoy, A., Wang, Y., Roux, B., Aksimentiev, A., Luthey-Schulten, Z., Kalé, L. V., Schulten, K., Chipot, C., and Tajkhorshid, E. *Scalable molecular dynamics on CPU and GPU architectures with NAMD*. *Journal of Chemical Physics*, 153(4):44130, 2020. doi: 10.1063/5.0014475.

[290] Durrant, J. D. and McCammon, J. A. *Molecular dynamics simulations and drug discovery*. *BMC Biology*, 9(January 2015):71, 2011. doi: 10.1186/1741-7007-9-71.

[291] Besaw, J. E., Booth, V., and Rowley, C. N. *A Computational and Experimental Study of the Structure of FOXl1 Protein*. *Biophysical Journal*, 108(2):374a, 2015. doi: 10.1016/j.bpj.2014.11.2054.

[292] Wells, M. M., Tillman, T. S., Mowrey, D. D., Sun, T., Xu, Y., and Tang, P. *Ensemble-Based Virtual Screening for Cannabinoid-Like Potentiators of the Human Glycine Receptor α1 for the Treatment of Pain*. *Journal of Medicinal Chemistry*, 58(7):2958–2966, 2015. doi: 10.1021/jm501873p.

[293] Parikh, N. D. and Klimov, D. K. *Molecular Mechanisms of Alzheimer's Biomarker FDDNP Binding to Aβ Amyloid Fibril*. *Journal of Physical Chemistry B*, 119(35):11568–11580, 2015. doi: 10.1021/acs.jpcb.5b06112.

[294] Raffaele, F. *Replica Exchange with Solute Tempering : application to the N-Terminal segment of human Aquaporin 4*. Msc thesis, Università degli studi di bari, 2015.

[295] Melvin, R. L., Gmeiner, W. H., and Salsbury, F. R. *All-atom molecular dynamics reveals mechanism of zinc complexation with therapeutic F10*. *Journal of Physical Chemistry B*, 120(39):10269–10279, 2016. doi: 10.1021/acs.jpcb.6b07753.

[296] Gaalswyk, K. and Rowley, C. N. *An explicit-solvent conformation search method using open software*. *PeerJ*, 2016(5):e2088, 2016. doi: 10.7717/peerj.2088.

[297] Timol, Z. *Chemical and Conformational studies of bacterial cell surface polysaccharide repeating units*. Msc thesis, University of Cape Town, 2017.

[298] Xiao, J. and Salsbury, F. R. *Molecular dynamics simulations of aptamer-binding reveal generalized allostery in thrombin*. *Journal of Biomolecular Structure and Dynamics*, 35(15):3354–3369, 2017. doi: 10.1080/07391102.2016.1254682.

[299] Godwin, R. C., Gmeiner, W. H., and Salsbury, F. R. *All-atom molecular dynamics comparison of disease-associated zinc fingers*. *Journal of Biomolecular Structure and Dynamics*, 36(10):2581–2594, 2018. doi: 10.1080/07391102.2017.1363662.

[300] Luo, D. and Haspel, N. *Multi-resolution rigidity-based sampling of protein conformational paths*. In *2013 ACM Conference on Bioinformatics, Computational Biology and Biomedical Informatics, ACM-BCB 2013*, pages 786–792. ACM Press, New York, New York, USA, 2013. doi: 10.1145/2506583.2506710.

[301] Kuttel, M., Gordon, M., and Ravenscroft, N. *Comparative simulation of pneumococcal serogroup 19 polysaccharide repeating units with two carbohydrate force fields*. *Carbohydrate Research*, 390(1):20–27, 2014. doi: 10.1016/j.carres.2014.02.026.

[302] Daday, C., Curutchet, C., Sinicropi, A., Mennucci, B., and Filippi, C. *Chromophore-Protein Coupling beyond Nonpolarizable Models: Understanding Absorption in Green Fluorescent Protein*. *Journal of Chemical Theory and Computation*, 11(10):4825–4839, 2015. doi: 10.1021/acs.jctc.5b00650.

[303] Weng, J., Yang, Y., and Wang, W. *Lipid regulated conformational dynamics of the longin SNARE protein Ykt6 revealed by molecular dynamics simulations*. *Journal of Physical Chemistry A*, 119(9):1554–1562, 2015. doi: 10.1021/jp5075708.

[304] Kuttel, M. M., Jackson, G. E., Mafata, M., and Ravenscroft, N. *Capsular polysaccharide conformations in pneumococcal serotypes 19F and 19A*. *Carbohydrate Research*, 406:27–33, 2015. doi: 10.1016/j.carres.2014.12.013.

[305] Biedermannová, L. and Schneider, B. *Structure of the ordered hydration of amino acids in proteins: Analysis of crystal structures*. *Acta Crystallographica Section D: Biological Crystallography*, 71(11):2192–2202, 2015. doi: 10.1107/S1399004715015679.

[306] Louros, N. N., Baltoumas, F. A., Hamodrakas, S. J., and Iconomidou, V. A. *A β-solenoid model of the Pmel17 repeat domain: Insights to the formation of functional amyloid fibrils*. *Journal of Computer-Aided Molecular Design*, 30(2):153–164, 2016. doi: 10.1007/s10822-015-9892-x.

