# A Monte-Carlo-based torsion construction algorithm for ligand design

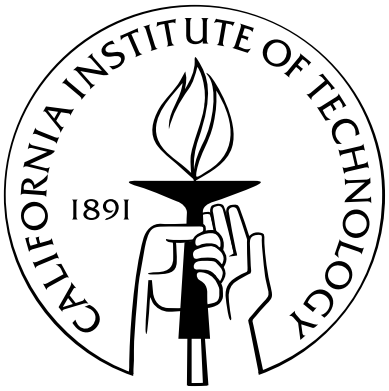
Thesis by

Peter Michael Kekenes-Huskey

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy




California Institute of Technology

Pasadena, California

2008

(Defended 12 September 2008)

# Acknowledgements

This thesis is dedicated foremost to my brother Patrick, who passed away on March 10th, 2006. Through the grief that ensued I learned to embrace life and endure its hardships. My parents and their unwavering support were essential in overcoming this adversity and maintaining course on my studies.

I would like to thank my advisor, Professor Bill Goddard, for dedicating his time and commitment to guiding me through the complexities of computation. His insight into topics ranging from material science to biological systems is remarkable and if a person manages to absorb even ten percent of this knowledge, he or she will certainly have a bright career. I also owe much of my intellectual development to Dr. Nagarajan Vaidehi, Professor Bert Holmes and David Weaner. Vaidehi was not only adept at parsing a complex problem into manageable chunks, she added a personal touch that is often lacking in academics. Bert was an exceptional professor of physical chemistry and urged me to apply to an upper-tier university for graduate school, which ultimately lead me to attend Caltech. I was originally leaning toward a Big Ten school where I could regularly attend football games in the fall. Dave's passion for science and dedication to his students motivated me to strive for excellence and learn from failure, which were key for helping me overcome my tumultuous start to high school.

Finally, I thank Jesica for her love and patience. She has been amazingly understanding and helpful in my pursuit of this lifetime goal and I eagerly want to return the favor by giving her the attention she truly deserves.

## 0.1 Thesis Overview

This document explains the my development of a molecular modeling package that optimizes the dihedral angles of a ligand within a protein environment. Conformation optimization is a subset of the docking problem, in which computation estimates the binding pose of a ligand in a host receptor. As the approach is substantially different from established methodologies, Chapter 2 is dedicated to describing the machinery that comprises my method. Although it is a lengthy discussion, its purpose is to illustrate the considerable amount of algorithmic and conceptual design that must be addressed before the method can be applied to real problems. Chapter 4 highlights the performance and optimization of the method, as well as its application to a relevant problem in drug design. Chapter A lists several of the key algorithms for managing and classifying molecular structures. They are not rigorously proven to any degree, but they demonstrate the strong role mathematics plays in designing efficient and elegant solutions to problems in computational chemistry.

# Contents

# List of Figures

.

# Abstract

A wealth of computational strategies [1, 2, 3, 4] is available for predicting the binding site and affinities of a putative ligand inside a target receptor. Although numerous techniques focus on the orientation of ligands or fragments thereof, few methods have delved into improving the accuracy of generating reliable ligand conformations within predicted binding modes. In an effort to comprehensively sample the torsion space available to a flexible ligand and focus on low-energy conformations, a recursive, Metropolis Monte Carlo (MC)-based rotamer design protocol has been developed. This approach recursively samples adjacent rotatable bonds from a defined anchor and directs the search along low-energy pathways, such that high-affinity conformations of the ligand can be identified. Furthermore, this program applies spatial constraints within the search that restrict the solutions to structurally dissimilar conformations, thus encouraging a diverse solution set. The performance of *moleculeGL* has been evaluated for a set of 55 co-crystals, for which the number of rotatable bonds ranged from 2 to 32. Approximately 90 percent of the structures are predicted within 2.0 $Å^2$ root mean square deviations (RMSD) with respect to the crystal structure, starting from an arbitrary ligand conformation. This level of accuracy suggests the program's applicability to the design of pharmacaphore substituents, for which the position of a chemically active pharmacaphore is well-known.

# Chapter 1

# Introduction

## 1.1 Background

The binding of small molecules to proteins of pharmacological importance has in the last decade become one of the most exciting and pursued avenues in theoretical chemistry. In its wake, a plethora of computational suites have flooded this market, featuring predictive algorithms that help uncover the binding orientation and relative affinities of subject ligands [5]. Direct drug design relies heavily on using the refined atomistic coordinates of a receptor structure to scan for plausible ligand positions and isolate those conducive to binding.

Current computational strategies [1, 2, 3, 4] are comprised roughly of the following stages: orientation search of the ligand, conformation search, and scoring. The orientation search primarily scans the free volume within a receptor for pockets sufficiently large to accommodate the ligand. The conformation search entails varying bond lengths, angles, and dihedrals to optimize the ligand's binding affinity; however, because bond stretching and angle bending modes are relatively stiff, ligand flexibility is driven largely by variation of the dihedral angles [6]. Scoring implies evaluating the interaction of the ligand with the receptor, enabling the elimination of unfavorable conformations, and retaining only the reasonable solutions.

Sampling packages differ in how these stages are approached. The orientation search can vary from docking the whole ligand or only parts thereof. The conformation search may vary bonds, angles, and dihedrals systematically, dynamically, or via

Monte Carlo (MC). Scoring approaches range from explicitly quantifying both valence and nonbond interactions to those that are based on purely statistically observed atom distances [1, 2, 3, 4].

## 1.2 Related Studies

An excellent review by Brooijman and Kuntz[5] classifies ligand flexibility methods as systematic, stochastic, or genetic. The more popular packages focused on direct drug design, including DOCK[4, 7], FlexX[1, 2, 8], ICM[9], Autodock[10], and Gold[3] successfully employ these approaches for ligand flexibility. DOCK is a forefather of docking suites developed by Kuntz and coworkers that relies on a sphere matching algorithm [4, 11] for docking a putative ligand within a fixed protein. Ligand flexibility is addressed in one of two modes: 1) a simultaneous search in which the ligand conformations are generated *outside* the protein and docked as rigid bodies, or 2) a fragment-based search by which rigid clusters of the ligand are docked independently and substituents are reconstructed to form a complete molecule. This incremental construction method employs a greedy algorithm, for which the best conformations from a given search level are passed to the subsequent stage in the search. FlexX [1, 8] also features an incremental construction algorithm, but utilizes a statistical database of torsions[12] for determining the relative orientation of fragments. In contrast, systematic algorithms randomly sample a subspace of the ligand conformation space to obtain low energy candidates, and may, in the case of evolutionary algorithms, further employ a fitness function to propagate solutions as viable progeny. Of these approaches, ICM is a powerful method that utilizes a MC approach reliant on biased probabilities from data like Ramachandran plots[9]. Finally, evolutionary approaches like Gold[3] and Autodock[10] use genetic algorithms coupled with random torsions to introduce ligand flexibility.

To compare the accuracy of these and similar algorithms, the predicted ligand conformation is compared with a high-resolution X-ray structure of the protein-ligand co-crystal. The aforementioned programs have performed remarkably well for predict-

ing the ligand poses within 2.0 $\mathring{A}$ root mean square deviations (RMSD) of the crystal structure, which is sufficiently close [13, 3] for convergent thermodynamic properties.

As shown in Table 1.1, these methods generally perform strongly for compounds with fewer than ten rotatable bonds, but as the number of bonds increases, the performance rapidly declines. Given that the smaller ligands are predicted well, it is evident that the protocols for placing the anchors is accurate, therefore it is the sampling of lengthy substituents that is the primary source of error, which necessitates an improvement of the conformation search protocols.

To optimize the binding interactions, it is crucial to have ligand conformations that accurately reflect plausible binding modes. The binding mode of a ligand is primarily determined by the position of the pharmacaphore, which is the set of atoms chiefly responsible for bioactivity. Nonbond interactions, like hydrogen bonds and van der Waals, strongly constrain the pharmacaphore and thus accurate sampling of all connected substituents is needed to identify optimal binders. The slightest variations in the torsional conformation can give rise to large fluctuations in the apparent interaction energy that may obscure the nature of binding. Unfortunately, despite exhaustive annealing molecular dynamics and minimization, oftentimes the local barriers to the global minimum are not readily overcome, thus a superior sampling approach is needed.

## 1.3 Contribution

The Metropolis MC-based protocol, *moleculeGL*, was developed for optimizing the conformations of substituents adjoined to fixed pharmacaphores. This strategy features an explicit nonbond force field with hydrogen bonding terms, as well as methods to sample low-energy regions of the protein. Diversity and filtering strategies are introduced for retaining an exhaustive set of conformations while pruning unphysical solutions. This method has been tested in predicting X-ray co-crystals from the Protein Databank (PDB) [15]. This algorithm can be used in conjunction with an orientation search procedure to generate viable solutions for the general docking

problem.

| | | Parameters | | | | |
|---|---|---|---|---|---|---|
| code | $N_r$ | MolDock | Glide | GOLD | FlexX | Surflex |
| 1nco | 8 | 0.39 | 6.99 | n/a | 5.85 | 8.26 |
| 1acm | 6 | 0.56 | 0.29 | 0.81 | 1.39 | 1.43 |
| 1rds | 8 | 4.34 | 3.75 | 4.78 | 4.89 | 9.83 |
| 1snc | 6 | 1.69 | 1.91 | n/a | 7.48 | 4.92 |
| 1atl | 9 | 1.59 | 0.94 | n/a | 2.06 | 7.01 |
| 1baf | 7 | 1.6 | 0.76 | 6.12 | 8.27 | 6.52 |
| 1stp | 5 | 0.76 | 0.59 | 0.69 | 0.65 | 0.51 |
| 1bbp | 11 | 0.99 | 4.96 | n/a | 3.75 | 1.07 |
| 1tka | 8 | 1.35 | 2.28 | 1.88 | 1.17 | 1.96 |
| 1bma | 12 | 1.04 | 9.31 | n/a | 13.41 | 1 |
| 1tmn | 13 | 5.58 | 2.8 | 1.68 | 0.86 | 1.30 |
| 1cbs | 5 | 1.43 | 1.96 | n/a | 1.68 | 1.77 |
| 1cbx | 5 | 1.06 | 0.36 | 0.54 | 1.35 | 0.7 |
| 1trk | 8 | 0.73 | 1.64 | n/a | 1.57 | 1.22 |
| 1dwd | 9 | 1.07 | 1.32 | 1.71 | 1.66 | 1.68 |
| 1eap | 10 | 2.52 | 2.32 | 3 | 3.72 | 4.89 |
| 1epb | 5 | 3.35 | 1.78 | 2.08 | 2.77 | 2.87 |
| 2cgr | 7 | 0.92 | 0.38 | 0.99 | 3.53 | 1.63 |
| 1etr | 9 | 1.96 | 1.48 | 4.23 | 7.24 | 4.05 |
| 2cmd | 5 | 0.5 | 0.65 | n/a | 3.75 | 1.60 |
| 1fkg | 10 | 1.89 | 1.25 | 1.81 | 7.59 | 1.81 |
| 2dbl | 6 | 1.55 | 0.69 | 1.31 | 1.49 | 0.81 |
| 1frp | 6 | 0.92 | 0.27 | n/a | 1.89 | 0.75 |
| 1glq | 13 | 7.09 | 0.29 | 1.35 | 6.43 | 5.68 |
| 1hdc | 6 | 1.71 | 0.58 | 10.49 | 11.74 | 1.8 |
| 2r07 | 8 | 1.81 | 0.48 | 8.23 | 11.63 | 1.35 |
| 1hri | 9 | 6.33 | 1.59 | 14.01 | 10.23 | 1.98 |
| 2sim | 5 | 1.29 | 0.92 | 0.92 | 1.99 | 1.10 |
| 3cpa | 5 | 1.63 | 2.4 | 1.58 | 2.53 | 1.90 |
| 1hyt | 5 | 1.61 | 0.28 | 1.1 | 1.62 | 0.55 |
| 3tpi | 6 | 0.36 | 0.49 | 0.8 | 1.07 | 0.52 |
| 1lic | 15 | 2.44 | 4.87 | 10.78 | 5.07 | 3.46 |
| 4dfr | 9 | 1.39 | 1.12 | 1.44 | 1.4 | 1.60 |
| 1lna | 8 | 3.04 | 0.95 | n/a | 5.4 | 0.88 |
| 1lst | 5 | 0.23 | 0.14 | 0.87 | 0.71 | 0.33 |
| 8gch | 7 | 4.07 | 0.3 | 0.86 | 8.91 | 4.51 |

**Table 1.1**: RMSDs of predicted cocrystal complexes from several leading docking suites. Table borrowed from [14] except that structures with fewer than five rotatable bonds are excluded.

# Chapter 2

# *moleculeGL* Model

## 2.1 Introduction

*moleculeGL* comprises a suite of functions that are geared toward evaluating protein-ligand interactions. Of these, its primary utility is the MC-based, torsion angle sampling algorithm for ligand design. Whereas many docking applications are geared toward identifying the placement and orientation of ligands, *moleculeGL* optimizes the dihedral angles of a suitably docked ligand to maximize binding interactions. This is done by exhaustively sampling the ligand's rotatable bonds in the presence of the protein, such that an ensemble of strongly interacting conformations are obtained. An example of this strategy is depicted in Fig. 2.1.

Drug design is a hierarchical process that can be described in three steps. The first step, docking, identifies the general placement of a ligand in a protein binding site. Typically the goal is to place molecular fragments known as pharmacaphores, which are groups of atoms that are responsible for the biological activity of a compound. Pharmacaphores are often rigid units, such as heterocycles, that ultimately anchor the ligand and determine the placement of flexible substituent groups. As such, we refer to these as anchors. The second step is the optimization of the ligand conformation within the binding site, which typically involves optimizing bond lengths, angles, and torsions. The final stage is molecular refinement via molecular dynamics (MD), through which thermodynamic analysis can be performed. It is the second step where *moleculeGL* is a powerful tool.

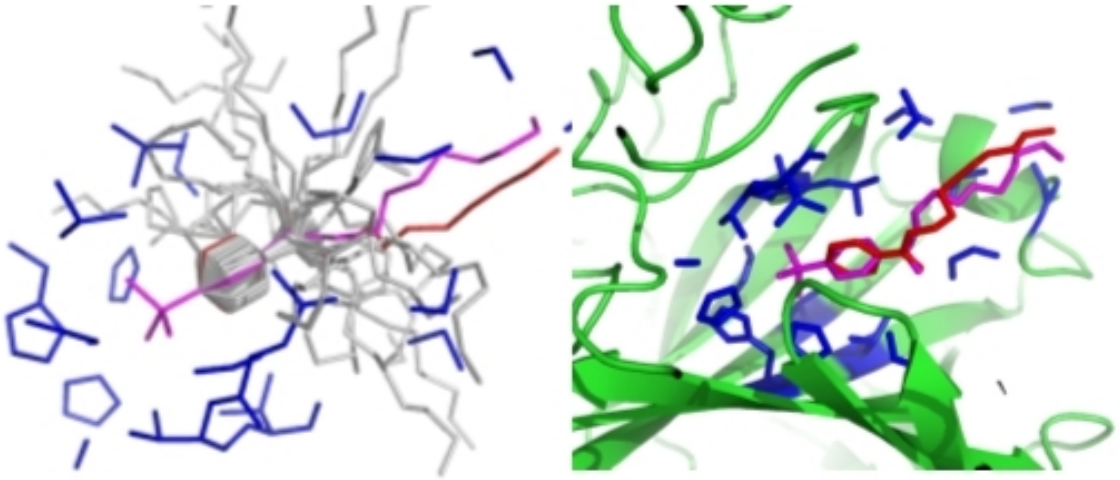


**Figure 2.1**: A depiction of torsion sampling for which a ligand (pink) is optimized in a receptor binding site (blue). Included are **(a.)** an ensemble of generated ligand conformations (gray) and **(b.)** the nearest conformation compared to the co-crystal (red)

*moleculeGL* accepts as input a collection of putative ligand positions and recursively *grows in* the substituent groups from a defined anchor. That is, given an anchor position, the ligand torsion angles are varied to improve compatibility with the binding site. The engine aims to balance nonbond terms like hydrogen bonding interactions, as well as ensuring adequate burial of ligands in the binding domain. The resulting rotamers, that is, the ensemble of conformations generated by *moleculeGL*, are prime starting configurations for further refinement and analysis. Illustrated in Fig. 2.2 is a simplified layout of the protocol, whereby a set of anchors are loaded, sampled, and sorted according to diversity, yielding a set of fully constructed rotamers.

*moleculeGL* was written as a standalone utility from the ground up and thus borrows little from other molecular mechanics routines. In developing a program from scratch, a host of topics must be addressed, ranking from graph theoretic algorithms for ring identification to schema for managing large data sets. The problems, as well as their computationally efficient solutions, are outlined in this chapter.

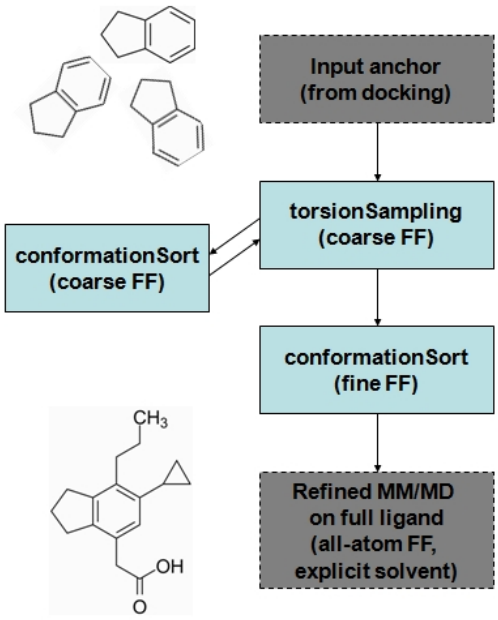



**Figure 2.2**: A flowchart depicting the stages of the *moleculeGL* torsion sampling algorithm (blue). Given an anchor starting position (from docking, for instance) the protocol iteratively samples each torsion angle (TorsionSampling) while maintaining a structurally diverse set of conformations (ConformationSort). A sampled pose may then be further refined using molecular mechanics or dynamics routines

The discussion begins with a quick summary of computational considerations in Section 2.2. The remainder of the chapter explicates the program flow shown in Fig. 2.3. In Section 2.3, the steps taken to prepare protein and ligand structure files for sampling are outlined. Section 2.4 discusses the torsion sampling protocol, Section 2.5 addresses the conformation sorting approaches for maintaining large rotamer sets, and Section 2.6 describes the scoring routines. In Section 2.7, the interplay of the torsion sampling, conformation sorting, and scoring is introduced. Lastly, miscellaneous topics including additional features and software architecture are explained in Section 2.8.

As a final note, an auxiliary aim of this chapter is to provide detailed documentation of the code for potential end users. Therefore, in the course of this discussion, parameters are listed that shape the program performance. In this way, a

user may refer to this chapter as a complement to the web-based documentation at http://pkh.caltech.edu/mgldocs/. A list of commands is provided in Section A.1.

## 2.2 Computational Complexity and Notation

Much of the model discussion revolves around the concept of computational expense, which describes the amount of computing needed to accomplish a defined task. The goal is to craft algorithms that can perform a task in the fewest number of operations. This is of vital importance for many functions in this program, such as bond rotation, which may be executed millions of times for a single run. Thus, any reduction in the computational complexity, that is, the total number of operations, can offer considerable performance improvements.

Oftentimes these performance gains are obtained by increasing the amount of information stored in memory. For example, a list of nearby atoms may be stored for rapidly computing pairwise interactions, as opposed to computing distances at each scoring call. Equivalently, a reduction in memory consumption usually requires additional processing, such as converting a compact data structure into practical form amenable to computation. These issues of computation versus storage must be balanced in order to develop a program that can operate in a reasonable time frame given finite memory resources. Thus in this chapter the sundry approaches pursued for optimizing the accuracy and performance of *moleculeGL* are explained.

As for notation, big O descriptions such as $\mathcal{O}(n^2)$ are used throughout this document. This expression, for instance, states that the run time of a task is strictly bounded from above and below by some constant times $n^2$, that is,

$$an^2 \leq \mathcal{O}(n^2) \leq bn^2; \quad a \leq b. \tag{2.1}$$

This is useful in comparing the complexity of algorithms whose run times can be described in polynomial form. Although other bounding nomenclature exists, they are not utilized in this discussion and are thus omitted.

Lastly, some algorithms in this text are succinctly expressed in terms of set theory or graph theory. Any introductory algorithm textbook should provide a sufficiently comprehensive explanation of these concepts and associated vocabulary.

As a further note to aid in the discussion of the model, functions, parameters and parameter option names are denoted in the sans-serif font. Functions performing operations on data and are written in plain text like Sort. Parameters governing the operation of a function and boldfaced, such as **scaleVDWradii**, while parameter options are designated with italics, such as *diversityFixed*.

## 2.3 Structure Preparation

The first part of the model discussion addresses ligand preparation, which is represented by the blue region in Fig. 2.3. *moleculeGL* accepts as input a molecular structure file that contains a list of atoms, positions, chemical forcefield type, and bonds. The first task is thus to parameterize the molecule's valence (bonds, angles, torsions) and nonbond terms (van der Waals (VDW) hydrogen bonds, Coulomb) with a forcefield such as Dreiding [16]. The second task is to represent the molecule in a form suitable for simulation.

A molecule consists of groups of atoms that may either be rigidly constrained, such as in heterocyclic rings, or flexible, such as in alkyl chains. At room temperature, it is safe to assume that bonds and angles are fixed, while torsional degrees of freedom are readily accessible at room temperature. Therefore, a chemically intuitive description of the molecule is one in which the rigidly connected atoms are described as clusters and the bonds between and stemming from these clusters are freely rotatable hinges, as demonstrated in Fig. 2.4(b). With this representation, torsion sampling is done by varying the dihedral angle of the hinge. Listed in Table 2.1 are parameters for initialization.

| Parameters | |
|---|---|
| fixedBond | define an otherwise freely rotatable bond as fixed |
| parameterFile | define parameter file to use |

**Table 2.1**: Parameters for ligand parameterization and rigid-body clustering

### 2.3.1 Cluster determination

The process of classifying rigid bodies from a bond list is described as rigid-body clustering. *moleculeGL* employs a graph theoretic approach that performs this clustering from an atom connectivity list in $\mathcal{O}(NlgN)$ time. This requires that the bonds be assigned to one of three classes: *fixed, rotatable*, and *freely rotatable*, using the torsion periodicity and barrier height information defined in a forcefield (FF), in addition to user-supplied information.

*Fixed* bonds are those that either cannot freely rotate, such as those belonging to a ring, or are rotationally symmetric, such as the C-N bond in a cyanyl group. *Freely rotatable* bonds refer to those for which the barrier to rotation is negligible, thus permitting free rotation at room temperature, such as in the case of $sp^3 - sp^3$ bonds. Bonds that ordinarily do not rotate but have discrete isomers are labeled as *rotatable*. The user may also explicitly classify a given bond, which may be desired for cases that prefer an isomer, such as a peptide chains.

To better illustrate the algorithms introduced in this section, a fictitious molecule is proposed in Fig. 2.4(a). This molecule includes a fused ring, multiple substituents and branched alkyl groups, which pose unique challenges for clustering. The objective is to appropriately assign atoms and rotatable bonds to clusters or hinges.

The clustering algorithm consists of two key functions, ClusterEdges and RingTest, which are outlined as pseudocode below (see Alg. 2 and Alg. 1 in Appendix) and depicted in Fig. 2.5. In essence, the algorithm represents the molecule as a connected graph, $G$, where the atoms are in the vertex set, $V$, and the bonds in edge set, $E$. Employing a graph theoretic approach, cycles (rings) are identified and condensed into single vertices called supervertices. Lastly, supervertices and remaining vertices are incrementally labeled as clusters, which leaves the remaining edges as hinges.

Specifically, Alg. 2 first consolidates the vertices representing a fixed edge into a supervertex. Next, a minimum spanning tree (MST), $T$, is generated, which is the subset of $G$ that connects all $V$ with a minimum number of edges ($|V|-1$). Referring to Fig. 2.5, for instance, in the first step the bond **a** of the fused four-member ring and bond **b** of the benzene ring share a common vertex **10**. In the second step, the spanning tree shows the shared bond **b** that is linked to bond **a** by vertex **10**. The MST can be computed in a variety of ways [17] in roughly $\mathcal{O}(n\,lg\,n)$ time.

By design, the MST guarantees an acyclic graph, which is enforced as edges are incrementally added to the tree. Thus the task remains of identifying the omitted edges that would otherwise create a cycle. The MST algorithm implemented in *moleculeGL* defines a set $M$ that contains all vertices with a degree of connectivity greater than 2. Therefore, if any member of $M$ is also a terminal vertex of a given branch in the MST, the edges linking these instances form a cycle. Referring to §3 of Fig. 2.5, vertex **10** is a member of $M$ and is also a terminal vertex, therefore the linking edges comprise a cycle. The vertices corresponding to these edges are then condensed into a supervertex.

After the supervertices are defined, Alg. 2 assigns a cluster number to each supervertex and each remaining vertex. All remaining edges represent the hinges between clusters and are subject to torsion sampling.

### 2.3.2 Internal coordinates

The clusters identified in the previous step are stored to a structured array that preserves the sequential ordering. Within the collection of clusters, one anchor cluster is designated as the lead cluster. Relative to the lead, all remaining clusters are designated as downfield and their positions are defined relative to the preceding upfield clusters. More precisely, the position of a downfield cluster is uniquely described by the torsion angle of hinge linking it to the upfield cluster. Therefore, given the Cartesian coordinates of the lead cluster, a ligand conformation can be compactly represented as a series of recursive hinge rotations. This definition underlies an inter-

nal coordinates (IC) description that reduces storage requirements to $(n-1)$ from $3n$ for the equivalent Cartesian representation. As will be shown in Section 2.3.3, this description also facilitates an efficient sampling method.

Another data structure, the clusterTree, enables the mapping of the IC conformation to Cartesian coordinates. This structure explains the atom membership of each cluster as well as the linkage between adjacent clusters. In practice, the conversion procedure applies a sequential series of rotations to a copy of the reference ligand, using the clusterTree as a guide.

Specifically, to generate the position of cluster $n$, the position of the hinge between the $n^{th}$ and $(n-1)^{th}$ clusters is recalled. For the sake of discussion, the atoms comprising this bond are labeled as the fixed $x_f$ and rotated $x_r$ atoms. A rotation operation about the axis defined by the vector $x_f\vec{x}_r$ is applied to all atoms belonging downfield to the $(n-1)^{th}$ cluster. This process is repeated for all subsequent clusters until the entire ligand is reconstructed. Overall, this compact notation substantially reduces storage requirements but presents a higher computational cost.

A Cartesian representation of an IC conformation is required for scoring pairwise interactions between atoms. A brute force approach could rebuild the entire conformation by applying rotations to each hinge of the IC in succession. However, as will be discussed below, the coordinates of only one cluster are required for a given sampling iteration. Therefore a method of determining a cluster position with a minimal number of operations is introduced.

Given the clusterTree setup described above, the position of a given cluster $n$ is ultimately determined by the position of its parent cluster, $n-1$, or more specifically, the bond between $x_f$ and $x_r$. Thus by retaining these Cartesian coordinates, the position of cluster $n$ can be determined up to a rotation. By retaining the position of two additional atoms comprising the dihedral, $\mathbf{x}'_f$ and $\mathbf{x}'_r$, the relative orientation can be exactly determined. Therefore, given the position of these four atoms and the accompanying value for its torsion angle, cluster $n$ can be determined without operating on the entire ligand, given the following protocol described in Alg. 3 of the Appendix.

### 2.3.3 Quaternion-based rotation

A rotation about a torsion can be expressed as a product of a point in $\mathcal{R}^3$ with a $3 \times 3$ rotation matrix, $R$. Typically, $R$ is defined in terms of the product of Euler matrices, which perform rotations about an orthogonal set of axes. Rotation about an arbitrary axis thus generally requires three operations around orthogonal axes. Not only is this numerically expensive, the reliance on orthogonal rotations is furthermore prone to a phenomenon called Gimbal [18] locking, in which the rotated vector is essentially locked along an axis. This behavior is due to singularities that may arise in the rotation matrix.

To circumvent these shortcomings, quaternion math enables one to perform rotations about an arbitrary axis and is not subject to singularities. [18] A quaternion is a complex number in $\mathbb{R}^4$ and for the purpose of rotation, it may be defined as

$$\mathbf{q} = \left( cos\frac{\theta}{2},\ sin\frac{\theta}{2} \cdot \mathbf{u} \right), \tag{2.2}$$

where $\mathbf{u} \in \mathcal{R}^3$ represents the rotation axis and $\theta$ is the angle of rotation. This extension of complex numbers allows rotation of point $x$ to point $x'$ via the relation

$$\mathbf{x}' = \mathbf{q} \cdot \mathbf{x} \cdot \mathbf{q}^{-1}. \tag{2.3}$$

Although quaternion multiplication involves more operations than a traditional matrix product, it still requires fewer operations than the equivalent rotation with Euler matrices, thus offering substantial improvements in performance and stability. This inexpensive shortcut to rotation is widely used throughout the *moleculeGL* sampling protocol.

### 2.3.4 Valence list generation

As will later be described in Section 2.6, computing intramolecular energies requires the identification of all bonds, angles, and torsions. Not only are these lists essential for scoring valence energies, but the atom pairs belonging to these lists $(i+1) \ldots (i+$

3) must be ignored when assessing nonbond interactions. Described in Alg. 4 is an elegant set-based method for identifying these groups of atoms. First, a set $E'$ containing all 'multiply-connected' edges is defined; a multiply connected edge, $\mathbf{e}'_{ij}$, is an edge for which both vertices, $v_i, v_j$ are connected to some other edge, $e_{ki}, e_{jl}$. A multiply-connected edge and its two complement edges form a tuple that represents a dihedral angle. Since a dihedral angle consists of two angles, splitting each tuple into pairs of adjacent edges yields the angles formed by those edges.

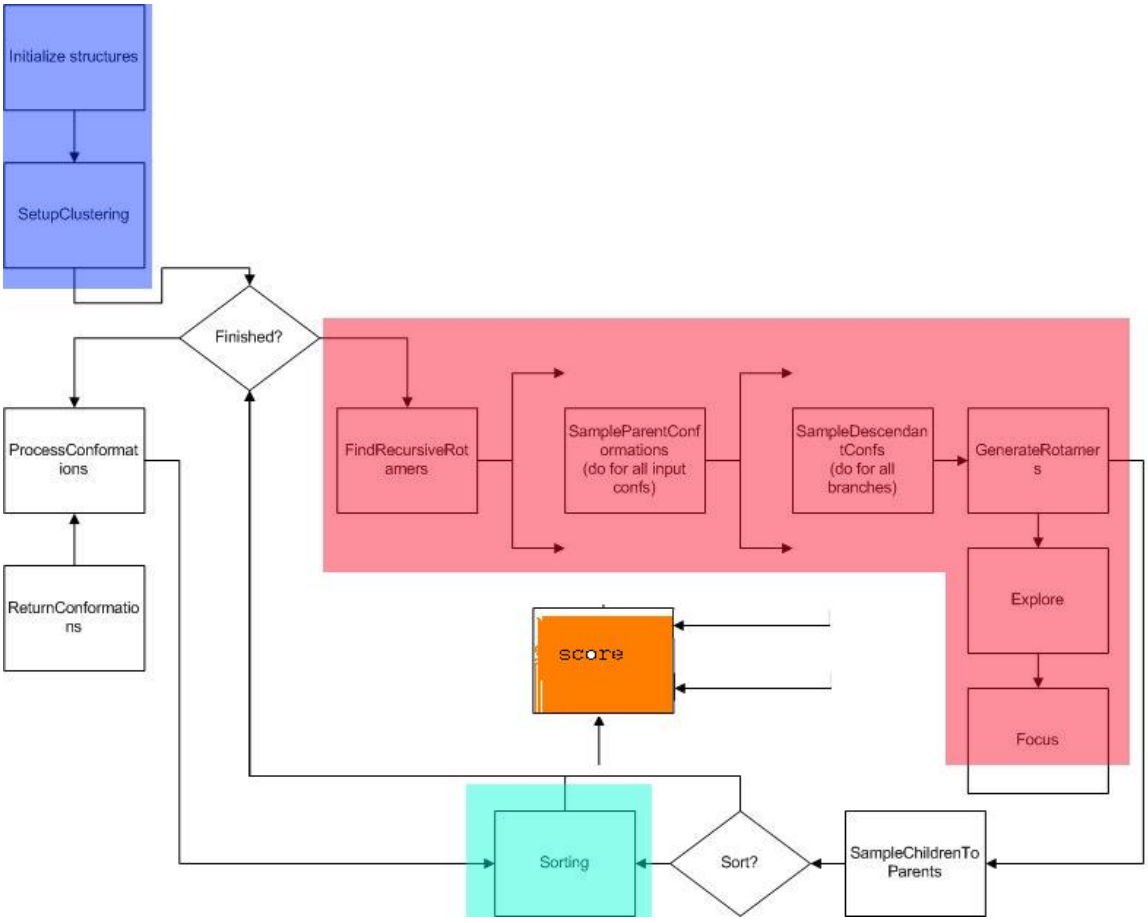
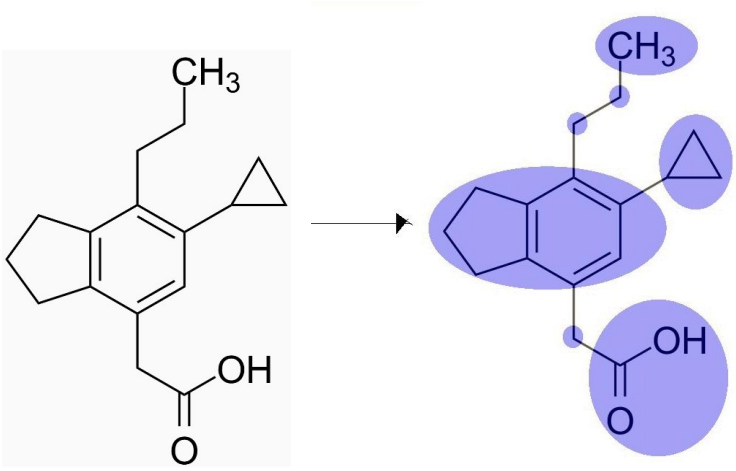



**Figure 2.3**: This *moleculeGL* flowchart is colored according to the model discussion, including Ligand Preparation (blue), TorsionSampling (red), ConformationSort (light blue), and Scoring (orange). The open brackets designate loops over all input parent conformations




(a) (b)

**Figure 2.4**: A fictitious molecule is introduced to further illustrate the rigid-body clustering process, which is necessary for identifying rotatable groups of atoms. The circled substituents (blue) represent sets of atoms that are automatically identified as clusters

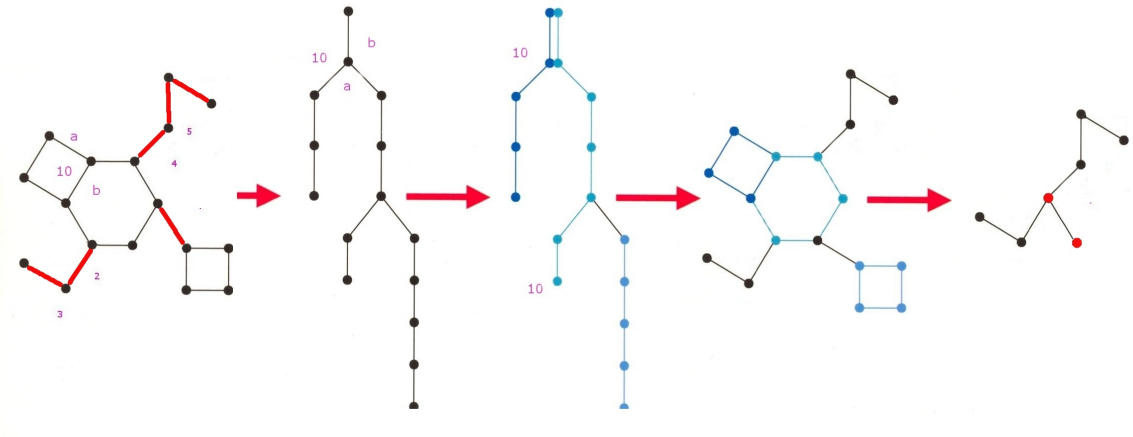



**Figure 2.5**: A step-wise representation of the clustering process that distinguishes rings from rotatable bonds. **(1.)** classifies the bond types, **(2.)** computes the minimum spanning tree, while **(3,4.)** involve ring determination and **(5.)** assigns cluster numbers to vertices and supervertices

## 2.4 TorsionSampling

With a clustering description and rotation protocol in place, the discussion turns to torsion sampling. This task describes the construction of a ligand within the protein environment by successively sampling each torsion. Unlike traditional MC, for which a known ligand position is optimized, torsionSampling assumes only the lead cluster position is defined. As such, all downfield clusters must be recursively sampled from this cluster.

This topic requires three issues to be addressed: the manner by which a single rotatable bond is sampled (Section 2.4.1), the direction sampling proceeds given a collection of clusters (Section 2.4.3), and the sampling approach given a sampling direction (Section 2.4.2).

### 2.4.1 Sampling of single iteration

Successful torsion sampling begins with accurate and sufficient sampling at each rotatable bond. This requires the identification of local minima but the search must also retain a diverse distribution to discourage bias. Since the position of the $k^{th}$ cluster ultimately determines the placement of subsequent clusters, any bias could make a fully constructed solution impossible. This is especially true given that a structure at its global minimum does not guarantee that its individual torsions are at local minima.

*moleculeGL* accomplishes this by dividing the sampling into a two-stage process shown in Fig. 2.7. The first stage, Explore, scans the nearby space at regular intervals to detect bad contacts. The latter stage, Focus, performs MC sampling within the favorable sectors identified by Explore. The number of rotamers generated in these functions are determined by **numRotamers** if the bond is freely rotatable and by **bondRotamers**, otherwise. The parameters governing these functions are listed in Table 2.2 and explained in the following sections.

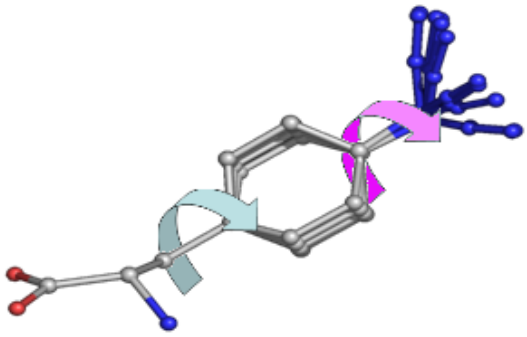


**Figure 2.6**: An example of torsion sampling in which iteration 1 samples the ring and iteration 2 rotates the azide)

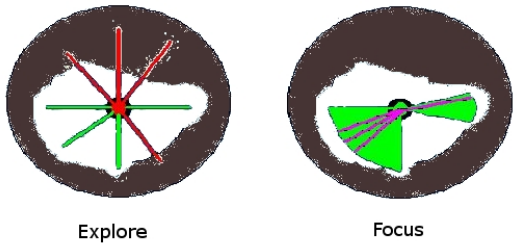



**Figure 2.7**: A two-dimensional representation of the Explore and Focus sampling approaches. Explore systematically rotates the downfield cluster over set intervals and computes the potential energy. Focus performs Monte Carlo sampling within energetically favorable regions

#### 2.4.1.1 Explore

Using the hinge between two clusters as an axis, the Explore stage rotates and scores the downfield cluster in fine increments over $[0, 2\pi]$ determined by **numExploreRotamers**. Each rotamer is scored based on the local environment of the current cluster, thus all other downfield clusters are ignored. These scores are used to identify favorable sampling regions by grouping the adjacent, low-energy rotamers. For example, if a bond were explored at 30 degree intervals and favorable rotations were found at 0, 30, 60, 120, and 150 degrees, this would result in two sectors spanning 0–60 and 120–150 degrees. A low-energy rotamer is one for which the average energy

| Parameters | |
|---|---|
| bondRotamers | number of conformations sampled for a freely rotatable bond. |
| focusSampling | after performing explore sampling, focus search in energetically favorable regions |
| numRotamers [1] | defines the expected number of rotamers returned for each bond (use this in favor of explore/focus functions) |
| numExploreRotamers | number of rotamers sampled during explore search |
| numFocusRotamers | number of rotamers to be accepted for each parent conformations in focus search |

**Table 2.2**: Parameters that shape the number of conformations generated at a given sampling iteration

of the sampled range (omitting clashing structures) is below a certain value.

#### 2.4.1.2 Focus

MC is performed in the sectors identified by Explore, which have a high probability of yielding physically relevant rotamers. The number of rotamers retained for a given sector is weighted based on the number of degrees the sector spans multiplied by **numFocusRotamers**. Using the example from above, sixty percent of the total would be requested from the 0–60 sector and only forty percent from the 120–150 increment. The Metropolis MC protocol used by Focus is described below.

**Metropolis Monte Carlo**: MC is essential for optimizing a given torsion in its local environment, in addition to removing any bias from the systematic Explore approach. It enables the search to focus on low-energy valleys in the potential energy surface by accepting only those conformations that satisfy the Metropolis criterion,

$$exp[-(\epsilon_0 - \epsilon_1)/kT] \leq R \tag{2.4}$$

where $\epsilon_1$ is the trial energy and $\epsilon_0$ is the previously accepted move. $\epsilon_0$ is initialized with the average torsion energy for the sector.

Since the goal of Focus is to provide comprehensive sampling, the MC sampling temperature is set to 1000K via **mcTemp**. This ensures that conformations are

randomly generated throughout the sector, while avoiding unfavorable regions. Lower temperatures might restrict the search to a narrow region and thus introduce bias.

The parameters governing the MC engine are summarized in Table 2.3.

| Parameters | |
|---|---|
| mcAcceptanceMode | accept conformations according to metropolis criterion, basic (only accept improvements in energy) or all |
| mcMaxSteps | total number of allowed Monte Carlo steps |
| mcTemp | temperature of Monte Carlo |

**Table 2.3**: Parameters pertaining to the Monte Carlo engine in FocusSampling

In summary, **numRotamers** governs the fineness of the torsion search and modulates the values of **numExploreRotamers** and **numExploreRotamers**, which are internal variables transparent to the user. Although the user does not directly modulate these values, their performance is illustrated in the context of the relevant sampling mode. The **numExploreRotamers** parameter defines the number of evenly spaced intervals at which a torsion is evaluated during the Explore stage of the search. If **focusSampling** is enabled, these torsions probe the local potential energy surface to find favorable regions. Thus, this parameter should be reasonably small such that sufficient resolution is obtained. Currently, these increments are spaced by $12^o$ intervals.

### 2.4.2 Sampling procedure

The sampling procedure outlined in flow charts Fig. 2.3 and Fig. 2.2 is discussed in this section. In general, regardless of the sampling path, the search proceeds from the lead cluster toward the downfield clusters. For each cluster $k$, the hinge between it and the $(k-1)^{th}$ cluster is rotated, resulting in a set of child conformations. These conformations then serve as parents for the $(k+1)^{th}$ cluster. This yields an ensemble of conformations referred to as the conformation tree, similar to Fig. 2.8, wherein each node represents an alternative torsion angle configuration at a given cluster and each branch represents a distinct conformation.

The search procedure utilizes a breadth-first approach, whereby the $k^{th}$ hinge is

sampled for all $N$ members of a conformation tree before proceeding to the $(k+1)^{th}$ cluster. This is in contrast to a depth-first search, for which all $K$ hinges of a given conformation $n$ are sampled before proceeding to the $(n+1)^{th}$ conformation. The breadth-first approach was implemented as it could simultaneously probe the receptor binding site while generating conformation ensembles suitable for statistical analysis. In contrast, a depth-first approach would spend most of the search time probing with conformations that would ultimately be incompatible with the receptor. Thus, convergent statistics are not likely to be obtained with this approach.

As described, this approach grows as $\prod_i^{Max} g_i N_i$, where $N_i$ is the number of rotamers found for step $i$ and $g_i$ is the degeneracy of the step (for when a path splits into two or more chains). This exponential growth in the conformation tree can quickly consume memory resources, even when using the internal coordinates representation described in Section 2.3.2. To address this, a trimming technique called conformationSort is discussed in Section 2.5.

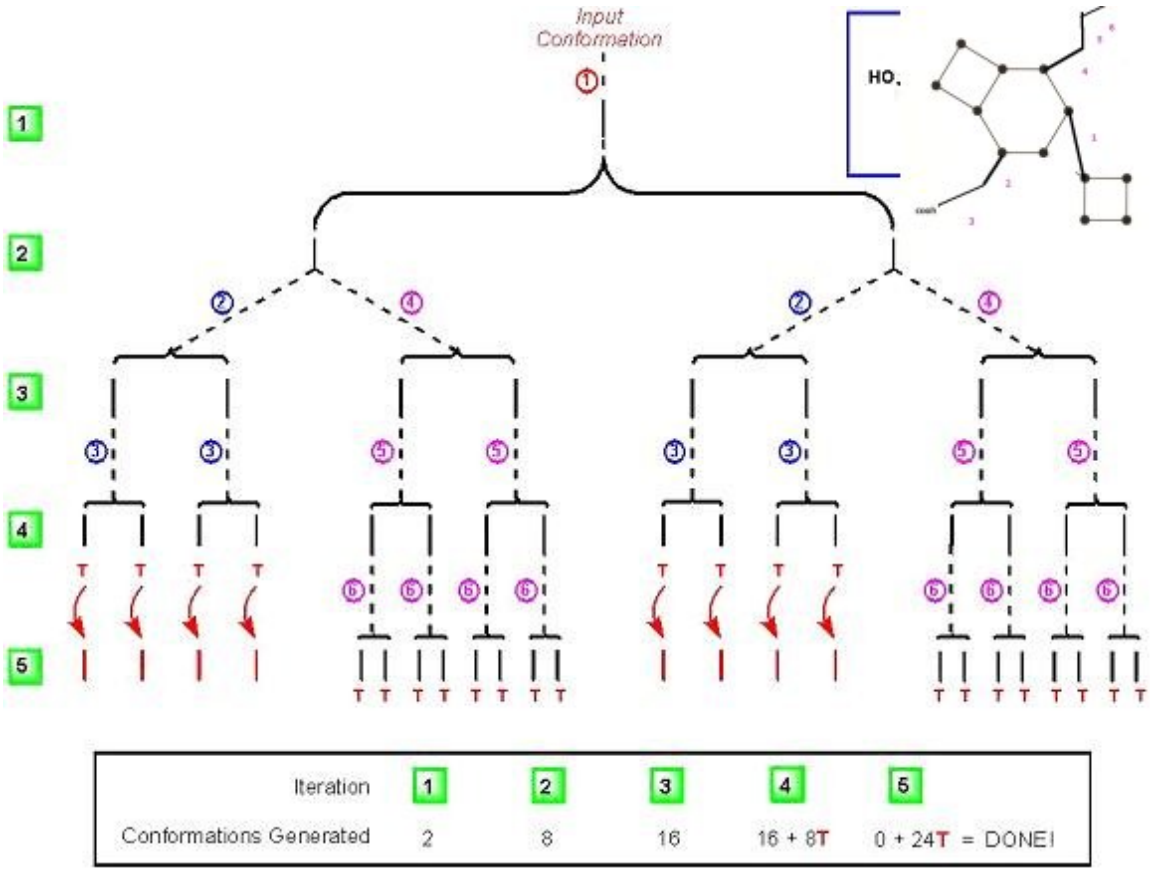



**Figure 2.8**: A depiction of a typical conformation tree. Starting from an input conformation, each bond is sampled outward from the parent cluster in a recursive fashion. In a given iteration, a parent bond generates a set of conformations, which then serve as parents in the following iteration

### 2.4.3 Sampling paths and branched chain strategy

#### 2.4.3.1 Defining the search trajectory

Before sampling, the order in which clusters are to be addressed must be established; this is referred to as the sampling path. While the sampling path is trivial to define for simple unbranched chains, there is considerable ambiguity when considering branched substituents. Therefore, a priority must be assigned to those branches that are most likely to be the strongest determinants of the overall ligand structure. Peptide chains are great examples of this principle, in that the main-chain (peptide bonds) largely determine the overall position, while the secondary branches (the amino acid side chains) optimize the ligand's interaction with the receptor. Given this, there is a priority in identifying solutions for the dominant chain before addressing the compatible positions of constituent branches.