[307] Godwin, R., Gmeiner, W., and Salsbury, F. R. *Importance of long-time simulations for rare event sampling in zinc finger proteins*. *Journal of Biomolecular Structure and Dynamics*, 34(1):125–134, 2016. doi: 10.1080/07391102.2015.1015168.

[308] Melvin, R. L. and Salsbury, F. R. *Visualizing ensembles in structural biology*. *Journal of Molecular Graphics and Modelling*, 67:44–53, 2016. doi: 10.1016/j.jmgm.2016.05.001.

[309] Dai, Y., Seeger, M., Weng, J., Song, S., Wang, W., and Tan, Y. W. *Lipid Regulated Intramolecular Conformational Dynamics of SNARE-Protein Ykt6*. *Scientific Reports*, 6(1):1–12, 2016. doi: 10.1038/srep30282.

[310] Melvin, R. L., Gmeiner, W. H., and Salsbury, F. R. *All-Atom MD Predicts Magnesium-Induced Hairpin in Chemically Perturbed RNA Analog of F10 Therapeutic*. *Journal of Physical Chemistry B*, 121(33):7803–7812, 2017. doi: 10.1021/acs.jpcb.7b04724.

[311] Godwin, R. C., Melvin, R. L., Gmeiner, W. H., and Salsbury, F. R. *Binding site configurations probe the structure and dynamics of the zinc finger of NEMO (NF-κB Essential Modulator)*. *Biochemistry*, 56(4):623–633, 2017. doi: 10.1021/acs.biochem.6b00755.

[312] Kuttel, M. M., Timol, Z., and Ravenscroft, N. *Cross-protection in Neisseria meningitidis serogroups Y and W polysaccharides: A comparative conformational analysis*. *Carbohydrate Research*, 446-447:40–47, 2017. doi: 10.1016/j.carres.2017.05.004.

[313] Barage, S., Kulkarni, A., Pal, J. K., and Joshi, M. *Unravelling the structural interactions between PKR kinase domain and its small molecule inhibitors using computational approaches*. *Journal of Molecular Graphics and Modelling*, 75:322–329, 2017. doi: 10.1016/j.jmgm.2017.06.009.

[314] Louet, M., Bitam, S., Bakouh, N., Bignon, Y., Planelles, G., Lagorce, D., Miteva, M. A., Eladari, D., Teulon, J., and Villoutreix, B. O. *In silico model of the human ClC-Kb chloride channel: Pore mapping, biostructural pathology and drug screening*. *Scientific Reports*, 7(1):1–15, 2017. doi: 10.1038/s41598-017-07794-5.

[315] Grouleff, J., Koldsø, H., Miao, Y., and Schiøtt, B. *Ligand Binding in the Extracellular Vestibule of the Neurotransmitter Transporter Homologue LeuT*. *ACS Chemical Neuroscience*, 8(3):619–628, 2017. doi: 10.1021/acschemneuro.6b00359.

[316] Karamzadeh, R., Karimi-Jafari, M. H., Saboury, A. A., Salekdeh, G. H., and Moosavi-Movahedi, A. A. *Red/ox states of human protein disulfide isomerase regulate binding affinity of 17 beta-estradiol*. *Archives of Biochemistry and Biophysics*, 619:35–44, 2017. doi: 10.1016/j.abb.2017.02.010.

[317] Louet, M., Labbé, C. M., Fagnen, C., Aono, C. M., Homem-de Mello, P., Villoutreix, B. O., and Miteva, M. A. *Insights into molecular mechanisms of drug metabolism dysfunction of human CYP2C930*. *PLoS ONE*, 13(5):e0197249, 2018. doi: 10.1371/journal.pone.0197249.

[318] Hlozek, J., Kuttel, M. M., and Ravenscroft, N. *Conformations of Neisseria meningitidis serogroup A and X polysaccharides: The effects of chain length and O-acetylation*. *Carbohydrate Research*, 465(May):44–51, 2018. doi: 10.1016/j.carres.2018.06.007.

[319] Asakura, T., Nishimura, A., and Tasei, Y. *Determination of Local Structure of 13C Selectively Labeled 47-mer Peptides as a Model for Gly-Rich Region of Nephila clavipes Dragline Silk Using a Combination of 13C Solid-State NMR and MD Simulation*. *Macromolecules*, 51(10):3608–3619, 2018. doi: 10.1021/acs.macromol.8b00536.

[320] Banner, D. W., Gsell, B., Benz, J., Bertschinger, J., Burger, D., Brack, S., Cuppuleri, S., Debulpaep, M., Gast, A., Grabulovski, D., Hennig, M., Hilpert, H., Huber, W., Kuglstatter, A., Kusznir, E., Laeremans, T., Matile, H., Miscenic, C., Rufer, A. C., Schlatter, D., Steyaert, J., Stihle, M., Thoma, R., Weber, M., and Ruf, A. *Mapping the conformational space accessible to BACE2 using surface mutants and cocrystals with Fab fragments, Fynomers and Xaperones*. *Acta Crystallographica Section D: Biological Crystallography*, 69(6):1124–1137, 2013. doi: 10.1107/S0907444913006574.