Toward this end, Alg. 5 was implemented to identify and prioritize possible sampling paths. This method first partitions the molecule into unique branches of maximum possible length. The longest of these branches is designated as the primary branch and is always sampled first. The remaining branches (secondary branches) may be ordered according to length, or by proximity to the primary branch tip. This procedure is summarized in Fig. 2.9.

After the branch order is established, torsion sampling is performed on the primary branch, yielding a conformation tree, $P$, with $N$ members. These members serve as parent conformations for the $k_0^{th}$ cluster of the first secondary branch, $b_1$. From here, the sampling of the secondary branches can proceed in one of two directions: combinatorial or sequential.

#### 2.4.3.2 Sampling strategy

**Combinatorial sampling**: For combinatorial sampling, sampling continues for all clusters of $b_1$ and independently of the remaining $(m-1)$ branches. This results in a conformation tree, $B_1$, whose members are sorted according to energy. This sampling process is continued for all $M$ branches to form the ensemble $B = \{B_1, B_2, \ldots, B_M\}$.

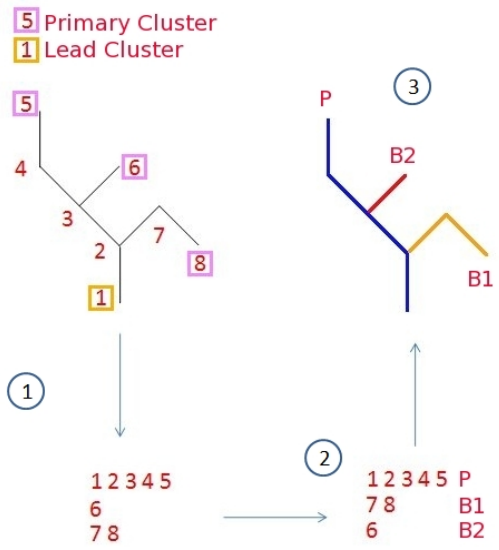



**Figure 2.9**: A depiction of the path-based sampling. **(1.)** Identify all possible sampling directions. **(2.)** Sort these according to length to identify primary and secondary branches. **(3.)** Conduct sampling according to the path order P (blue), B1 (yellow) and B2 (red)

Solutions for the entire molecule are constructed by drawing a single member from each of the $B_m$ sets. For example, the first solution, $S_0$, combines the lowest member from $P$ with the lowest conformations from each $B_i$. The second solution, $S_1$ replaces the first $B_i$ conformation with the second lowest energy. This process is repeated until the desired number of solutions are obtained or all conformations available in $B$ are exhausted. This combinatorial approach is represented by Eq. 2.5 and the first row of Fig. 2.10.

$$C = \{B_0|P\} \times \{B_1|P\} \times \cdots \times \{B_n|P\}. \tag{2.5}$$

An algorithm for constructing solutions according to this protocol is presented in Alg. 6.

**Sequential sampling**: Alternatively, the results from sampling of one branch can serve as input to the next branch as shown in Eq. 2.6. In this way, a much smaller conformation tree can be maintained as opposed to storing an ensemble for each $B_i$

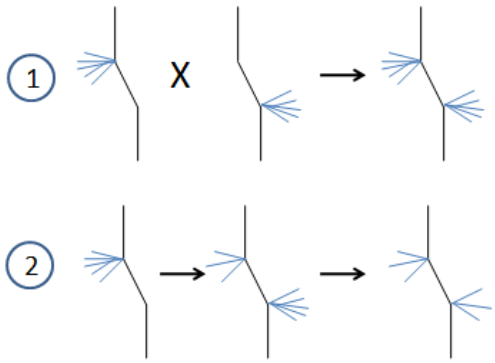



**Figure 2.10**: A depiction comparing combinatorially and sequential sampling. 1) Combinatorial sampling generates each branch independently and combines the ensembles for the final configuration. 2) Sequential sampling progressively samples each branch and prunes the total number of solutions to maintain a constant number of configurations

and $P$.

$$C = \{B_n \bigcup \ldots \{B_1 \bigcup \{P \bigcup B_0\}\}\} \tag{2.6}$$

This approach is represented by the second row of Fig. 2.10.

The primary shortfall of the sequential approach is that as new branches are sampled, the total number of conformations stored must remain constant. This necessarily requires pruning solutions from prior branches, which risks the loss of conformations that would ordinarily form the global minimum structure.

The parameters impacting the depth and direction of torsion sampling are listed in Table 2.4. **samplingPath** defines the order in which the ranked branches are searched although by default, *moleculeGL* samples the longest path first and then proceeds to sample paths furthest from the base. **recursionDepth** affects the number of torsions sampled given a sampling path. This option may useful for sampling branches for which the base and terminal regions are well defined, but the middle region is unknown.

| Parameters | |
|---|---|
| samplingPath | defines order in which branches are sampled |
| recursionDepth | number of clusters sampled along defined samplingPath |

**Table 2.4**: Parameters impacting the depth and direction of torsion sampling

## 2.5 ConformationSort

As described in Section 2.4, torsionSampling exhaustively probes the receptor binding site by generating an expansive conformation tree; conformationSort concentrates this growth by creating a smaller subset of the conformation tree. This sorting protocol illustrated in Fig. 2.11 consists of three components, DiversitySort, Filters, and ConformationScoring. Section 2.5.1 describes DiversitySort, which is the set of methodologies for reducing a conformation tree into a smaller number of diverse conformations. Filters are used to further reduce the conformation pool by enforcing constraints such as burial, and are described in Section 2.5.3. The third component, ConformationScoring, is used to rank conformations according to an energy in Section 2.5.4.4. Lastly, a collection of topics pertaining to the implementation and fine-tuning of conformationSort is provided in Section 2.5.4.

### 2.5.1 DiversitySort

The first component of conformationSort is DiversitySort, which consolidates the conformation tree into a spatially diverse subset. This involves a procedure for grouping data according to common properties called clustering. Since the objective is to explore the free space of the receptor binding region, the most germane criterion for clustering is the spatial distribution of the data. This is equivalent to grouping conformations according to their position in the binding cavity.

Conversely, a subset of the conformations can be selected that guarantees a minimum distance between solutions. This approach is the essence of diversity, which relates the distance between data groupings. Two- and three-dimensional examples of clustering are provided in Fig. 2.12(a) and Fig. 2.12(b).

*moleculeGL* supports two approaches for achieving diverse conformations. The

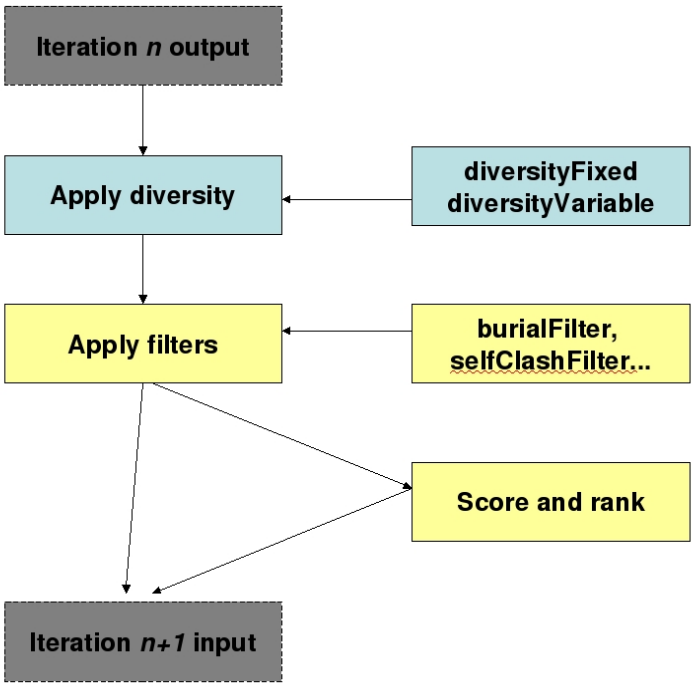
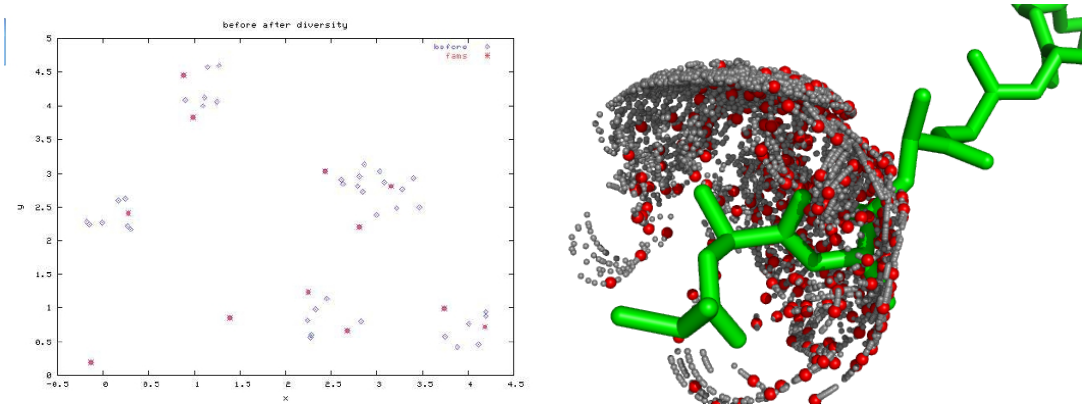



**Figure 2.11**: ConformationSort flowchart **(1.)** DiversitySort reduces the conformation pool to a structurally diverse set. **(2.)** **Sorting filters** use chemically motivated criteria, such as **burialFilter**, to further reduce the set. **(3.)** Scoring isolates the most physically viable conformations using a set of energy functions.

first, **diversityFixed**, finds all conformations whose distance is at least some constant from all other members. Alternatively, the **diversityVariable** approach finds a constant number of conformations that have a maximum distance from all other members. The modes are illustrated in Fig. 2.13(b) and summarized in Table 2.14.

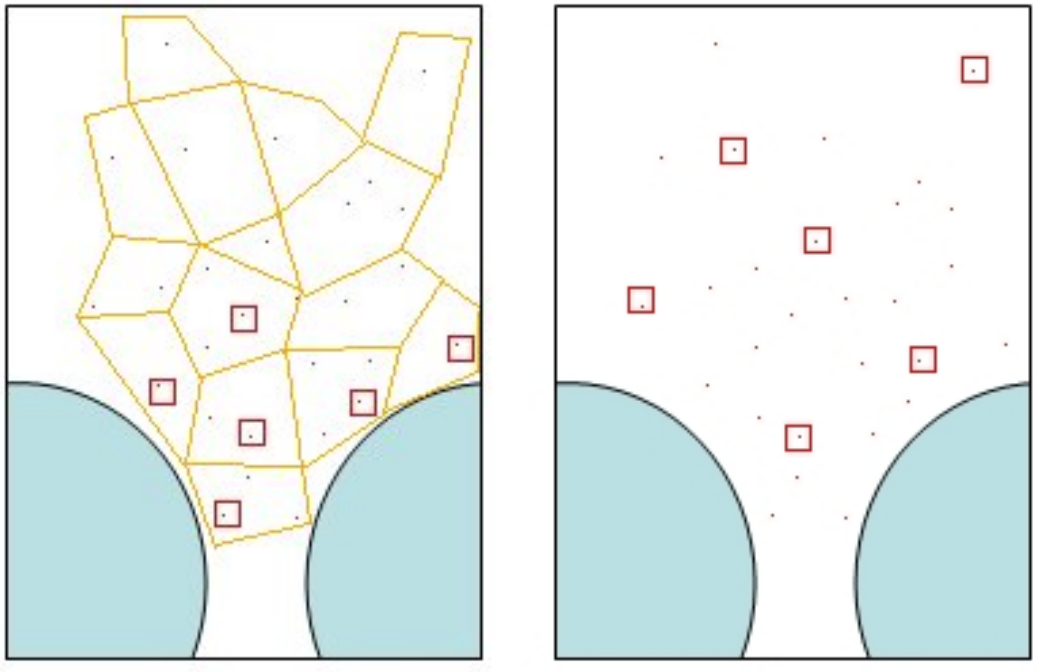
| Diversity approach parameters | |
|---|---|
| diversityMode | specify diversity method |
| *diversityFixed* | retain conformations above diversity value |
| *diversityVariable* | retain most diverse set of conformations |
| diversityFixedValue | set minimum diversity for retaining conformations |

**Table 2.5**: Parameters for determining the diversity strategy




(a) 2D diversity (b) 3D diversity

**Figure 2.12**: Diversity clustering demonstrated for sampling data in **(a.)** two and **(b.)** three dimensions. In both figures, the original data (blue) is condensed into spatially distinct clusters (red)



(a) diversityFixed (b) diversityVariable

**Figure 2.13**: **diversityFixed** and **diversityVariable** diversity sorting modes. **(a.)** **diversityFixed** partitions data into clusters satisfying a fixed diversity criterion. **(b.)** **diversityVariable** adjusts the diversity to retain a constant number of conformations

#### 2.5.1.1 diversityVariable

The goal in diversityVariable is to reduce a large data set to a constant number of points according to some measure of diversity. To achieve this number, the diversity is increased, which groups data points below the cutoff into common clusters. The diversity value required to obtain a given number of conformations reflects the free space available to the ligand and varies between iterations. For instance, if the search expands into an open space, selecting the $N$ most diverse conformations will lead to an ensemble with larger diversity than selecting $N$ in a narrow region.

k-means and hierarchical are two strategies that have been well supported in the literature for performing clustering; k-means attempts to consolidate clusters to minimize the objective function

$$F = \sum_{j=1}^{K} \sum_{n \in S_j} |x_n - \mu_j|^2, \tag{2.7}$$

where $x_n$ is the position of component $n$ in cluster $j$ and $\mu_j$ is the mean position of cluster $j$ [19]. This method is strongly dependent on the initialization conditions and relies on a cluster geometric mean, $\mu$, which may be unphysical given that the fixed bond and angle requirements heavily constrain the system.

Alternatively, hierarchical sorting iteratively groups data points by pairs until a desired number of clusters are obtained. In other words, the process clusters according to

$$min\{d_{xy} : x \in A, y \in B\}, \tag{2.8}$$

where $A$ and $B$ are sample clusters in the conformation tree. The advantage of this protocol is that there is little dependence on the initialization conditions. That is, provided that the $n(n-1)$ pairwise distances are unique, the clusters solutions will also be unique. This strategy is portrayed in Fig. 2.14(a).

In general, hierarchical clustering groups data into clusters, but this by itself does not condense the data set. DiversitySort is an adaptation of this procedure to retain only one solution when merging two nearby conformations, as is shown in Fig. 2.14.

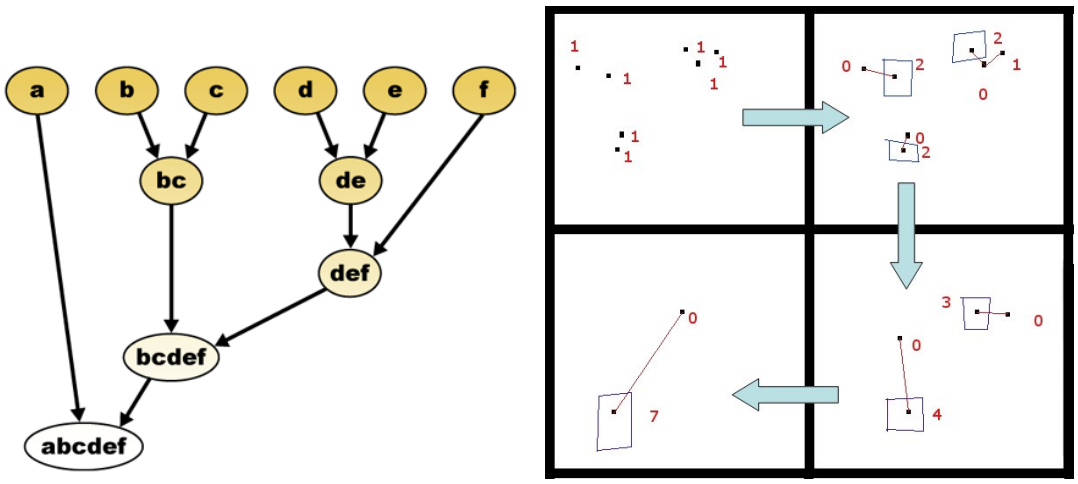
The selection process compares scores (defined below) between two solutions and discards the loser. If scores are equivalent for both members, then a solution is chosen randomly.

**usePrimaryWeights** represents the first scoring metric underlying this selection procedure. This approach is a reflection of the number of child conformations a given solution represents. In Fig. 2.14, for example, the remaining solution has survived three rounds of consolidations and collectively represents seven conformations. By consistently choosing the cluster with higher weight, the remaining conformations tend to represent the center of a nearly Gaussian distribution of points, as is demonstrated in Fig. 2.12(a). This procedure is outlined in Alg. 7.

Other weighting options include **useCumulativeWeights** and **useSecondaryWeights**. **useCumulativeWeights** accumulates the weights from all previous DiversitySort iterations. Maintaining this history biases the results toward conformations that tend to have a tightly clustered solutions at each iteration. Similarly, **useSecondaryWeights** currently selects conformations according to their burial extent. In tandem, these weighting schemes select for well-buried conformations that tend to represent the median of a cluster of solutions.

One primary difficulty with this approach is the arbitrariness of the stopping criterion. Typically the criterion is based on the maximum number of conformations that can be stored to memory, although rules may be devised that exploit the characteristics of the distribution. Another problem is that the diversity is applied uniformly to all points in the receptor free space. In practice, the distribution of conformations is sparse far from the binding site and dense near the binding region. Applying a uniform diversity parameter thus may sacrifice density in regions crucial to binding. This motivates a technique introduced later in Section 2.5.3.1, which eliminates outlying conformations prior to clustering.

Table 2.14 below lists the parameters supported by the diversityVariable module. The parameters **usePrimaryWeights** and **useSecondaryWeights** toggle the use of weight arrays when consolidating conformations, while **useCumulativeWeights** stores the weighting arrays from previous iterations.




(a) (b)

**Figure 2.14**: **(a.)** Hierarchical clustering iteratively groups data until a desired number of clusters is reached. **(b.)** The *moleculeGL* implementation consolidates data points into a defined number of clusters .

| Parameters | |
|---|---|
| usePrimaryWeights | tallies number of consolidations a solution has undergone |
| useCumulativeWeights | accumulates PrimaryWeights from prior iterations |
| useSecondaryWeights | burial of given solution |

**Table 2.6**: Parameters for the hierarchical diversity approach

#### 2.5.1.2 diversityFixed

diversityFixed partitions the conformation tree into a variable number of clusters whose centers are separated by a fixed diversity value. Because this method only groups conformations, it must be combined with another metric to reduce the subset. An energy score is thus used to discriminate between viable structures. This approach yields a unique set of low-energy families spanning the accessible search space with a consistent density. To this end, the raw data set is separated according to groups defined as families and children, as outlined in the following steps and presented in Fig. 2.15:

$$A \subset C : \forall i, j \in \{A\}, d_{ij} \geq cutoff. \tag{2.9}$$

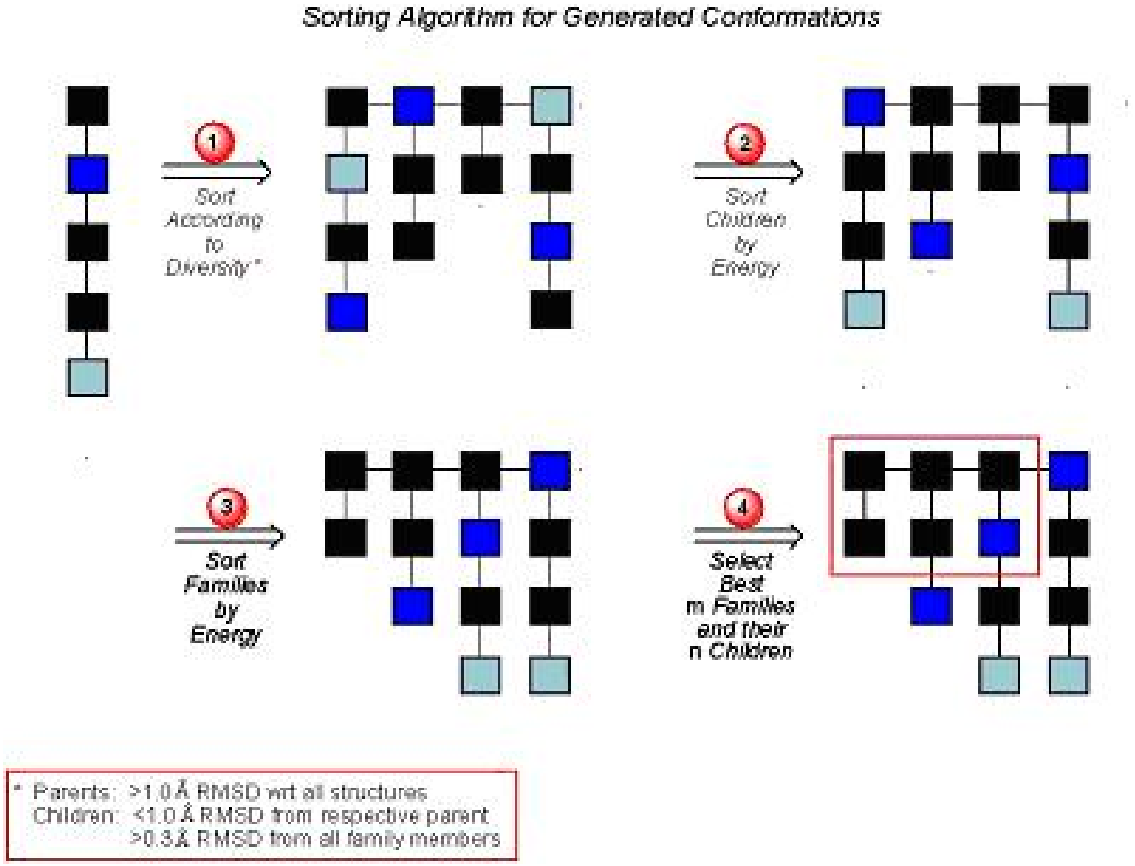



**Figure 2.15**: The **diversityFixed** protocol consists of **(1.)** sorting generated conformations into diverse families with related children, **(2.)** ranking children in each family according to energy, **(3.)** sorting families according to the energy of the top-ranked member, **(4.)** retaining the top $m$ families and their best $n$ children

**Step 1.** The sequential list of rotamers is partitioned into families, for which each family represents a class of solutions whose RMSD with respect to all other members of the data set is greater than **familyDiversity**. Solutions which resemble a given conformation (RMSD $<$ **familyDiversity**) are grouped into the same family and denoted as children. Within each family, children that share less than **childDiversity** with respect to all other family members are discarded as degenerate solutions.

**Step 2.** With the conformations separated according to diverse families, the members of each family are ranked according to energy. The lowest energy candidate is designated as the *family head*.

**Step 3.** Repeat **Step 2** for ordering families by the top-ranked child.

**Step 4.** At the user's request, the program returns the top m members from each of the top n families as the solutions to the rotamer generation problem.

This protocol is disadvantageous for several reasons. One, the results are directly dependent on the order in which the families are defined, thus a unique solution is not guaranteed for a given data set. Second, when there are more clusters than can be accepted, energy is used as a selection criterion, which is often unreliable for partially constructed conformations. Lastly, the constant diversity constraint may introduce a bias when the conformation tree grows prohibitively large, since there is an upper bound on the number of structures that can be retained.

#### 2.5.1.3 Burial-weighted diversity

One drawback of enforcing uniform density of conformations across the sampled space is that solutions far from the protein binding site inflate the average diversity of the ensemble. Thus, the conformations obtained after diversity sorting tend to be biased toward vacuous regions of the sampling space. It is desirable, however, to have a higher density of solutions near areas of the ligand-binding domain (LBD) where burial can be optimized. To enforce this, the diversity score is weighted by the relative burial of the clusters. To this end three schemes have been proposed:

$$linear = \gamma \times \alpha \tag{2.10}$$

and

$$sublinear = \gamma \times \sqrt{\alpha} \tag{2.11}$$

and

$$superlinear = \gamma \times \alpha^2 \tag{2.12}$$

where $\gamma$ is the diversity and $\alpha$ is the burial extent, which is computed for each conformation. When consolidating two conformations, the burial extent for the pairing is computed as the geometric mean of the individual burial values. In this way, the mean burial extent will be biased toward the conformation with greater burial extent and thus improve its likelihood of being retained. The burial extent described above can be decribed in terms of the **burialAvgNum**, which average number of atoms near the atoms comprising a cluster, or **burialTotalNum**, which refers to the total number atoms near a cluster.

| Parameters | |
|---|---|
| DiversityVariableuseBurialWeight | burial-weighted diversity scores |
| *burialAvgNum* | mean number of protein atoms near ligand atom |
| *burialTotalNum* | total number of protein atoms near cluster |

**Table 2.7**: Parameters pertaining to burial-weighted diversity

### 2.5.2 RMSD approximations

The diversity discussion in the preceding sections relied on a distance-based metric for discriminating between conformations. A commonly used metric is RMSD, which for two molecules is defined as

$$RMSD \equiv \sqrt{\frac{1}{N}\sum_{i=1}^{N} |x_i - x_i'|^2}, \tag{2.13}$$

where $x_i$ and $x_i'$ are the positions of the $i^{th}$ atoms and $N$ is the total number of atoms. Regardless of the diversity protocol used, this relationship must be computed for all possible conformation pairings, which is inherently a $\mathcal{O}(n^2)$ operation, where

$n$ refers to the number of atoms. Thus by identifying subsets of atoms the provide a reasonable estimate of the overall ligand position, a linear increase in performance is obtained. In this regard, the function SetRMSDComparison allows the user to select from several reduced representations, which are depicted in Fig. 2.16.

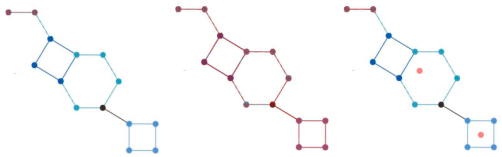


**Figure 2.16**: The reduced representations available to **rmsdComparison** include **(a.)** **vectorOnly**, **(b.)** **vectorAllprior**, and **(c.)** **vectorCOM**. The marked atoms (red) serve as the basis for the computed RMSD values

One such approximation, **vectorOnly**, uses the atoms comprising the rotatable bond between the most recently sampled cluster and its parent. It is also possible to augment the vector description with information about the parent clusters by using the *prior* tag. **vectorCOMprior** approximates the positions of previous clusters by computing the center of mass (COM) for each cluster. This measure provides a reasonable estimate of the cluster position but loses information about the orientation, which could be especially important for clusters that are planar in nature. Alternatively, the **vectorAllprior** adds the positions of all parent cluster atoms. Finally, the **all** mode uses all atoms (including those of the current cluster) and is thus an exact RMSD. This mode is used by default after the entire ligand is reconstructed. These parameters are summarized in Table 2.8.

| Parameters | |
|---|---|
| rmsdComparison | how to define RMSD for conformation comparison |
| *all* | use all atoms |
| *vectorOnly* | use atoms from last rotatable bond |
| *vectorAllprior* | use vector description and all atoms for prior confs |
| *vectorCOMprior* | use vector description and prior confs center of mass |

**Table 2.8**: Parameters governing the manner by which RMSD is computed and the clusters involved

### 2.5.3 Sorting filters

Filters utilize criteria based on chemical intuition to eliminate conformations that are unlikely to bind to the protein. As such, they complement the diversitySort protocol, which thins the conformation tree by a geometric criterion alone. The **burialFilter** penalizes conformations that are not sufficiently buried in the protein. The **selfClashFilter** penalizes conformations that have internal nonbond clashes between atoms. These filters are exemplified in Fig. 2.23. Additionally, **strainFilter** discards conformations whose internal strain exceeds a defined value, while **hbFilter** penalizes conformations with unsatisfied hydrogen bond partners.

These filters are generally performed after diversity on a smaller subset of conformations, since the procedures are computationally expensive. One exception is **burialFilter**, as this filter has a substantial impact on the diversity and is worth the expense. These filters are applied on or after the fifth iteration of the search in order to accumulate a sufficiently large pool of conformations.

| Parameters | |
|---|---|
| burialFilter | enable/disable use of burialfilter |
| selfClashFilter | enable/disable filter based on eliminating self-clashing conformations. |
| strainFilter | enable/disable filter for eliminating highly strained conformations |
| hbFilter | filter based on excluding confs with unpaired hbonds |

**Table 2.9**: Various filters available to conformationSort

#### 2.5.3.1 burialFilter

**burialFilter** discards conformations that are insufficiently buried within the receptor, which is a requisite for a strong binding interaction. Burial is measured as the percentage of ligand atoms that are within **burialDist** of any protein atom, or in other words,

$$Pass = \{c \,|c : |Buried|/|All| \leq \textbf{burialPercent}\}, \tag{2.14}$$

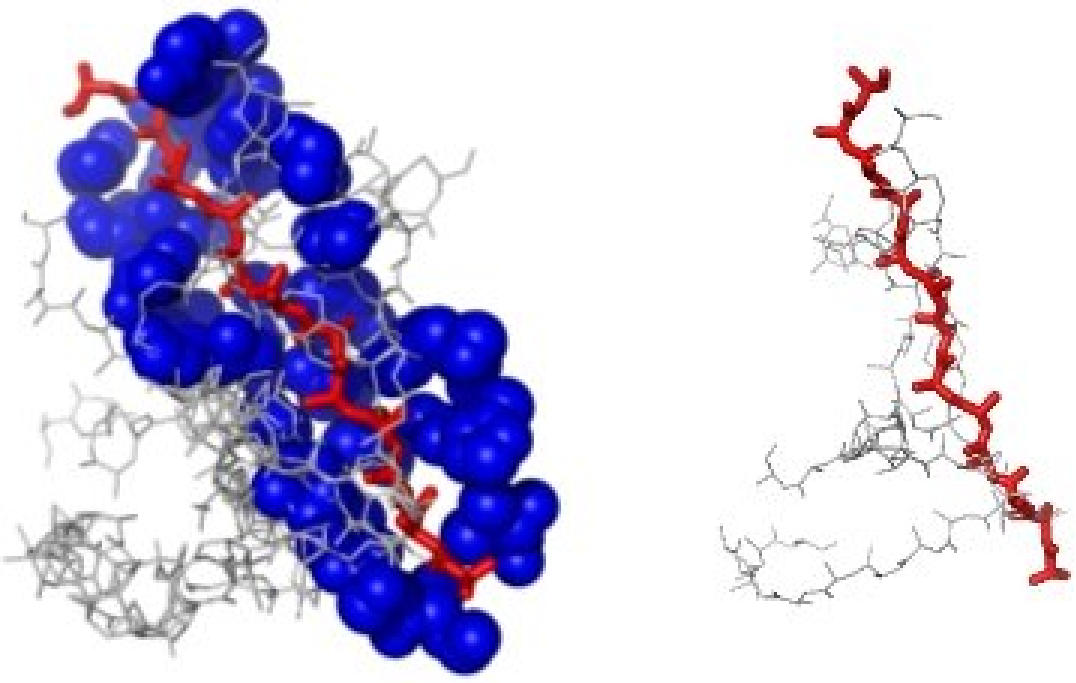


**Figure 2.17**: Examples of the **(a.)** **burialFilter** and **(b.)** **selfClashFilter** filtering techniques. The conformations in gray were eliminated according to the respective filtering criteria

where $|Buried|$ and $|All|$ are the sets of buried atoms and all ligand atoms, respectively. A buried atom is formally defined as

$$i \in \{Buried\}\ if\ \exists j \in \{Protein\} : d_{ij} < \textbf{burialDist}, \tag{2.15}$$

where $d_{ij}$ is the distance between atoms $i$ and $j$, while {Buried} and {Protein} are the sets of buried atoms and receptor atoms, respectively. A **burialDist** value of 4.0 was selected for this parameter, as this is a generous approximation of the average VDW distance between non-hydrogen atoms.

One important consideration is which subset of ligand atoms to consider for burial. Two possibilities are to use 1) all atoms of the ligand or 2) just the subset of the current sampled cluster. The latter is an inexpensive approximation for tightly bound

ligands and is the default. However, some compounds, especially polypeptides, have overall high burial percentages but often contain several loosely buried regions that would ordinarily be penalized. The **burialCumulative** option was introduced to retain the burial percentage from prior clusters when assessing total ligand burial. In this way, loosely buried regions can be accommodated provided the overall burial exceeds **burialPercent**.

The motivation for this is that, in order to avoid clashes with the protein, the zero-potential energy surface outside the protein does not penalize conformations that deviate from the protein surface. Since the rotamer sampling scoring function does not include solvation, interaction energies beyond the VDW equilibrium distance tend to zero. On one hand this behavior is advantageous, since a ligand may have weakly interacting regions that bridge two well-stabilized anchors. On the other hand, some conformations may deviate too far from the protein to be considered strong binders. Moreover, 'unburied' conformations can bias the diversity engine away from the protein. As the conformation tree migrates away from the protein, the diversity between ensemble members increases, while those within the binding groove will likely have a higher density. Since the diversity engine attempts to maximize diversity between conformations, an overly large diversity cutoff would potentially thin these good solutions or eliminate them completely.

| Parameters | |
|---|---|
| burialFilter | enable/disable use of burialfilter |
| burialPercent | minimum percentage of buried atoms needed to declare entire conformation buried. |
| burialDistance | minimum distance for an atom to be considered buried. |
| burialCumulative | use entire, partially built conformation for burial computation |

**Table 2.10**: Parameters for **burialFilter**

#### 2.5.3.2 selfClashFilter

Intramolecular interactions are neglected in rotamer sampling due to their computational expense, which increases the likelihood of self-clashing conformations Fig. 2.23.

To address this, the **selfClashFilter** discards conformations for which self-intersection is likely. To reduce the computational expense of this assessment (an $\mathcal{O}(n^2)$ operation), the filter compares the distances between the anchor cluster and the most recently sampled cluster. Thus, a passing conformation is one that satisfies Eq. 2.16:

$$Pass = \{c \,|c : \forall i, j \in \{Clusters\}, d_{ij} \geq \textbf{clashDist}\}. \tag{2.16}$$

Including self-interaction terms during the rotamer sampling introduces an significant computation expense, since an entire conformation must be regenerated from internal coordinates as opposed to just a single cluster. Moreover, usually only regions that are separated by several bonds have significant self-interaction terms, thus a brute-force $\mathcal{O}(n^2)$ ligand-ligand nonbond interaction evaluation would be inefficient.

Neglecting the self-interaction term greatly improves the sampling time, but there is a propensity for longer chains to curl up, or equivalently, clash with posterior regions. To counteract this effect, the self-interaction assessment is done at the conformationSort stage, where the expensive self-interaction computation can be done on a smaller subset of conformations.

In practice, **clashDist** is set to a value slightly less than the average VDW distance. However, a special consideration must be made for hydrogen bond pairs, which have equilibrium distances below the van der Waals distance. It is also expected that this filter would be especially critical for branched chains as a test for compatibility between secondary branch solutions.

| Parameters | |
|---|---|
| selfClashFilter | enable/disable filter based on eliminating self-clashing conformations. |
| clashDist | conformations with distances smaller than this value are rejected |

**Table 2.11**: Parameters for **selfClashFilter**

#### 2.5.3.3 strainFilter

As with intramolecular scores, internal strain is ignored during torsion sampling. The algorithm assumes that all hinges have no barrier to rotation, yet in reality, internal strain can profoundly impact the types of rotamers available to a ligand. Intuitively, if more energy is spent contorting a ligand to the binding site than is compensated through intermolecular interactions, then there is no energetic advantage to binding. In this regard, the strainFilter was implemented, which discards conformations whose internal strain is in excess of the conformations binding energy. A passing conformation is one which satisfies Eq. 2.17:

$$Pass = \{c \,|c : score(c) \leq \textbf{internalEnergyCutoff}\}. \tag{2.17}$$

| Parameters | |
|---|---|
| strainFilter | enable/disable filter for eliminating highly strained |
| strainCutoff | conformations with strain greater than this are rejected |

**Table 2.12**: Parameters for **strainFilter**

#### 2.5.3.4 hbFilter

To ensure that all hydrogen-bonding atoms are satisfied in a given conformation, the **hbBurial** filter has been proposed. For a given conformation, if any of its polar atoms are not within a cutoff of a complementary hydrogen bonding partner, then the atom is considered unsatisfied. Conformations with unpaired hydrogen bond atoms are marked for deletion.

$$Pass = \{c \,|c : |hbBuried|/|Atoms| \leq \textbf{hbPercent}\}, \tag{2.18}$$

where $|Atoms|$ is the set of considered atoms. A hydrogen-bonded atom is formally defined as

$$i \in \{hbBuried\} \; if \; \exists j \in \{Polar\} : d_{ij} < \textbf{hbDistMax} \tag{2.19}$$

where $d_{ij}$ is the distance between atoms $i$ and $j$.

| Parameters | |
|---|---|
| hbFilter | filter based on excluding confs with unpaired hbonds |
| hbPercent | conformations with a smaller burial pctg are cutoff |

**Table 2.13**: Parameters for **hbFilter**

### 2.5.4 Miscellanea

**Sampling density**: The goal of conformationSort is to reduce an input conformation tree into a more compact and potent set. It is therefore crucial to identify a minimum value for the number of conformations needed to ensure adequate sampling. The parameter **intermediateFamiles** establishes an upper bound on the size of the reduced set and it is expected that the optimal value for this parameter will be dependent on the density of the solutions (which itself is a function of **numRotamers**).

#### 2.5.4.1 SortingModes

For maintaining a diverse conformation set, ConformationSort offers two conceptually different techniques, **sortingTraditional** and **sortingDiversityOnly**. **sortingTraditional** proceeds by performing both diversity sorting and energy ranking. This approach is best suited for tightly interacting binding sites, for which each local minima in individual cluster positions are likely to be close to the global minimum. **sortingDiversityOnly** relies solely on diversity and neglects scoring. This approach emphasizes saturating the empty space of the receptor before considering energies, which is crucial for sampling loose binding sites. Moreover, it avoids biasing the search toward favorably scoring partial conformations that may have little bearing on the overall conformation. By default, **sortingDiversityOnly** is used for the intermediate iterations, while at the final iteration, **sortingTraditional** is called. These parameters are listed in Table 2.14 below.

#### 2.5.4.2 Diversity by family

While the approaches summarized above yield diverse conformations, oftentimes the first few iterations consolidate the most recently generated rotamers into a single conformation. Not only does this undo the sampling of the previous iteration, it forces the search of the subsequent iteration into an arbitrary direction. To address this, the **sortByParents** mode was introduced, which calls the diversity engine with a smaller number of requested conformations. After retaining a list of acceptable solutions, the original list of conformations is culled of all members that share the same parent number as the diversity-selected solutions.

#### 2.5.4.3 Triggering the diversity engine

**numIntermediateFamilies** describes the number of conformations that are retained when diversity is called. Generally, diversity is triggered when the total number of conformations exceeds a threshold that is a magnitude higher than this parameter. By doing this, the program avoids calling diversity at every iteration, which allows the retained solutions to grow somewhat before being pruned once again. These additional iterations without pruning can often eliminate a significant portion of the conformation pool via clashes with the receptor and thus concentrates the remainder of solutions in viable LBD regions. Lastly, at the final iteration, **finalFamilies** defines the number of top-ranked conformations to return. The details of the scoring functions are described in Section 2.6.

#### 2.5.4.4 Conformation scoring

Depending on the sortingMode selected (see Section 2.5.4.1), conformations may be scored and ranked as a final step in ConformationSort. At this stage, partially constructed conformations are scored with the coarseScore energy functions, while at the final iteration, the fineEnergy scoring function is used. For both cases, an energy cutoff for accepting conformations can be defined using **confScoreCutoff**.

| Parameters | |
|---|---|
| sortingMode | |
| *sortingTraditional* | sort according to diversity cutoff |
| *sortingDiversityOnly* | sort for spatially diverse set |
| sortByParents | sort based on parent clusters, not children |
| numIntermediateFamilies | families retained during growth |
| numIntermediateChildren | for **sortingTraditional** |
| numFinalFamilies | families retained after final sorting |
| numFinalChildren | for **sortingTraditional** |
| confScoreCutoff | cutoff score for final sorting |

**Table 2.14**: Parameters for determining conformationSort strategy

#### 2.5.4.5 Filtering schemes

A variety of filtering schemes have been proposed that aim to reduce the number conformations for consideration. It is not expected that a single filter will be robust across all molecule types or environment; rather, the tandem usage of several filtering modes offers the strongest potential for drastically reducing the conformation pool. Typically, a set of filters would be executed in a hierarchical fashion, such that a generally applicable, inexpensive filter would be applied before more detailed and computationally involved filters.

To this end, the hierarchicalFiltering.pl script was developed to simulate the application of customizable filtering schemes to output data files. In this fashion, the performance of different strategies can be evaluated without having to rerun the original job. The hierarchical scheme chosen for this study executes the following steps in sequential order:

1. discard bottom 50% ranked by coarse energy
2. discard bottom 50% ranked by burial number
3. discard bottom 50% ranked by fine energy.

## 2.6 Scoring

*moleculeGL* assesses both intermolecular and intramolecular nonbond interactions, as well as valence terms between directly connected atoms. These nonbond interactions include VDW, hydrogen bonds, and electrostatics, which are discussed in Section 2.6.3, Section 2.6.4, and Section 2.6.5, respectively. However, to motivate the functional forms of the energy expressions, an aside on fine-grain versus coarse-grain scoring is included in Section 2.6.1. Lastly, Table 2.15 below summarizes the scoring functions and equations.

| Parameters | |
|---|---|
| vdwMode | vdw96,vdw126,vdwPiecewise |
| hbMode | hbPiecewise, hbLinear, hbNone |
| coulMode | coulInvR,coulFdWeighted,coulNone |
| fineScore | vdw96+hbdreiding+invR +*intramolecular* |
| coarseScore | vdwPiecewise+hbPiecewise+codecoulNone |
| nonbondCutoff | cutoff for assessing nonbond interactions |
| vdwRadiiScale | vdw radii used for heavy atoms |
| hvdwRadiiScale | vdw radii used for hydrogen atoms |

**Table 2.15**: Parameters related to scoring

### 2.6.1 Coarse- versus fine-grain approaches

*moleculeGL* utilizes both coarse-grain and fine-grain scoring functions, depending on whether rapid evaluation or high accuracy is desired. The fine-grain scoring mode, **fineScore**, applies conventional polynomial energy expressions based on quantum mechanics that accurately capture the interaction between various atoms types. For valence and hydrogen bond terms, these potentials consist of multibody expressions for which a minimum of three atoms must be evaluated. These expressions often involve significant computational cost and thus are not practical for brute-force MC simulation, and so are reserved for discriminating between fully constructed ligands. As implemented in *moleculeGL*, **fineScore** uses the 9-6 Lennard-Jones (LJ) potential for VDW, Dreiding-style hydrogen bonds, and the Coulomb potential for electrostatic interactions.

Coarse-grain potentials approximate nonbond interactions with expressions that are less expensive to compute than their fine-grain equivalents. Typically, these potentials are piecewise step functions that assign constant function values for several pairwise distance ranges and return a zero potential beyond a defined cutoff. As implemented in *moleculeGL*, **coarseScore** uses piecewise potentials for the VDW and hydrogen bond terms, while electrostatics are neglected. The **coarseScore** mode is primarily used during the torsionSampling stages, where the focus is on filling the receptor binding site with reasonable pose estimates.

### 2.6.2 Intramolecular scoring and valence interactions

Although the vast majority of energy calls are intermolecular, the FineScoring may be configured to additionally compute intramolecular interactions. Whereas pairwise interactions are computed for all atoms when scoring two molecules, the $(i+1)\ldots(i+3)$ nearest-neighbor interactions must be excluded for intramolecular scoring. This is because these interactions are addressed via valence terms, such as bonding, angle, and torsion terms. It is necessary, therefore, to separate nearest-neighbor pairs from those normally addressed with the nonbond scoring. Since the connectivity of atoms does not change during the course of simulation, this list is generated at initialization using the efficient algorithm described in Section 2.3.4.

The internal energies are assessed using the valence code from LAMMPS [20], which includes standard harmonic descriptions for bonds, torsions, and angles. Some methods, like CHARMM [21], do in fact compute nonbond interactions between the $i^{th}$ and $(i+3)^{th}$ atoms, although it is commonplace to capture this with an appropriate torsion barrier.

### 2.6.3 van der Waals functions

VDW refers to the dispersion interaction between atoms, which is repulsive at close distances and oftentimes slightly attractive within certain ranges, depending on the atom types [22]. The function forms used in this program are discussed in Sec-

tion 2.6.3.2 and Section 2.6.3.3. Before addressing these, however, an aside on VDW radii is provided in Section 2.6.3.1, as these scalings form the basis of the VDW functions used.

### 2.6.3.1 Atomic radii

VDW radii determine the optimal distance between atoms and vary as a function of atom type. *moleculeGL* uses reduced VDW radii to increase the size of conformational space sampled and thus tolerates non-optimal contacts between atoms. Furthermore, the radii for hydrogens are completely eliminated, since they are generally ignored during torsion sampling.

### 2.6.3.2 Lennard-Jones

The Lennard-Jones function is the status quo for describing VDW interactions and is expressed in Eq. 2.20

$$E_{LJ}(r_{ij}, r_0) = \frac{D_0}{R - A}\left\{A\left(\frac{r_0}{r_{ij}}\right)^R - R\left(\frac{r_0}{r_{ij}}\right)^A\right\}, \tag{2.20}$$

where $r_{ij}$ is the distance between atoms $i$ and $j$, $r_0$ and $d_0$ are the optimum distance and well-depth, while the $A$ and $R$ exponents define the attractive and repulsive curvature, respectively. *moleculeGL* supports the 12-6 and the default 9-6 forms, where the first and second numerals refer to $R$ and $A$ exponents. The general function above is modified to accommodate scaled VDW radii in Eq. 2.21. . This piecewise function smoothly shifts the potential minimum inward while retaining the favorable energy at $r_0$. In this way, there is more tolerance in accepting close conformations without penalizing those situated at the unscaled equilibrium distance. A plot of these functions is provided in Fig. 2.18(a).

$$E = \begin{Bmatrix} E_{LJ}(r_{ij}, r_{\sigma^*}) & r_{ij} \leq r_{\sigma^*} \\ E_{LJ}(r_\sigma, r_\sigma) & r_{\sigma^*} < r_{ij} \leq r_\sigma \\ E_{LJ}(r_{ij}, r_\sigma) & r_{ij} > r_\sigma \end{Bmatrix} \tag{2.21}$$

#### 2.6.3.3 Piecewise

For **coarseGrain** scoring, a piecewise potential is available that is similar to the hard-sphere approximation, with several important additional features. Firstly, a zero-slope potential is assumed at the optimal pairwise distance, $d_\sigma$, as this favors near-optimal distances without biasing the conformation to a particular value. Secondly, the shelf near $d_{\sigma_1}$ allows the incorporation of some close contacts without a stiff penalty and is based on **vdwRadiiScale**. The resulting parameterization yields a potential depicted in Fig. 2.18(b).

$$E_{piecewise}(r_{ij}) = \left\{ \begin{array}{cc} E_{clash} & r_{ij} \leq d_{clash} \\ E_{near} & d_{clash} < r_{ij} < d_{\sigma_1} \\ E_{opt} & d_{\sigma_1} \geq r_{ij} \leq d_{\sigma_2} \\ E_{far} & r_{ij} > d_{\sigma_2} \end{array} \right\} \tag{2.22}$$

where $d_{ij}$ is the pairwise distance between atoms $i$ and $j$. The distances used in the aforementioned formula are defined below:

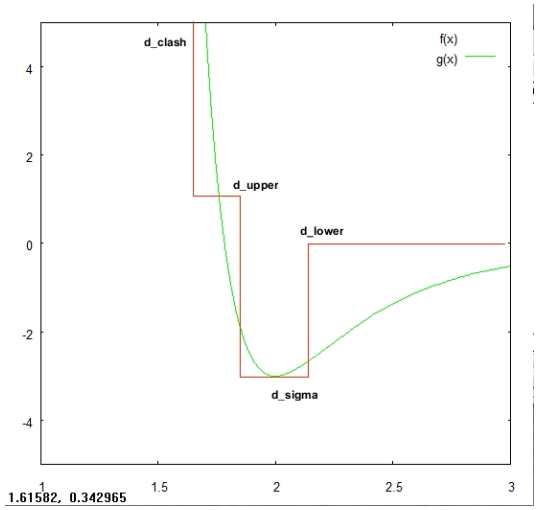
$$\begin{aligned} d_{clash} &= d_{\sigma^*} - d_{tol} \\ d_{\sigma^*} &= \gamma \times d_\sigma \\ d_{\sigma_1} &= d_\sigma - d_{tol} \\ d_{\sigma_2} &= d_\sigma + d_{tol} \end{aligned} \tag{2.23}$$

for which $d_\sigma$ is the geometric mean of the VDW radii for each atom, $\gamma$ is the scaling factor, $d_{\sigma_1}$ and $d_{\sigma_2}$ are the lower and upper bounds for the potential well, and $d_{tol}$ is the tolerance factor. The $d_{tol} = 0.85$ was chosen based on analyzing a large number of Lennard-Jones 9-6 plots and determining where y = 1.0 occurred on average.

### 2.6.4 Hydrogen bond functions

#### 2.6.4.1 Linear and Dreiding

Hydrogen bonding refers to a hydrogen-mediated attraction between two polar atoms, which are designated as the hydrogen donor and hydrogen acceptor. Hydrogen bonds




(a) (b)

**Figure 2.18**: Plots of van der Waals potentials for the **(a.)** Lennard-Jones (red) and **(b.)** piecewise expressions (blue)

have both an electrostatic and weak bonding component. The former component is described as a function of distance between the two polar atoms, while the latter component adds a dependence on the hydrogen position. Hydrogen bonding can be reasonably approximated with a two-body potential resembling the LJ expression:

$$E_{linear}(r_{ij}) = D_{HB}\left[5\left(\frac{r_0}{r_{ij}}\right)^{12} - 6\left(\frac{r_0}{r_{ij}}\right)^{10}\right]. \tag{2.24}$$

The hydrogen contribution can be addressed by assessing DHA angle formed by the donor, hydrogen, and acceptor atoms. An optimal interaction is obtained when these atoms are collinear and decays to zero as the $DHA \rightarrow$90 degrees. This description underlies the Dreiding hydrogen bond potential, which is expressed in Eq. 2.29:

$$E_{dreiding}(r_{ij}, \theta) = [cos(\theta_{DHA})]^4 \times E_{linear}. \tag{2.25}$$

As a side note, some simulation packages do not include a hydrogen bond term, as it is believed that hydrogen bonding can be adequately described by the Coulomb potential, given appropriate charge assignment. While this may be adequate for describing the bulk properties of a homogeneous fluid, the approach may be insufficient

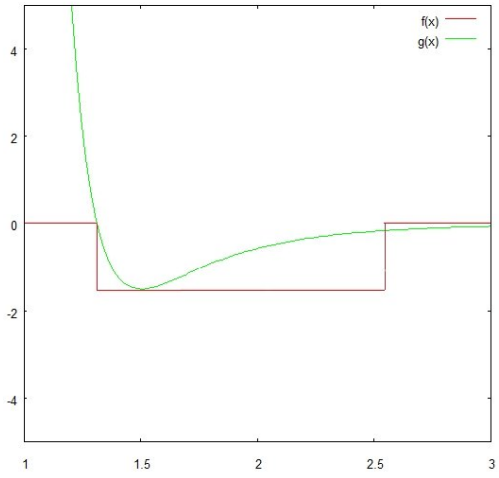
for describing the interface between two components.

#### 2.6.4.2 Piecewise

The piecewise formulation of the hydrogen bond assigns a favorable, constant energy to polar atoms satisfying a reasonable hydrogen bond distance. This is given by Eq. 2.26,

$$E_{HBpiecewise}(r_{ij}) = \left\{ \begin{array}{ll} 0 & r_{ij} \geq 4.0 \\ -C & 2.0 \geq r_{ij} < 4.0 \\ E_{VDW} & r_{ij} < 2.0 \end{array} \right\} \tag{2.26}$$

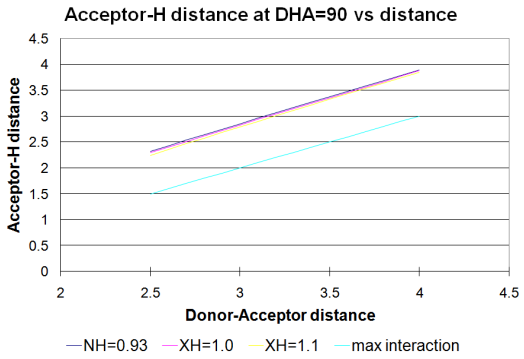
where $r_{ij}$ is the distance between atoms $i$ and $j$ and $C$ is a constant. This function considers all polar atoms, including sulfur, oxygen, and nitrogen, with a 2.0 to 4.0 Å interatomic distance. Pairs below this range are scored with the piecewise VDW function. This method does not verify the donor/acceptor eligibility, which requires verifying the existence and position of an intermediary hydrogen. This grants considerable tolerance to donor/acceptor pairs satisfying a hydrogen bond distance. Fig. 2.19(a) plots the function described by Eq. 2.26.




(a)

**Figure 2.19**: Plots of **(a.)** linear (blue) and piecewise (red) hydrogen bond functions

**Hydrogen angle**: Ordinarily, hydrogen positions are ignored in this algorithm, since




(a)

**Figure 2.20**: Plots of the possible HA distances for a given DA distance. The maximal iteration is found when the donor, acceptor and hydrogen are collinear. The upper curve denotes the hydrogen position when the DHA is perpendicular

their positions are variable and may be determined in a refinement step. However, in some cases, notably $sp^2$ hybridized nitrogens, the hydrogen position is rigidly defined by the donating atom. As the hydrogen position relative to the donor and acceptor atoms ultimately determined the strength of hydrogen bonding, it is imperative to include this constraint in the scoring function.

The DHA angle argument of the Dreiding forcefield indicates that the hydrogen bond is valid for $\theta \in [0, \pi/2]$. At $\theta = 0$, the maximum hydrogen bond interaction is maximized and this diminishes to zero as $\theta \rightarrow \pi/2$. Therefore, for each donor-acceptor (DA) distance, $r_{DA}$, optimal and worst-case hydrogen-acceptor (HA) distances are determined. A piecewise function is derived based on these values that assumes a linear relationship between the optimal and worse case hydrogen positions. DHA angles beyond 90 degrees are set to zero. This gives the corrected piecewise potential

$$E_{HB}(r_{ij}, r_{HA}) = \omega_{corr} E_{HB}(r_{ij}) \tag{2.27}$$

with

$$w_{HB} = max\left(m_{HA}\dot{r}_{HA} + (r_{DA} + Corr), 0\right) \tag{2.28}$$

where $m_{HA}$ and $Corr$ are parameters determined from a linear-least squares (*See* Fig. 2.20(a)) fit to the best and worst case hydrogen positions for a variety of DA

and HA distances. This term gives more favorable scores to hydrogen bond pairs for which the hydrogen is optimally placed. At the present, the implementation of this correction is limited to $sp^2$ hybridized nitrogens; however, this is the most common of the donors for which the hydrogen position is constrained.

**Recovery of the repulsive wall**: As a slight variation to the standard angle-dependant hydrogen bond potential, a weighted VDW term was added to yield

$$E_{HB+VDW}(r_{ij},\theta) = [cos(\theta_{DHA})]^4 \times E_{linear} + [1 - cos(\theta_{DHA})]^4 \times E_{vdw}. \tag{2.29}$$

. The inclusion of this term reinstates the VDW repulsion at $\theta_{DHA} = 90^oC$, which is normally neglected in the hydrogen bond expression. Most standard molecular mechanics packages tend to disable Coulombic and van der Waals potentials when the hydrogen bond function is used, as the expression typically encompasses both terms over typical hydrogen bonding ranges. As shown in Fig. 2.21, as $\theta$ approaches the maximum accepted DHA angle of 1.57 radian, the repulsive term approaches zero and hence fails to penalize the clashing atoms. The modified expression instead gradually adds van der Waals character to the energy expression along the periphery and thus restores the full repulsive force when atoms are too closely spaced. This is an important consideration for Monte Carlo sampling, which can place hydrogen-bonding atoms in positions that would ordinarily give rise to strong VDW clashes.

### 2.6.5 Electrostatics

Electrostatics describe the attractive and repulsive interaction between oppositely and similarly charged particles, respectively. The Coulomb potential of Eq. 2.6.5 provides an exact description of the electrostatic interaction,

$$E_{coul} = \frac{q_i q_j}{\epsilon\, r_{ij}}, \tag{2.30}$$

where $q_i$ refers to the charge of atom $i$, $r_{ij}$ is the pairwise distance, and $\epsilon$ represents the dielectric constant and vacuum permittivity. This representation is generally

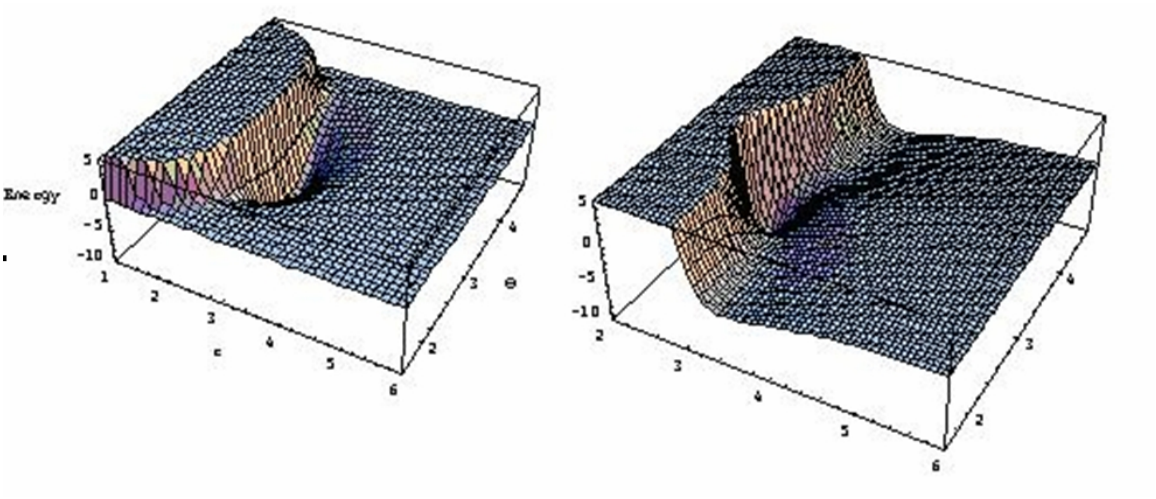


Figure 2.21: Plots of the potential energy surfaces corresponding to the Dreiding hydrogen bond expression (left) and the proposed hydrogen bond/vdw potential (right)

valid only for fully constructed molecules, for which integral charges are guaranteed. Otherwise, long-range artifacts decaying as $1/r$ will result. Since coarseScore primarily scores partially constructed conformations for which charges are non-integer, a piecewise representation was not pursued.

Even when complete conformations are scored, there is a propensity for long-range charge interactions to dominate the electrostatics. It is expected that a solvated charge would exert an appreciable electrostatic field within about three Å. In solvated systems, an ion is normally stabilized by the solvent through the formation of water shell in which the surrounding water molecules align their dipoles to counterbalance the ion's charge. The resulting water cluster does have an effective charge on its surface, which would be counterbalanced by the adjacent water shell, but this behavior generally dies out within two to three water shells, as suggested by radii of gyration. As such, the electrostatic potential induced by the charge should be relevant only within a few angstroms. To incorporate this behavior into the Coulomb potential, a weighting factor motivated by Fermi-Dirac statistics is introduced in Eq. 2.31

$$\omega_{coul}(r_{ij} = \frac{1}{e^{\alpha(r-r_{cutoff})} + 1} \tag{2.31}$$

where $\alpha$ defines the rate at which the Fermi-Dirac potential decays and $r_{cutoff}$ is the cutoff. As modified above, the weighting function evaluates to 1.0 as $r \rightarrow 0.0$, but rapidly decays to zero for $r > r_{cutoff}$ (red curve in Fig. 2.22). By weighting the Coulomb potential by this function, interactions beyong the cutoff are smoothly attenuated.

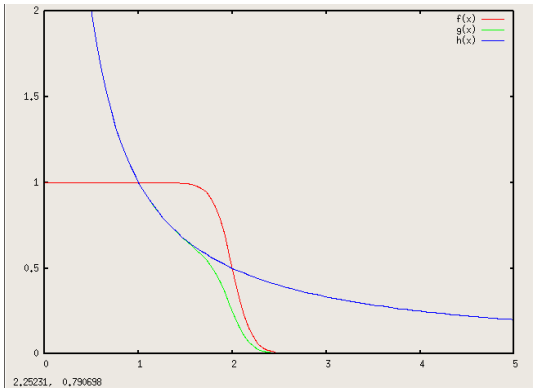



**Figure 2.22**: Plot of the weighting expression for which the Fermi-Dirac function (red) attenuates the Coulomb potential (blue) at increasing distances

### 2.6.6 Nonbond cutoff

The current *moleculeGL* implementation explicitly computes pairwise interactions between the ligand and the protein. In order to reduce the number of computations, a nonbond radius may be defined, beyond which all protein atoms are ignored. At initialization, all protein residues with at least one atom within the cutoff are retained for scoring, while all others are discarded. Thus, care must be exercised in defining a sufficiently large radius to ensure a nonzero protein contribution in all possible sampling regions. For elongated ligands, this usually requires defining a very large radius that offers negligible computational advantage over using all protein atoms.

Other methods for handling long-range nonbond interactions exist, including the cell multipole method [23] and Ewald summation [24].

## 2.7 Overall Sampling Method

At this stage, scoring, sampling, and sorting have all been formally introduced, thus the discussion turns to how these are combined for a functional protocol (Fig. 2.3). **traditionalSampling** and **wagSampling** describe two conceptually different protocols for approaching torsion sampling.

**traditionalSampling** emphasizes using energies during torsion sampling and sorting, which is both computationally expensive and may bias sampling, depending on the size of the LBD. As such, it is best suited for small, tightly bound ligands. This mode uses fineScore scoring, focusSampling, and sortingTraditional.

On the other hand, **wagSampling** prioritizes unbiased sampling of the conformation space with minimal computational expense; as such, the mode uses **coarseScore** scoring with focusSampling disabled, while **diversityOnly** sorting mode is used. The **wagSampling** mode is preferred for large, loosely bound ligands.

| Parameters | |
|---|---|
| samplingMode | option for performing sampling |
| *traditionalSampling* | focusSampling on |
| | exact energy functions |
| | sortingTraditional (offline) |
| *wagSampling* | focusSampling off |
| | vdwPiecwise |
| | hbPiecewiseCorr |
| | sortingVariable |
| | selfInteraction off |

**Table 2.16**: Parameters governing the overall sampling approach

## 2.8 Other Considerations

Though the primary use of *moleculeGL* is its torsion sampling algorithm, additional features have been included that are of utility for general applications in drug design. This section discusses some of the additional features and options *moleculeGL* supports.

| Parameters | |
|---|---|
| design | option for clash design |
| alanize | option for mutating residues to alanine |
| cavityanalysis | option for performing cavity analysis |

**Table 2.17**: Additional functions for protein design, binding site alanization, and analysis of interactions within the binding cavity

### 2.8.1 Alanization of the receptor binding site

*moleculeGL* supports sampling modes that mutate local binding site residues to alanines to increase the sampling space. Because the protein structures obtained from X-ray crystallography are typically optimized for the co-crystal with which they are bound, the LBD may preclude the binding of a similar compound if the side chains are unable to reposition themselves. Moreover, even when binding the cognate compound, less well-resolved co-crystals may have residues whose positions are ill-defined.

By converting the local residues within the binding sites to alanines, the protein binding site can be reduced to an empty shell bound by the protein backbone. This effectively removes the bias introduced by the native residues and any imprecision in their positioning. This strategy, which is known as alanization, describes the replacement of protein side chains with residues truncated at the alpha carbon. In this configuration, ligand sampling can be expected to yield a more spatially diverse set of conformations than would be possible in the native protein.

After an ensemble of conformations is generated, the LBD is dealanized, which describes the replacement of alanines with the native amino acid types. Many of these residues are expected to clash with the generated conformations and thus their

positions must be regenerated to ensure compatibility with the ligand. This is a recurrent theme in protein design that is referred to as side-chain replacement, which has been shown [25] for optimizing protein and substrate binding.

Typically side-chain replacement protocols, such as Side-Chain Replacement Method (SCREAM) [26, 27], combinatorially sample a set of rotamers for each amino acid and attempt to minimize the overall energy. The rotamers are not typically generated on-the-fly but are drawn from amino acid rotamer libraries derived from the Protein Data Bank [15]. Similarly, the conformation tree from *moleculeGL* can be formatted into a rotamer library and thus subjected to side-chain regeneration in the presence of other LBD amino acids.

Although all residues within a binding site can be alanized indiscriminately, some residues, like the apolar and bulky, can be integral for defining the binding cavity. Thus, it is often advantageous to preserve the theses classes of residues and only allow the alanization of polar entities. *moleculeGL* currently supports alanization of the following types: all residues (*all*), polar residues (*polar*), hydrophobic (*hydrophobic*) and the set W,R,F,I,L,V (*wag*). These parameters are listed in Table 2.18.

| Parameters | |
|---|---|
| **alanizeMode** | (all,polar,hydrophobic,wag) |
| **SaveAllConfsScream** | save conformations in SCREAM format |

**Table 2.18**: Parameters pertaining to alanization

### 2.8.2 Protein design

Traditional drug design seeks to uncover a strongly interacting ligand for a given drug target. In contrast, the goal instead is to optimize a protein to specifically bind a given ligand. This process is referred to as protein design and is a corollary of the alanization principle discussed above. As opposed to simply resampling clashing amino acids from an alanized binding site, the amino acids may be mutated to improve binding. For example, a tyrosine at position 420 (Y420) may clash with the ligand, regardless of the rotamers used. Therefore, Tyr420 could be mutated to each of the

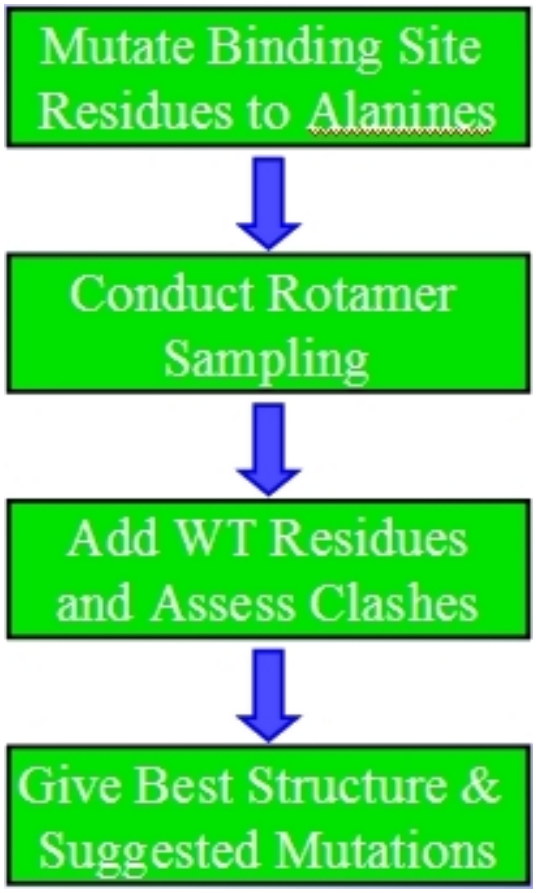



**Figure 2.23**: Flow chart depicting the protein design protocol. Step **(1.)** alanizes the binding site, Step **(2.)** performs rotamer sampling, Step **(3.)** introduces the wild-type residues, and Step **(4.)** identifies mutation possibilities

19 remaining amino acids and tested for compatibility. It is very likely that the bulky tyrosine would be replaced with a smaller apolar residue such as isoleucine, valine, or leucine, resulting in three possible mutants (Y420I, Y420V, Y420L).

### 2.8.3 Cavity analysis

The evaluation of nearby nonbond interactions between a ligand and a protein binding site, otherwise known as cavity analysis, is a crucial step in assessing the integrity of binding. Such procedures typically enumerate nonbond interactions within a specified radius, but usually neglect internal strain and solvation effects. Higher accuracy

function forms are typically assumed for these atomistic interactions, such as the Morse potential for van der Waals and a multibody hydrogen bond term. These functions are coded into *moleculeGL* and are routinely employed in the FineScoring stage of the rotamer search.

### 2.8.4 Software architecture

In its original form, the *moleculeGL* code was written in C and assumed a sequential architecture. This architecture merely executes functions in a sequential order, which often involves the gratuitous use of global variables and exhibits high interdependence. From a programmer's perspective, this primitive structure discourages code reuse and can be difficult to understand conceptually. The ultimate result is that subsequent code development must resolve interdependence, or cohesion, between code bits and this presents difficulties implementing outside code.

In contrast, an Object-oriented (OO) interface follows a more intuitive assembly-line structure, in which code bits are separated into distinct and minimally dependent domains of related functions. The advantage of this is that modules can be easily modified or substituted with minimal changes to the overall code structure. Moreover, this structure is easily adaptable to more advanced architectures that maintain a desired control flow. *moleculeGL* reflects what is called an observer-command architecture, in which communication proceeds from higher level modules to primitive objects, with little cohesion between objects. The advantage of this approach is that the primitives can be easily replaced if necessary, so long as they adhere to the communication protocol defined by the high levels.

An additional advantage of this architecture is that the low-level code is manipulated only through the use of function calls, accessors, and mutators. Accessors allow a user to access the value of a state variable, while mutators allow the state to be changed. Adhering to this architecture facilitates its interface with other programs, as relevant functions and variables can be manipulated while protecting components unrelated to the interface. This also simplifies its compulation as a library, such that

the code's functionality can be safely exported to molecular simulation programs. To this end, an interface was developed for accessing *moleculeGL*'s public library functions through Perl (Fig. 2.24).

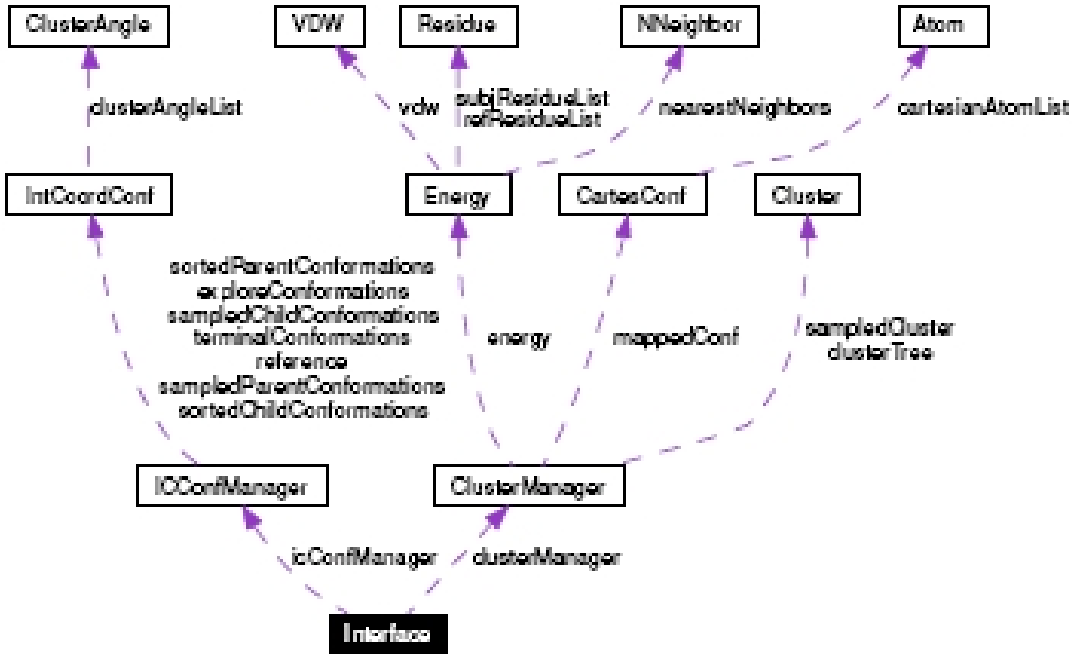



**Figure 2.24**: Schematic of the interface for accessing underlying *moleculeGL* library functions

A recent interfacing suite, SWIG [28], facilitates access to libraries of arbitrary origin with a large variety of scripting languages like Perl or Python, though the code is written in C++. In this way, a researcher could conceivably call *moleculeGL* to perform torsion sampling, then directly pass the results to a Fortran object for molecule dynamics, all while using the native high-level functions of the scripting language. To accomplish this, one need only to write an interface that accesses the intended functions within the library; thereafter SWIG writes the wrapper code which enables the interaction with a given scripting language. More importantly, the swig interface is sufficiently general that no revisions are necessary for wrapping the

library into an alternative language. Lastly, by relying on a scripting language for higher level functions like string manipulation, operations which are unrelated to the actual function of the code can be eliminated. (*See* Fig. 2.25).

The code base for this project is written in Perl, and orchestrates the writing and reading of data files and performing system calls to various molecule mechanics codes and *moleculeGL*. Whereas *moleculeGL*1.0 was called from the command line within Perl and required external file operations, *moleculeGL*2.0 is directly integrated with the code. Moreover, by adhering to a generic data structure for the molecule files, other swigged codes like SCREAM can operate directly on the molecules without having to implement File I/O routines.

```
#!/usr/bin/perl

# initialize
use MoleculeUtil::MGL;
my $c = MoleculeUtil::MGL::new();

# set
$c->SetParameterFile("./neutral-0.11.par");
$c->SetNonbondCutoff($dist);

# run
my $energy = $c->CavityAnalysis($lig,$prot);

# interpret
my $numRes = $c->AccessNumResidues($protName);
for my $i (0..$numRes)
{
        my $vdw = $c->AccessResidueEnergyVDW($i);
}
```

**Figure 2.25**: An example Perl script using the *moleculeGL*-SWIG interface

# Chapter 3

# Methods

## 3.1 Methods

### 3.1.1 Co-crystal prediction

The *moleculeGL* protocol has been applied to a co-crystal dataset based on a publication from Eldridge et al. [29] that includes several co-crystallized structures of trypsin and for hosts like neuraminidase, ribonuclease $T_1$, and carbonic anhydrase. The featured ligands have as many as thirty-five sequentially-linked rotatable bonds, which is a challenging test of the balance between accuracy and maintaining a data set of manageable size. For the purpose of discussion, several co-crystals with a large number of rotatable bonds were chosen and these are listed in. Table 3.1 and displayed in Fig. 3.1.

The common protocol for evaluating the performance of docking algorithms is the comparison of its accuracy in reproducing the X-ray structure of a ligand in a fixed protein, for which a successful match reports an RMSD less than 2.0 Å. This is an acceptable value per [13, 3], but the basis of this choice is developed further in Section 4.2. Generally, docking algorithms combine the prediction of the correct orientation of an entire ligand or fraction thereof with the flexibility sampling. Since *moleculeGL* is a flexibility algorithm and is not coupled with an orientation search, a starting position for a fragment of the ligand must be provided, from which all substituents are grown in.

| PDB code | $N_{rot}$ | description |
|---|---|---|
| 1apt | 15 | pepstatin analog (aspartyl proteinase penicillopepsin) |
| 1apu | 13 | pepstatin analog (aspartyl proteinase penicillopepsin) |
| 1cnx | 12 | sulfonamide (carbonic anhydrase) |
| 1icm | 13 | fatty acid analog (fatty acid-binding protein) |
| 1icn | 17 | fatty acid analog (fatty acid-binding protein) |
| 1seb | 32 | superantigen (human class II histocompatibility molecule) |
| 2ifb | 15 | palmitate (fatty acid-binding protein) |
| 5tln | 8 | hydroxamic acid inhibitor (thermolysin) |
| 6cpa | 14 | phosphonate (carboxypepsidase) |
| 6tmn | 14 | thermolysin inhibitor (thermolysin) |

**Table 3.1**: A subset of ligands with a large number of rotatable bonds

### 3.1.2 Co-crystal preparation

All complexes were obtained from the PDB and were prepared uniformly. Explicit waters, metals, and cofactors were removed from the complexes, including the ligand binding site, while their respective ligands were extracted for the study. Explicit hydrogens were added to the proteins and assigned CHARMm22 [21] charges, with the apolar hydrogen charges summed onto the heavy atoms, according to the parameters set forth in the DREIDING FF [16]. The ligands were assigned charges using the charge equilibration method [30]. Each result was scrutinized to verify that appropriate bond orders and hybridizations were identified. $Na^+$ and $Cl^-$ counter-ions were added to neutralize the side-chain charges in the absence of salt bridges, while crystal waters and other bound molecules were removed for docking to maximize the available surface of the protein. The potential energy of the entire structure was minimized using conjugate gradients to an RMS force of 0.1 kcal/mol via MPSIM [31]. The minimized protein-ligand complex was used as the reference to evaluate the accuracy of the predicted conformations.

The parent cluster is defined as the largest rigid body of the ligand, or when ambiguous, the most buried cluster is chosen by inspection. In many of the cases, the defined anchor is a heterocyclic ring of aromatic nature, though the sampled substituents vary from largely hydrophobic chains to polar tails with carbonyl groups.

Additionally, regardless of the actual periodicity of angular torsion energy shared by two atom types, all single bonds are treated as fully rotatable. Torsion sampling yields a set of energy-ranked final conformations whose size is defined by the optimized parameter, **numFinalFamilies**. The results reported in this document give the lowest RMSD conformations and their relative ranks with respect to the lowest energy conformation.

### 3.1.3 Parameter optimization

The impact of parameters on the efficacy of *moleculeGL* is investigated for a number of compounds. The trials assume the default parameter values listed in Table 3.2 for all but the parameter in question. The performance gains or losses are determined from the percentage of validation cases that meet a 2.0 Å RMSD cutoff. Where applicable, failure analysis and parameter estimations are obtained using standard spreadsheets such as Microsoft Excel.

| parameter | value |
|---|---|
| SamplingMode | wag |
| FocusSampling | 0 |
| NumRotamers | 8 |
| SetSortingMode | diversityOnly |
| IntermedFamilies | 1000 |
| FinalFamilies | 200 |
| SetRMSDComparison | vectorAllprior |
| DiversityMode | hierarchical |
| BurialFilter | 1 |
| BurialPercent | 0.8 |
| SelfClashFilter | 1 |
| VDWMode | vdwPiecewise |
| VDWRadiiScale | 0.9 |
| HVDWRadiiScale | 0.5 |
| HBMode | piecewisecorr |
| CoulMode | off |
| NonbondCutoff | 90 |

**Table 3.2**: Default parameters for *moleculeGL* validation

### 3.1.4 Comparison of calculated trypsin inhibitor binding to experimental inhibition constants

Predicting binding affinities for a set of trypsin inhibitor co-crystals are compared with reported inhibition constants, $K_i$ [32]. A $K_i$ is approximately proportional to $e^{(\frac{-Ei}{RT})}$, thus the predicted binding affinities are expected to scale linearly with the log of the inhibition constants. The conformations of for each trypsin inhibitor is predicting using the default scheme described above. The top 50 conformations based on *moleculeGL* energy scores are subjected to 500 steps of conjugate gradient minimization via MPSIM to reconcile discrepancies between the reduced and full van der Waals radii. The RMSD change with respect to the *moleculeGL* predicted structure was generally less than 0.5 Å for all cases. The best-ranked conformations by RMSD for each compound was used for computing the binding energies.

The relative binding energies for the best ligand conformations are defined as the difference between the ligand in protein versus in solution given by

$$\Delta\Delta G(calcd) = \Delta G(protein + ligand) - \{\Delta G(protein) + \Delta G(ligand)\} \qquad (3.1)$$

where $\Delta G(protein{+}ligand)$ is the free energy for the protein-ligand complex, $\Delta G(protein)$ is the free energy for the protein, and $\Delta G(ligand)$ is the free energy for the ligand alone. Since the free energy can be very difficult to estimate and often requires extensive conformation sampling to provide reasonable estimates, the strength of interaction can by approximated by the vertical binding energy, which is computed as

$$E(vertical) = E(complex) - \{E^*(protein) + E^*(ligand)\}$$

where the protein- and ligand-only energies correspond to the configurations extracted from the complex without minimization. This energy is referred to as the Single Point Energy (SPE) and neglects contributions to the binding energy due to structural relaxation. As such, the SPE represents a maximum bound to the binding interaction. All SPE calculations were computed with MPSIM [31] according to the Dreiding [16]

FF and AVGB [33] continuum solvation.

### 3.1.5 F-M-R-F-$NH_2$ bound to mouse MrgC11

The F-M-R-F-$NH_2$ neuropeptide in Fig. 3.2 and the G-Protein coupled receptor. (GPCR) MrgC11, were prepared according to the procedure outlined in [34]. Since the exact position of residues comprising the LBD were unknown, all non-polar residues within 4.0 Å of the ligand were alanized. Sampling was performed with the default parameters listed in Table 3.2, except for those listed in Table **??**. The dreiding-0.3.par FF [35] parameterization was used for the scoring functions native to *moleculeGL*. Lastly, peptide bonds were held fixed, as these are expected to be static at physiological temperatures.

The set of conformations from *moleculeGL* were printed to a rotamer library, which was used as an input to SCREAM [27]. These rotamers were sampled along with the native residues rotamers to yield optimal configurations within the LBD. The final complexes were minimized with MPSim and the binding energies were computed.

| parameter | value |
|---|---|
| FinalFamilies | 1500 |
| VDWRadiiScale | 0.3 |
| HBMode | piecewisecorr |

**Table 3.3**: Exceptions to the default parameters in Table 3.2 used for the FMRF trial

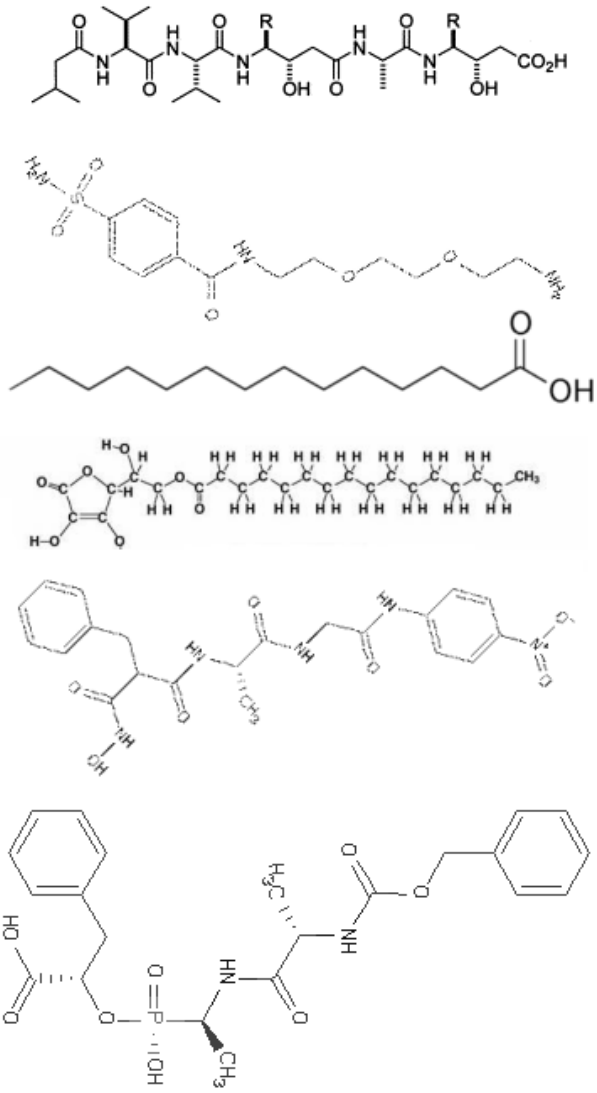


**Figure 3.1**: From top: 1apt, 1cnx, 1icm, 2ifb, 5tln, 6cpa

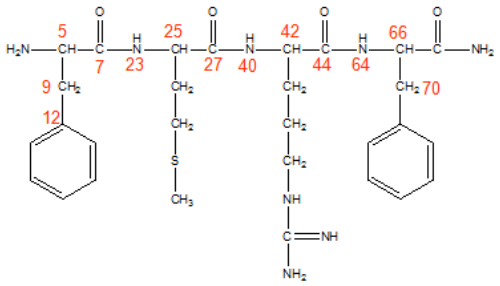



**Figure 3.2**: F-(D)M-R-F-$NH_2$ molecular structure

# Chapter 4

# Results and Discussion

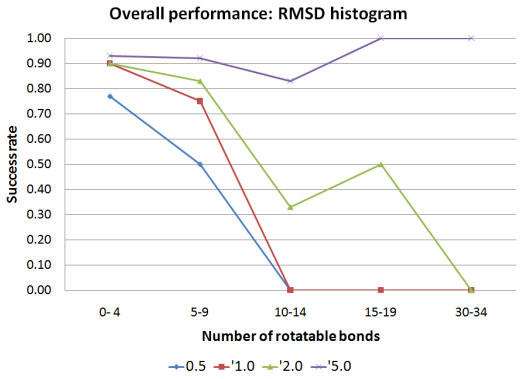
In this chapter the performance of the *moleculeGL* protocol is summarized and discussed. The most convincing testament of its utility is its performance on co-crystal test data, which is discussed in Section 4.1. The choice of metrics used in evaluating a ligand conformation is discussed in Section 4.2, while additional error analysis is presented in Section 4.3. In Section 4.4, the parameters underlying the *moleculeGL* protocol are analyzed and lastly, in Section 4.5 the successful prediction of the F-M-R-F-$NH_2$ neuropeptide in MrgC11 is summarized.

## 4.1 Validation by Co-crystal Prediction

### 4.1.1 General cases

As summarized in Table 4.1, *moleculeGL* predicted 90 percent of the co-crystal set within 2.0 Å RMSD using default parameters. In Fig. 4.1, the performance is plotted as a function of ligand size, which indicates that the success rate deteriorates as the number of bonds increased. However, ligands with well-defined hydrogen bond or electrostatic interactions, such as PTS (4S-trans)-4-(amino)-5,6- dihydro-6-methyl-4H-thieno(2,3-B)-thiopyran-2-sulfonamide-7,7-dioxide in 1cim and the peptide inhibitor PKI(5-24) in 1fmo, were exceptions to this trend. For these cases, strong interactions with the protein cavity limit the number of possible conformations and thus substantially simplify the search.

While it is evident that cases with a small number of rotatable bonds are easily predicted, the algorithm performance does not necessarily decay as the torsional degrees of freedom increase. In fact, the performance is dependent on factors such as the number of polar contacts or the extent of burial within the LBD. Thus, cases such as fatty acid inhibitors are typically easier to simulate given that they are entirely buried within the protein. On the other hand, polypeptides are typically more difficult given that they often bind along the protein surface and are thus more loosely bound.




(a)

**Figure 4.1**: Number of cases meeting various RMSD values depicted as a function of the number of rotatable bonds

| PBD code | RMSD | Time | $N_{rot}$ | PBD code | RMSD | Time | $N_{rot}$ |
|---|---|---|---|---|---|---|---|
| 1add | 0.82 | 11 | 2 | **1apt** | 5.90 | 5676 | 12 |
| 1apu | 1.66 | 3831 | 10 | 1bra | 0.22 | 21 | 2 |
| 1bzm | 0.16 | 400 | 4 | 1cbx | 0.30 | 490 | 5 |
| 1cil | 0.16 | 44 | 2 | 1cim | 0.18 | 38 | 2 |
| 1cnx | 1.59 | 8766 | 11 | **1etr** | 3.69 | 2102 | 9 |
| 1ets | 0.78 | 3623 | 8 | **1ett** | 4.58 | 0 | 0 |
| **1fmo** | 0.10 | 8 | 2 | 1gsp | 0.51 | 6 | 2 |
| 1htt | 0.06 | 30 | 2 | **1icm** | 2.20 | 2292 | 13 |
| 1icn | 1.79 | 3316 | 17 | 1nnb | 0.46 | 1428 | 4 |
| 1nsc | 0.42 | 1676 | 4 | 1nsd | 0.25 | 1521 | 4 |
| 1okl | 0.22 | 105 | 3 | 1phd | 0.22 | 42 | 2 |
| **1phf** | 2.98 | 8 | 2 | 1phg | 0.56 | 1034 | 4 |
| **1pph** | 5.22 | 806 | 6 | 1rhl | 0.30 | 5 | 2 |
| 1rls | 0.31 | 6 | 2 | **1seb** | 4.72 | 41011 | 25 |
| 1ses | 0.18 | 33 | 2 | 1sre | 0.37 | 106 | 5 |
| 1tng | 0.18 | 27 | 2 | 1tni | 0.50 | 696 | 5 |
| 1tnj | 0.22 | 145 | 3 | 1tnk | 0.29 | 452 | 4 |
| 1tnl | 0.29 | 13 | 2 | 1tpp | 0.22 | 662 | 4 |
| 2csc | 0.28 | 746 | 4 | 2ctc | 0.76 | 129 | 4 |
| **2ifb** | 2.18 | 2876 | 15 | 2xim | 0.34 | 1046 | 5 |
| 2xis | 0.37 | 1080 | 5 | 3cpa | 0.40 | 1097 | 5 |
| 3ert | 1.27 | 1657 | 6 | 3ptb | 0.22 | 16 | 2 |
| 3tmn | 0.49 | 1943 | 6 | 4tim | 0.30 | 323 | 3 |
| 5abp | 0.13 | 32 | 2 | 5tln | 0.52 | 3803 | 6 |
| **6cpa** | 2.37 | 5171 | 12 | 6tim | 0.29 | 435 | 4 |
| **6tmn** | 4.66 | 6149 | 12 | | | | |

**Table 4.1**: Predictive performance using the default parameter set for the validation co-crystals. 90 percent were predicted within 2.0 Å

### 4.1.2 Tough cases

Ligands for which the number of rotatable bonds exceeds ten represent the most challenging test cases of the validation set. Whereas brute force suffices for simpler ligands, these cases require a delicate balance between exhaustive sampling of the conformation space and maintaining a tractable set of conformations. To better illustrate the efficacy of method, a collection of notable successes and failures are explained in Section 4.1.2.1 and Section 4.1.2.2, respectively. Since the default parameter configuration often failed to yield satisfactory solutions, parameter tweaks for each case are listed in Table 4.3. The success rate among these was 87.5 percent and is detailed in Table 4.2.

| PBD code | $N_{rot}$ | RMSD |
|---|---|---|
| 1apt | 11 | 1.25 (update) |
| 1cnx | 11 | 1.65 (1.14) |
| 1ets | 11 | 0.78 () |
| 1icn | 11 | 2.04 (1.73) |
| 2ifb | 11 | 1.40 (1.08) |
| 1seb | 11 | 1.70 (update) |
| 5tln | 11 | 0.85 (0.55) |
| 6cpa | 11 | 1.34 (1.17) |

**Table 4.2**: Performance of the algorithm on the challenging cases

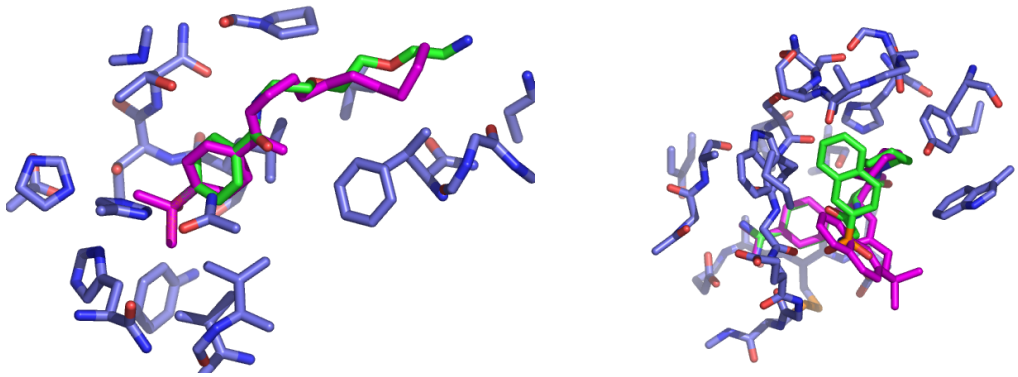
| case | div weight | ala/nonala | burial | code version |
|---|---|---|---|---|
| 1apt | linear | alanized | 0.4 | 090301 |
| 1cnx | sublinear | nonalanized | dflt | 090301 |
| 1icn | sublinear | nonalanized | dflt | 090301 |
| 1ets | sublinear | nonalanized | dflt | 090301 |
| 1seb | linear | non alanized | dflt | 090301 |
| 2ifb | sublinear | nonalanized | dflt | 090301 |
| 5tln | sublinear | nonalanized | dflt | 090301 |
| 6cpa | sublinear | nonalanized | dflt | 090301 |

**Table 4.3**: Additional parameter settings for improving results for the challenging cases

#### 4.1.2.1 Acceptable solutions

Among the challenging cases listed in Table 4.2, 1cnx, 1ets, 6cpa, and 5tln performed particularly well using the default parameters. In this section, the key interactions within the binding site are discussed.

**1cnx**: The sulfonamide inhibitor in carbonic anhydrase II (1cnx) [36] is predicted within 1.65 (1.14) Å of the co-crystal reference in Fig. 4.2(a), using its sulfonamide benzene as a base anchor. The ligand backbone aligned with the co-crystal, while its amine-terminated alkyl chain was positioned less accurately. The crystal structure exhibits a weak hydrogen bond between the terminal amine and the Gln136 amide (4.0 Å) that constrains the alkyl chain position and this was not reflected in the predicted model. Nevertheless, the predicted structure forms hydrogen bonds with Thr199, His96, and Gln96, which appear to be responsible for most of the stabilization energy. Curiously, the MPSIM energy for the minimized predicted conformation was 66.47, while the best-ranked structure from *moleculeGL* was 31.51.



(a) (b)

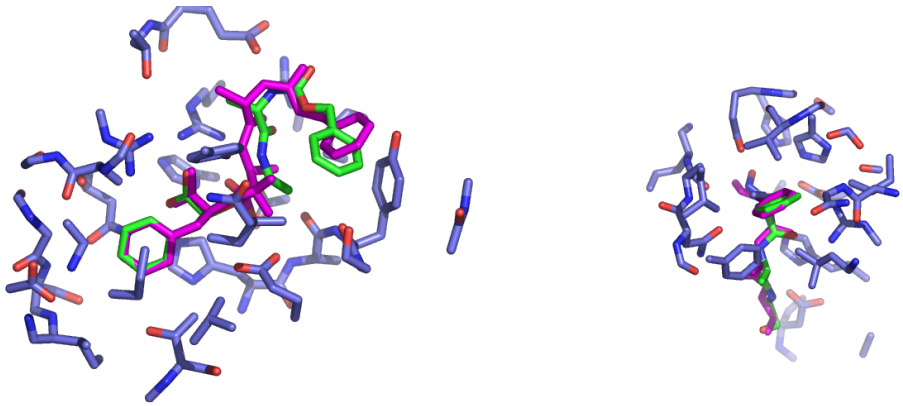
**Figure 4.2**: The predicted solutions (pink) overlaid with the reference ligand (green) for **(a.)** 1cnx and **(b.)** 1ets

**1ets**: The 2-napthalenesulfonic acid inhibitor in bovine thrombin (1ets) [37] is predicted within 0.78 () Å of the co-crystal reference in Fig. 4.2(b), using the napthyl group as the base anchor. This anchor is situated along the protein surface and is wedged from the top by Ile174 and Glu97 and from the bottom by the remainder of

the ligand. The carbonyl portion of the piperidine substituent points toward Lys60 but, at 4.9 Å, is beyond normal hydrogen bonding distance. However, the crystal structure has a crystallized water within this region that ordinarily stabilizes this interaction. Additional interactions include hydrogen bonding between the sulfamide and Gly216 backbone, as well as a salt-bridge between Asp189 and the benzamidine group. The MPSIM energy for the minimized predicted conformation was 45.08, while the best-ranked structure from *moleculeGL* was 51.87.

**6cpa**: The phosphonate in carboxypeptidase A (6cpa) [38] is predicted within 1.34 (1.17) of the crystal structure, using the phenyl group as an anchor. This anchor is secured by a salt-bridge between the carboxylic acid on the $\beta$ carbon and the Arg144 side chain (*See* Fig. 4.3(a)). The remainder of the inhibitor is anchored by its two peptide groups bound to Tyr248, Arg127, and Glu163. While the predicted conformation captures most of the anchoring, the terminal carbonyl varies considerably from the reference structure. Since this group binds loosely along the protein surface, successful binding is driven by relatively weak VDW interactions that are difficult to capture with this methodology. The MPSIM energy for the minimized predicted conformation was 18.19, while the best-ranked structure from *moleculeGL* was 35.00.

**5tln**: The hydroxamic acid-based inhibitor in thermolysin (5tln) [39] was predicted within 0.85 (0.55) of the co-crystal, using the well-buried hydroxamic acid group as the base anchor. (*See* Fig. 4.3(b)). This anchor is part of a peptide-like (Phe+Ala+Gly) chain that terminates with a nitrobenzene. The hydroxyamide group forms hydrogen bonds with Glu143, His142, Tyr157, Glu166, and the backbone of Phe114. While the peptide backbone appeared to be adequately placed, the nitrobenzene group was ill-positioned. However, given that this component represents only a small portion of the ligand binding energy, it is anticipated that the solution is a sufficient candidate for refinement. The MPSIM energy for the minimized predicted conformation was -11.25, while the best-ranked structure from *moleculeGL* was not obtained.



(a) (b)

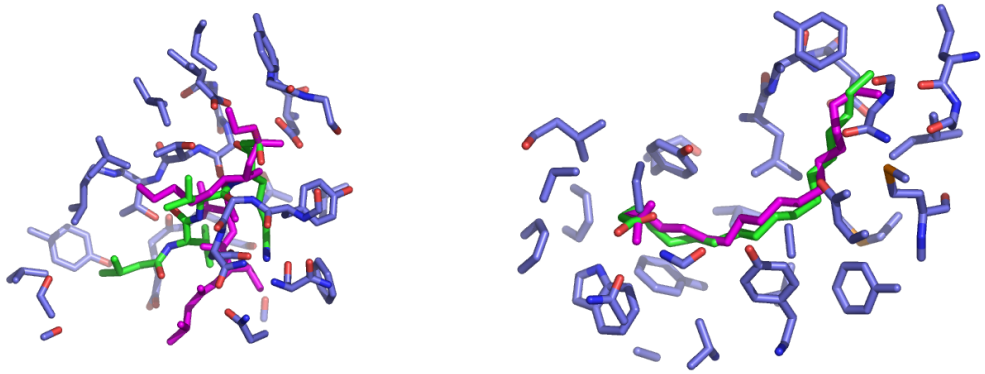
**Figure 4.3**: The predicted solutions (pink) overlaid with the reference ligand (green) for **(a.)** 6cpa and **(b.)** 5tln

#### 4.1.2.2 Unacceptable solutions

In general, there are several reasons that explain *moleculeGL*'s inability to identify acceptable solutions. The first and most basic reason is that increasing numbers of rotatable bonds lead to an exponential growth in the conformation space and without coarsening the search to some extent, the problem becomes intractable. Thus, to operate within memory bounds the density of solutions in a given sampling volume must be reduced, which in turn weakens the possibility of finding a good candidate. Lastly and not insignificantly, small changes in the dihedral angles of ligands can lead to vastly different structures, which implies that the margin of error is small.

A variety of physical arguments help explain the algorithm's shortcomings as well. For some of the longer ligands, the LBD is found along the protein surface which greatly expands the space to be probed. Moreover, the protein surface also often presents numerous local minima that obfuscate the search. Given this scenario, the search engine must generate a finite number of uniformly distributed rotamers that may be too coarse to capture solutions accurately. Additionally, ligands that primarily rely on weak hydrophobic forces for stabilization, such as fatty acid co-crystals like 1icn, typically require more comprehensive sampling as there are no strong polar interactions to guide the search. Increasing the VDW to the full values may benefit these cases, but this bias is not representative of trial conditions for which some

error in the protein structure determination can be expected. Lastly, in some cases, co-crystallized waters help stabilize ligand positions and predicting these *a priori* is a difficult subproblem by itself. Despite these limitations, the unacceptable cases yield qualitatively reasonable results that may be rectified with additional refinement approaches like simulated annealing.



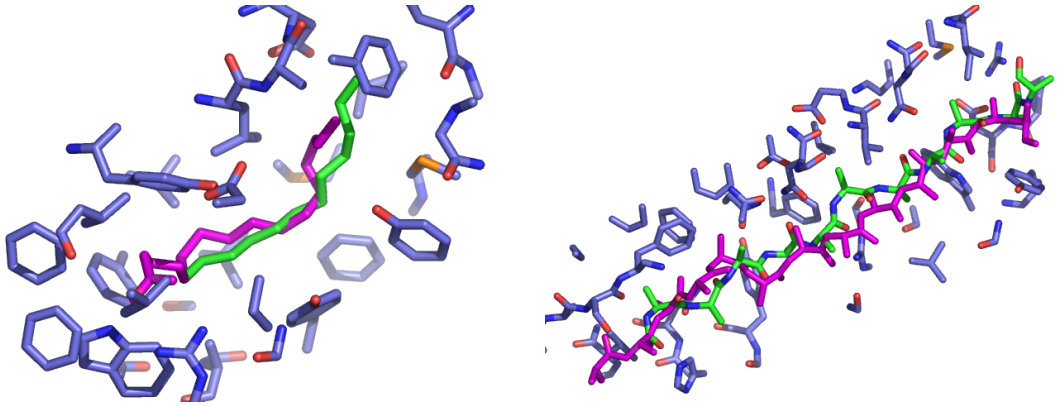
(a) 1apt (b) 1icn

**Figure 4.4**: The predicted solutions (pink) overlaid with the reference ligand (green) for **(a.)** 1apt and **(b.)** 1icn

**1apt**: The pepstatin analog bound in aspartyl proteinase penicillopepsin (1apt) [40] was predicted within 1.25 (update) Å of the reference structure (*See* Fig. 4.4(a)). The ligand is a peptide derivative with much of its stability derived from hydrogen bonds between its main-chain polar atoms and receptor. Reasonable hydrogen bonds were established with Gly76, Asp77, Thr216, and Thr217, although there was difficulty in correctly placing the terminal isoleucine. Ordinarily the backbone amide of the ligand participates in a hydrogen bond with Thr217, but in the predicted structure, the carbonyl was solvent exposed, which forced the isoleucine backbone into an incorrect position. Overall, the MPSIM energy for the minimized predicted conformation was 101.3, while the best-ranked structure from *moleculeGL* was 164.14. Interestingly, this case benefited from alanization of the binding site, whereas using the native residue positions led to inferior results.

**1icn**: The prediction of myristate in rat intestinal fatty-acid-binding protein (1icn) [41] was predicted with an RMSD of 2.04 (1.73) (*See* Fig. 4.4(b)), using the acetate

as the lead anchor. Aside from this acetate, the ligand is exclusively hydrophobic, which means the conformations were guided by relatively unspecific VDW interactions. Therefore, considerable error accumulated during sampling which lead to non-optimal solutions despite obtaining qualitatively reasonable answers. The MPSIM energy for the minimized predicted conformation was 36.27, while the best-ranked structure from *moleculeGL* was 48.14.



(a) (b)

**Figure 4.5**: The predicted solutions (pink) overlaid with the reference ligand (green) for **(a.)** 2ifb and **(b.)** 1seb

**2ifb**: The predicted conformation of palmitate in rat intestinal fatty-acid-binding protein (2ifb) [42] was sampled from the acetate group and was predicted at 1.40 (1.08). The position of the alkyl chain was qualitatively correct (*See* Fig. 4.5(a)), but the disparity is again due to the deficiency of predicting nonpolar entities in a nonpolar environment. The MPSIM energy for the minimized predicted conformation was 2.60, while the best-ranked structure from *moleculeGL* was 17.3.

**1seb**: Prediction of the peptide-based superantigen in the human MHC class II glycoprotein HLA-DR1 (1seb) [43] is the most challenging co-crystal in the validation set. Not only does the large number of rotatable bonds complicate the search, the compound binds in a groove along the protein surface and thus a considerable free volume must be probed. As shown in Fig. 4.5(b), the predicted structure does in fact bind along this groove, although the ligand has poorly buried kinks in several regions. This suggests that there is room for improvement in the burial term. Despite this,

a large percentage of the hydrogen bonding ligand atoms found achieved reasonable geometries with donor and acceptors of the LBD. Ultimately, a strategy may need to be developed that enforces hydrogen bonds with the protein, while allowing for some percentage to interact with the solvent. Also, this case performed better when a linear weighting scheme was employed, as opposed to the sublinear term used for the other examples. This may indicate that a greater emphasis on burial is needed for generating reliable results.

#### 4.1.2.3 Failure analysis

In addition to the qualitative discussion above, an analysis of the failures in the aformentioned cases is summarized in Table 4.4 In this table, the details about the iteration in which the most accurate conformation was lost is recorded, including the number of conformations before and after sorting. The last two columns report the results of the hierarchical filtering strategy (Section 2.5.4.5) applied to the conformation pool prior to diversity sorting. The purpose of this post-processing is determine whether additional procedures not in the current *moleculeGL* implementation may have helped retain good conformations.

One important observation is that the best conformation is typically lost very early in the search. For each of these examples, the conformation pool was reduced from one-half (1apt) to one-fourth (2ifb) of the original size. The diversity rankings ranged from a very reasonable 83rd percentile for 1apt to a dismal 33rd percentile for 2ifb. Therefore, it is clear that merely adjusting the diversity cutoff will not offer a consistent performance gain. If instead the hierchical filtering strategy were applied, all cases would have yielded at least a few conformations with RMSDs less than 2.0 and one case with an RMSD less than 0.5. These data therefore suggest that combining diversity with the hierarchical filtering scheme may improve the odds of reliable conformations.

| Code | Iter lost | In | Out | Div rank/Pct | Filter(2.0) | Filter (0.5) |
|---|---|---|---|---|---|---|
| 1apt | 3 | 1070 | 510 | 83.2 | 5/267 | 1/267 |
| 1icn | 4 | 1361 | 515 | 49.9 | 4/340 | 1/340 |
| 1seb | 6 | 2044 | 766 | 88.0 | 4/511 | 1/511 |
| 2ifb | 4 | 2844 | 631 | 33.5 | 10/711 | 1/711 |

**Table 4.4**: Post-processing analysis of the iterations in which the best RMSD conformation was discarded. The table reports the PDB code and failed iteration, the number of conformations before and after sorting, the diversity percentile of the discarded solution and results of the hierarchical filtering protocol

## 4.2 Validation of Selection Criteria

The effectiveness of torsion sampling is measured by the accuracy in predicting the ligand binding reflected in an X-ray co-crystal. The concept of accuracy assumes several notions: one is that the co-crystallized ligand (or hereafter, the reference) is at a global minimum and all other nearby configurations have equivalent or greater energies, as in Section 4.2.1; two, a prediction within 2.0 Å RMSD of the reference is a sufficiently close guess such that minimization either improves or does not change the RMSD with respect to the reference, as explained in Section 4.2.2; three, an accurately predicted conformation is sufficient to estimate its binding affinity, as discussed in Section 4.2.5. Finally, it is shown in Section 4.2.3 that a subset of generated conformations can be identified that has a high probability of containing at least one accurate solution.

### 4.2.1 Energy landscape near the global minimum

In Fig. 4.6, a set of conformations varying in proximity to the reference structure is scored with the Dreiding FF. The results reflect that the reference ligand energy is below that of all other nearby poses and is the global minimum. Since a protein-ligand X-ray structure determination is at thermal equilibrium, thus it is expected that the bound ligand is near the global energy minimum. Moreover, within a small neighborhood of the reference, the potential energy surface should be smooth and

monotonic. These trends are observed in the results and thus the use of the Dreiding FF is justified. Although the data is not perfectly monotonic, the noise can be attributed to error derived from quantum mechanics data used to fit the FF, which is generally no less than 3.0 kcal/mol in magnitude [44].

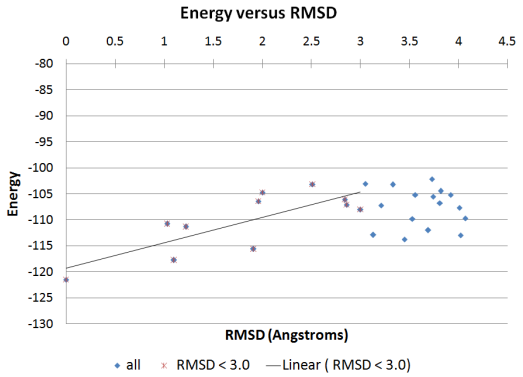



**Figure 4.6**: Comparison of energy score versus RMSD

### 4.2.2 Minimization of nearby solutions

In Fig. 4.7 it is equivalently shown that a structure within a small neighborhood of the global minimum (2.0 Å) can be minimized to a position and energy that resembles that of the reference. Beyond this neighborhood, minimization has virtually no chance of recovering the reference structure, which is usually indicative of a pose that is distinct from the global minimum. Hence, this motivates this value as an upper limit for an accurately predicted conformation. As will be shown in Section 4.2.5, conformations predicted below this value also yield reliable thermodynamic data.

### 4.2.3 Selection of final conformations

Given the inexactness of molecular simulation, the generation of a structure that precisely matches the global minimum is a rare event. Instead, an ensemble of near-matches are expected that are close, but nonetheless have energies less favorable than the reference. Depending on how far these solutions deviate from the global minimum,

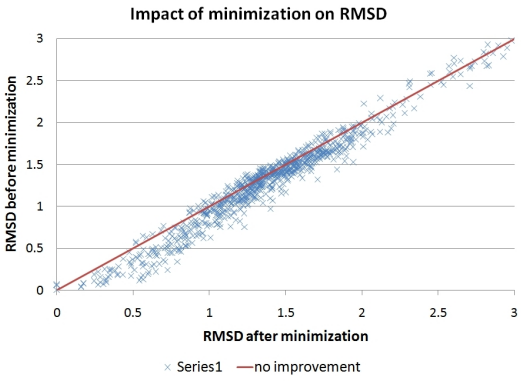



**Figure 4.7**: RMSDs for all generated conformations before and after minimization. Structures below the solid line improve RMSD after minimization, while those above worsen

their energies may be indistinguishable or even greater than nearby (but incorrect) poses. For this reason a subset of solutions and not a single entry should be retained from a given simulation run. Thus, in this section, the size of this subset is explained and forms the basis of the **numFinalFamilies** parameter.

The cumulative probabilities of low RMSD structures are plotted in Fig. 4.8. It was anticipated that for a well-sampled system, the lower portion of the energy spectrum would be dominated by low RMSD structures, while the higher energy region would be populated by poor guesses. In this way a confidence criterion could be optimized, but unfortunately, a clear preference was not observed in the data. This could be due to any number of factors, including using too small of a sample population or the FF insufficiently describing off-equilibrium energies accurately. However, given that there is still a finite probability of finding a low-RMSD conformation within the top 150 ranked conformations, this value is used as the default for **numFinalFamiles**.

### 4.2.4 Refinement of *moleculeGL* solutions

*moleculeGL* is intended to provide a comprehensive set of potential strongly binding ligand poses. As such, further refinement of the output structures is of paramount importance for obtaining reliable binding constants. Minimization of the ligand in the

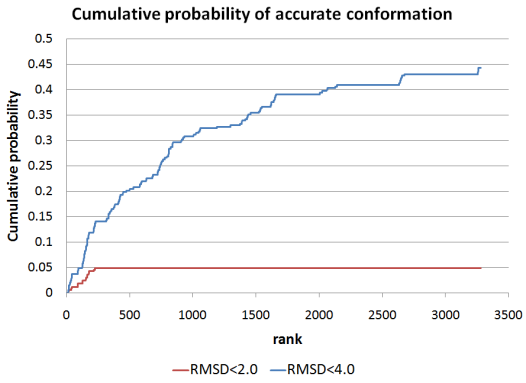



**Figure 4.8**: Cumulative probability of finding an accurate solution at or below 0.5, 1.0 and 2.0 Å RMSD .

presence of a fixed protein is the next logical step in the refinement process. Provided for the purpose of illustration is an example hierarchical approach for isolating low RMSD structures when provided a large set of fully constructed ligands. These steps score, rank, and discard the conformation pool according to

1. the coarse-grain force field
2. the fine-grain force field
3. MPSim minimization and scoring

To demonstrate the refinement power of each step,this procedure was applied to the final set of 450 conformations generated for 5tln. In Fig. 4.9 the number of low-RMSD conformations missed for increasing score cutoffs are plotted. This chart portrays the ability of each scoring step to appropriately order the low-RMSD solutions, which in general should have the most favorable interaction energies. For instance, if fifty percent of the original conformation pool were selected based on the coarse-grain energy alone, approximately 15 of the 55 total low-RMSD conformations would have been lost. This is an improvement over a completely random ordering of conformations, for which the probability of losing one half of the desired conformations is 50%. In this framework, the best metrics will concentrate the density of good candidates toward lower scores (left on the x-axis). Moreover, the sharper the peak

near x=0, the more selective the measure.

Several interesting trends are observed in these data. First, all three measures effectively concentrate the low-RMSD structures in the lowest score percentiles. Second, the fine-grain and coarse-grain curves have roughly the same shape, which indicates that the more detailed fine-grain scores offer little or no improvement of the coarse-grain scores. Nevertheless, even a liberal threshold which discards fifty percent of the conformation pool would have retained over seventy percent of the low-RMSD conformations. Third, MPSim step increases the number of low-RMSD conformations to 83 from 55, which is better than a fifty percent improvement and is a testament to its ability to ameliorate near-misses. Fourth, this step generates a more sharply peaked distribution, which suggests its superior discrimination power over *moleculeGL*'s scoring engines.

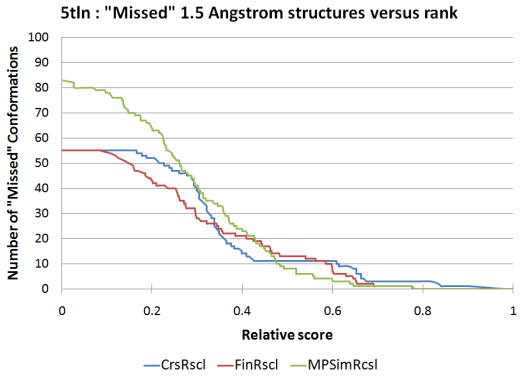



**Figure 4.9**: Number of good conformations lost as a function of threshold value for coarse-grain (blue), fine-grain (red) and MPSim (green) energies

Given that both *moleculeGL* and MPSim do reasonable well at isolating the low-RMSD structures, the use in tandem should at a minimum offer an improvement in efficiency. However, increasingly accurate (and very likely more computationally demanding) measures are needed to obtain more sharply peaked distributions and thus greater discrimination power. Ultimately, nanosecond-timescale molecular dynamics of both the protein and ligand in explicit solvent is the golden standard for obtaining an ensemble of optimal binding modes, but this requires considerable computational

resources and are beyond the scope of this validation study.

### 4.2.5 Binding affinity of trypsin

The motivation for achieving a high level of accuracy is that these solutions are the most likely to offer reliable estimates of the ligand-protein free energies of binding. To demonstrate the correspondence between a successfully predicted ligand conformation and its experimentally observed binding affinity, a set of eight trypsin inhibitor co-crystals [32, 45] were sampled via *moleculeGL* and the results are summarized in Table 4.5. These molecules generally feature a p-guanidino phenylalanine ring lodged deeply into the protein interior with largely aliphatic substituents of varying length extruding toward the protein surface. These chains terminate with a nitrogen atom that is stabilized via hydrogen bonding with neighboring residues in the binding cavity. 1pph is an exception to this trend, as the substituent is instead an N-tosylated piperidide.

The binding affinity of the predicted trypsin co-crystals was compared to the natural log of the experimental dissociation constants [32], $K_i$ (Fig. 4.10), which are reflections of the strength of binding. Increasingly negative values of $\ln[K_i]$ suggest a higher degree of inhibition and thus are negatively correlated with the predicted binding affinities. *in vitro*, the observed inhibition constants for these compounds are relatively weak (in the micromolar range), but nevertheless the compounds exhibit strong interactions with the protein interior.

As shown in Fig. 4.10, barring the outlier 4-phenylbutylamine (1tni), the overall correspondence is in good agreement with experimental inhibition values ($R^2$=0.87). Although 1tni was predicted within 2.0 Å of the crystal structure, the incorrectly placed polar amine did not recover the stabilizing hydrogen bond with the LBD. Therefore, a lower than expected binding affinity resulted, despite extensive minimization of the complex.

Aside from this outlier, the correlation could have been improved by including co-crystallized waters, as was done in a prior study [46]. Inclusion of these waters were

| co-crystal | energy | ln[$K_i$] | RMSD |
|---|---|---|---|
| 1tng | 29.5 | -2.94 | 0.18 |
| 1tni | 12.0 | -1.70 | 1.00 |
| 1tnj | 29.0 | -1.96 | 0.21 |
| 1tnk | 32.1 | -1.49 | 0.45 |
| 1tnl | 30.1 | -1.88 | 0.29 |
| 1tpp | n/a | unk | 0.28 |
| 1pph | 48.2 | -6.23 | |
| 3ptb | 40.2 | -4.74 | 0.22 |

**Table 4.5**: Energies (kcal/mol) for the trypsin inhibitors were obtained with MPSim and the associated binding constants, $K_i$, are from [32]. *The energy values were inferred from Fig. 4.10*

found to be indispensable for computing accurate binding affinities. Additionally, unpublished research suggests that neutralizing the charges on residues may give more stable energies, thus this assessment may benefit from this approach. Nonetheless, these results suggest that structures to within 2.0 Å of the global minimum are sufficiently accurate for obtaining realistic interaction energies without resorting to molecular dynamics.

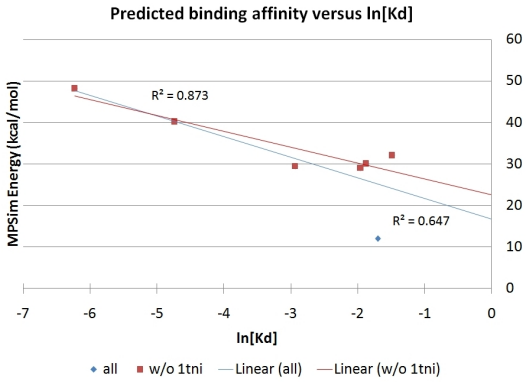
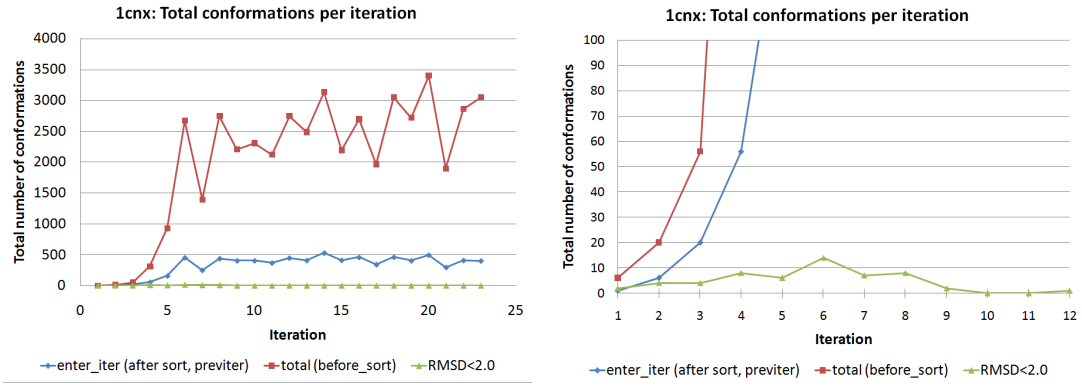



**Figure 4.10**: The predicted binding affinities of the trypsin inhibitors correlate well with experiment ($R^2$=0.87) when 1tni is omitted. Including this outlier reduces the coefficient of correlation to 0.64 .

## 4.3 Error Analysis

While the *moleculeGL* performed well in predicting the aforementioned cases, it is nevertheless important to understand under which conditions it fails. For this section, 1cnx is used as a representative example given its large number of rotatable bonds. In almost all cases where a suitable solution was not in the final conformation pool, it was because either because it was not generated in sampling or retained during sorting. This observation is illustrated for the 1cnx in Fig. 4.11. The first subfigure is a plot of the total number of conformations before (red) and after (blue) sorting for 1cnx as a function of iteration. Marginally visible in green are the number of conformations whose RMSDs are less than 2.0Å.

As expected, the algorithm maintains a consistent number of post-sorting conformations after the diversity algorithm is applied to the total set of conformations. Shifting to the second subfigure, it is evident for the first iterations that the number of low RMSD conformations increases while the total number of sorted conformations remains fixed. By the sixth iteration, however, the numbers begin to thin, primarily because the algorithm cannot consistently distinguish good poses from the overall pool. Therefore, in cases such as these, by the time the ligand is completely sampled, there are no suitable candidates for refinement.




(a) (b)

**Figure 4.11**: Plot of the total number of conformations before and after sorting for 1cnx as a function of iteration. The number of conformations below 2.0 Å with respect to the reference ligand is plotted in green. **(b.)** is a zoomed in version of **(a.)**.

### 4.3.1 Density of solutions

The first step to understanding this limitation is to investigate the density of solutions needed to retain good approximations of the crystal conformations. By varying **numRotamers** for different sampled chain lengths, the density of accurate solutions can be analyzed. Based on sampling data obtained for 1seb in Table 4.6, several trends were observed. For increasing values of **numRotamers**, a higher density of low-RMSD structures were retained (assuming 150 final conformations are returned in all cases). This high number of conformations is preserved through several iterations of diversity, until a precipitous drop in accuracy is observed (iteration 10 for 3 rotamers, iteration 8 for 9 rotamers). Understanding this drop in accuracy will likely be a key step in improving the algorithm's performance. It was also noted that a larger value for **numRotamers** yields conformations that are considerably closer to the crystal structure for the initial few iterations, but this edge was quickly lost as diversity was applied. Therefore, a modest number of rotamers can be expected to yield a reasonable density of good candidates, while larger values may actually result in sub-par performance.

### 4.3.2 Saturation of initial sampling iterations

One improvement integrating into the *moleculeGL* code was a requirement that the search saturates the first sampling iterations, as opposed to the linear growth enforced in subsequent iterations. Fig. 4.12 plots the density of low-RMSD solutions obtained when a higher number of conformations are generated in the initial sampling iterations versus using a constant number throughout. It was postulated that generating a higher density of solutions in the initial sampling iterations could yield a greater number of low-RMSD structures for subsequent iterations. This was based on the reasoning that a sparse sampling would yield few candidates for future samplings and thus have a low probability of surviving diversity pruning. Alternatively, a high density of low-RMSD structures would not only have strength in numbers for passing the diversity step, they have a higher likelihood of achieving favorable energies compared

| **numRotamers** | 3 | | | 6 | | | 9 | | | 12 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| chain length | suc | near | div | suc | near | div | suc | near | div | suc | near | div |
| 1 | 3 | 999 | | 6 | 999 | | 9 | 999 | | 12 | 999 | |
| 2 | 9 | 0.6 | | 36 | 0.35 | | 81 | 0.24 | | 144 | 0.18 | D |
| 3 | 17 | 0.6 | | 77 | 0.36 | | 150 | 0.24 | D | 150 | 0.2 | D |
| 4 | 17 | 0.6 | | 85 | 0.36 | D | 150 | 0.24 | D | 150 | 0.28 | D |
| 5 | 35 | 0.6 | | 150 | 0.5 | D | 143 | 0.38 | D | 138 | 0.69 | D |
| 6 | 35 | 0.6 | | 150 | 0.36 | D | 150 | 0.24 | D | 150 | 0.38 | D |
| 7 | 35 | 0.6 | | 144 | 0.63 | D | 150 | 0.38 | D | 150 | 0.55 | D |
| 8 | 35 | 0.6 | | 149 | 0.36 | D | 66 | 0.94 | D | 89 | 0.97 | D |
| 9 | 35 | 0.6 | | 150 | 0.57 | D | 0 | 999 | | 46 | 0.98 | D |
| 10 | 35 | 0.6 | | 18 | 1.24 | D | 0 | 2.16 | D | 12 | 1.41 | D |
| 11 | 35 | 0.6 | | 6 | 1.86 | D | 9 | 1.78 | D | 12 | 1.77 | D |
| 12 | 35 | 0.6 | | 18 | 1.67 | D | 0 | 2.99 | D | 0 | 2.59 | D |
| 13 | 6 | 0.6 | D | 6 | 1.38 | D | 9 | 1.73 | D | 11 | 1.76 | D |
| 14 | 6 | 0.95 | D | 0 | 3.04 | D | 0 | 3.91 | D | 0 | 5 | D |
| 15 | 6 | 0.95 | D | 6 | 1.95 | D | 0 | 5.33 | D | 0 | 4.31 | D |
| 16 | 6 | 0.6 | D | 0 | 2.9 | D | 0 | 2.43 | D | 0 | 4.91 | D |
| 17 | 3 | 5.89 | D | 0 | 5.76 | D | 0 | 4.05 | D | 0 | 3.43 | D |
| 18 | 3 | 4.97 | D | 0 | 999 | | 0 | 9.09 | D | 0 | 5.61 | D |

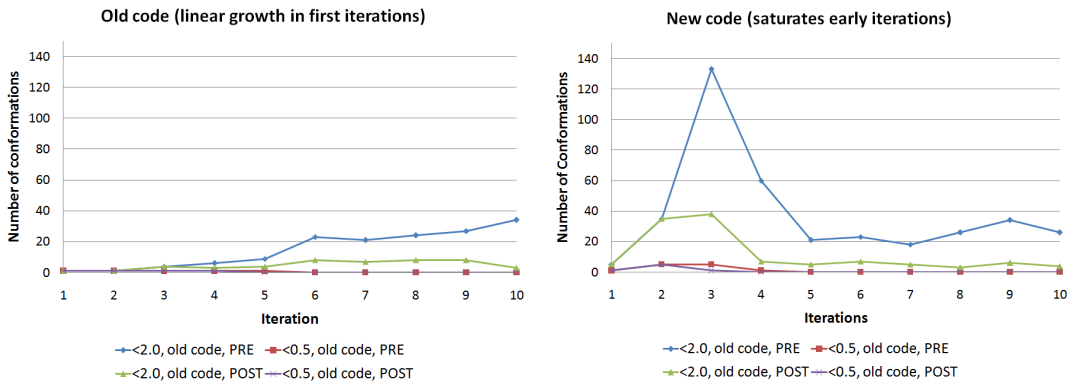
**Table 4.6**: Assessment of the number of accurately predicted (RMSD≤2.0) conformations for different chain lengths. Also reported are the lowest RMSD conformations and whether diversity (D) was applied

to the higher RMSD solutions.

These plots demonstrate that increasing the sampling density in the early iterations does improve the number of low-RMSD conformations for several iterations thereafter, but this advantage dissipates as the search proceeds. It is likely that the diversity engine discards a high number of good structures because of their relatively small intermolecular RMSD in comparison to the more sparsely sampled structures.

### 4.3.3 Filtering schemes

Fig. 4.13 demonstrates the efficacy of the filtering strategies described in Section 2.5.4.5 as a function of iteration. For Fig. 4.13(a) the percentage of low-RMSD conformations within the total conformation pool is plotted in log units against the iteration

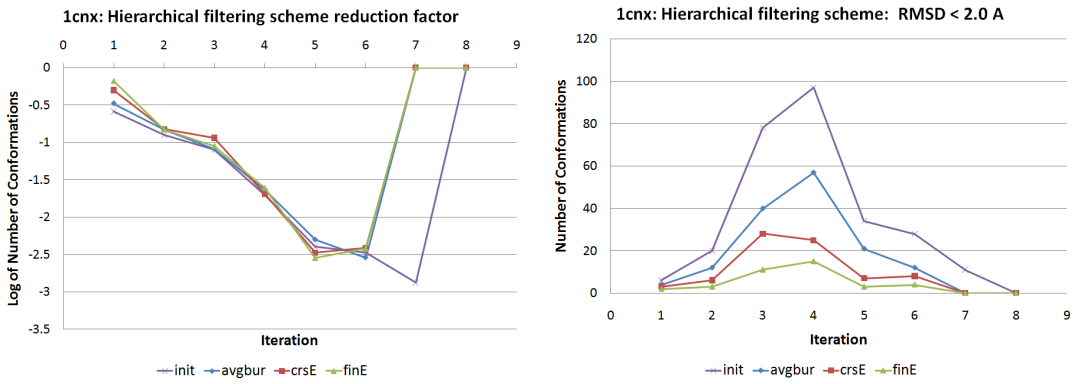



(a) Without saturation (b) With saturation

**Figure 4.12**: Plots summarizing the number of conformations below 2.0 and 0.5 Å with respect to the reference ligand as a function of sampling iteration, before and after sorting. **(a.)** plots the conformation distribution after saturating the first iterations, while **(b.)** is the original procedure

number. As subsequent filters are applied, this percentage should improve and by substantial margins. This is clearly shown at the first iteration, where the percentages are roughly 25%, 35%, 50% and 68% at the initial, average burial, coarse energy and fine energy filters, respectively. For later iterations, however, the margins shrink and eventually the percentages actually decrease as filters are applied. This evidences a decreasing efficacy of the filtering techniques as they discard good conformations at a higher rate than the bad. The impact that this loss of efficacy has on the overall conformation pool is depicted in Fig. 4.13(b).

Based on preliminary trials for the cases listed in Section 4.1.2, sorting by this scheme does not offer nearly the discrimination power needed for robust performance. Promising cases such as 1cnx tend to benefit from the scheme, which increases the density of good solutions with each subsequent filter, however, the opposite effect is observed in other cases. This shortcoming arises primarily when the burial criterion is applied. In these situations, it was discovered that a large number of generated conformations had greater burial scores than were exhibited for the reference conformation. That is, several portions of the reference ligand were less buried than other generated structures, which demonstrates the balancing act between optimal burial and overall binding. Therefore, while a hierarchical filtering strategy is generally




(a) (b)

**Figure 4.13**: **(a.)** The number of low RMSD conformations (log units) in the total pool as a function of iteration. **(b.)** Shows the number of low RMSD conformations after each filter is applied

beneficial, it cannot be applied indiscriminately.

## 4.4 Parameters

*moleculeGL* features a multitude of parameters that shape the torsion sampling, conformation sorting, and pose-scoring approaches. In this section, a qualitative assessment of their impact is outlined in Table 4.7, which compares the accuracy for various parameter values. The discussion is intended to follow the same structure as outlined in Chapter 2, although the discourse is limited to those parameters that have the greatest impact.

| Parameter | Value | Run | Success | $Time_avg$ |
|---|---|---|---|---|
| DEFAULT/ | | 0.89 | 0.80 | 1723 |
| | | | | |
| alanization | all | 0.75 | 0.55 | 1451 |
| alanization | hydrophobic | 0.80 | 0.55 | 1450 |
| alanization | polar | 0.78 | 0.60 | 1599 |
| alanization | wag | 0.84 | 0.65 | 1498 |
| burialcumulative | 0 | 0.89 | 0.78 | 1550 |
| burialfilter | 0 | 0.87 | 0.82 | 879 |
| burialpercent | 0.2 | 0.89 | 0.78 | 2100 |
| burialpercent | 0.5 | 0.89 | 0.78 | 1919 |
| coul | on | 0.89 | 0.75 | 1722 |
| DEFAULT | out | 0.89 | 0.75 | 1723 |
| diversityVariable | useCumulativeWeights | 0.87 | 0.76 | 1594 |
| diversityVariable | usePrimaryWeights | 0.89 | 0.75 | 1505 |
| diversityVariable | useSecondaryWeights | 0.87 | 0.80 | 1448 |
| exploreNumRotamers | 12 | 0.85 | 0.71 | 2720 |
| exploreNumRotamers | 3 | 0.89 | 0.84 | 921 |
| hb | linear | 0.89 | 0.75 | 1594 |
| hb | none | 0.89 | 0.75 | 1585 |
| hb | piecewise | 0.89 | 0.78 | 1622 |
| intermedFamilies | 1000 | 0.00 | 0.00 | 9999 |
| intermedFamilies | 100 | 0.85 | 0.67 | 360 |
| intermedFamilies | 1500 | 0.89 | 0.75 | 2645 |
| intermedFamilies | 2000 | 0.89 | 0.78 | 3547 |
| intermedFamilies | 500 | 0.89 | 0.71 | 1017 |
| rmsdcomparison | all | 0.87 | 0.75 | 1525 |
| rmsdcomparison | vectorAllprior | 0.87 | 0.71 | 2368 |
| rmsdcomparison | vectorOnly | 0.89 | 0.71 | 2180 |
| hbfilter | 1 | 0.73 | 0.49 | 390 |
| selfclashfilter | 0 | 0.85 | 0.69 | 1272 |
| strainfilter | 1 | 0.89 | 0.78 | 1610 |
| vdwscale | 0.3 | 0.89 | 0.71 | 1915 |
| vdwscale | 0.6 | 0.89 | 0.75 | 1790 |
| vdwscale | 0.9 | 0.00 | 0.00 | 9999 |

**Table 4.7**: Results for parameter variations. Default values are given in Table 3.2. The *Run* column gives the percentage of jobs that run to completion and *Success* gives the percentage with conformations less the 2.0 Å RMSD

### 4.4.1 TorsionSampling

#### 4.4.1.1 Sampling single iterations

When **focusSampling** is disabled, **numExploreRotamers** defines the number of solutions retained for a given sampling interval. As shown in Table 4.7, 6 conformations are needed at a minimum while 12 provides the greatest accuracy. Greater numbers apparently overwhelm the search algorithm and thus degrade accuracy and speed.

When **focusSampling** is enabled, the intervals identified by Explore are enriched according to **numFocusRotamers**. As shown in Table 4.7 a value of 8 maximizes the overall fitness, but with a significant increase in search time. Ironically, a value of 12 leads to the lowest fitness value, yet has the largest computational expense. This indicates that a large value tends to overwhelm the search. Generally, it is difficult to analyze these parameters in isolation, as their performance is intimately tied to **numIntermediateFamilies**. For this reason, the interplay of these parameters are analyzed in further detail in Section 4.4.2.5.

#### 4.4.1.2 Sampling path and combinatorial sampling

The ordering of secondary branches in the sampling path may play a role in the integrity of sampling, but at this time, the different schema for ordering the branch sampling have not yet been evaluated. It is expected that sampling the longest chain first should be a requisite in any approach, given that its larger size makes placement considerably more difficult than the secondary branches. Moreover, its solutions place strong constraints on the number and relative positions of the secondary branches. For branches that have little or no interaction with other secondary branches, the order in which the secondary branches are sampled should have no appreciable impact on the success of prediction, whereas branches with significant correlation may require an intelligent ordering. Using the same logic stated for the main chain, it is anticipated that sampling secondary branches according to decreasing length would yield the best results.

The sequential approach to branch sampling was employed in favor of the com-

binatorial variation. This choice is reasonable, since the secondary branches in the validation set are on average three rotatable bonds in length, which is generally short enough that cross-interactions are infrequent. The combinatorial approach to ligand construction from secondary branches requires an additional level of complexity that has not yet been implemented. Combinatorial sampling may ultimately be necessary when the set of compatible branch solutions are relatively high in energy compared to the lowest energy conformations in each independently sampled ensemble. This could be evaluated by assessing the relative rank of the reference structure among the predicted branch solutions.

### 4.4.2 ConformationSort

#### 4.4.2.1 Diversity approaches

The **diversityVariable** mode served as the default for the validation testing after preliminary results suggested the **diversityFixed** suffered from undersampling. As was explained in Chapter 2, the combination of the fixed cutoff inherent to the **diversityFixed** approach and the bounds on the conformation pool (**numIntermediateFamilies**) routinely failed to yield conformations that adequately spanned the search space. This was especially evident for co-crystals that bind along the solvent-exposed protein surface. Therefore, the results presented in Table 4.7 only pertain to **diversityVariable** options.

As shown in Table 4.7, **usePrimaryWeights** are essential in obtaining reasonable performance, as this metric favors candidate conformations that represent a cluster of solutions. Use of the **useCumulativeWeights** parameter also appears to give a marginal improvement of the results. This suggests that having some memory of the density in prior iterations improves the chances of retaining good solutions. Lastly, **useSecondaryWeights** further improves the success rate, suggesting the importance of burial in selecting a diverse conformation pool.

#### 4.4.2.2 RMSD approximations

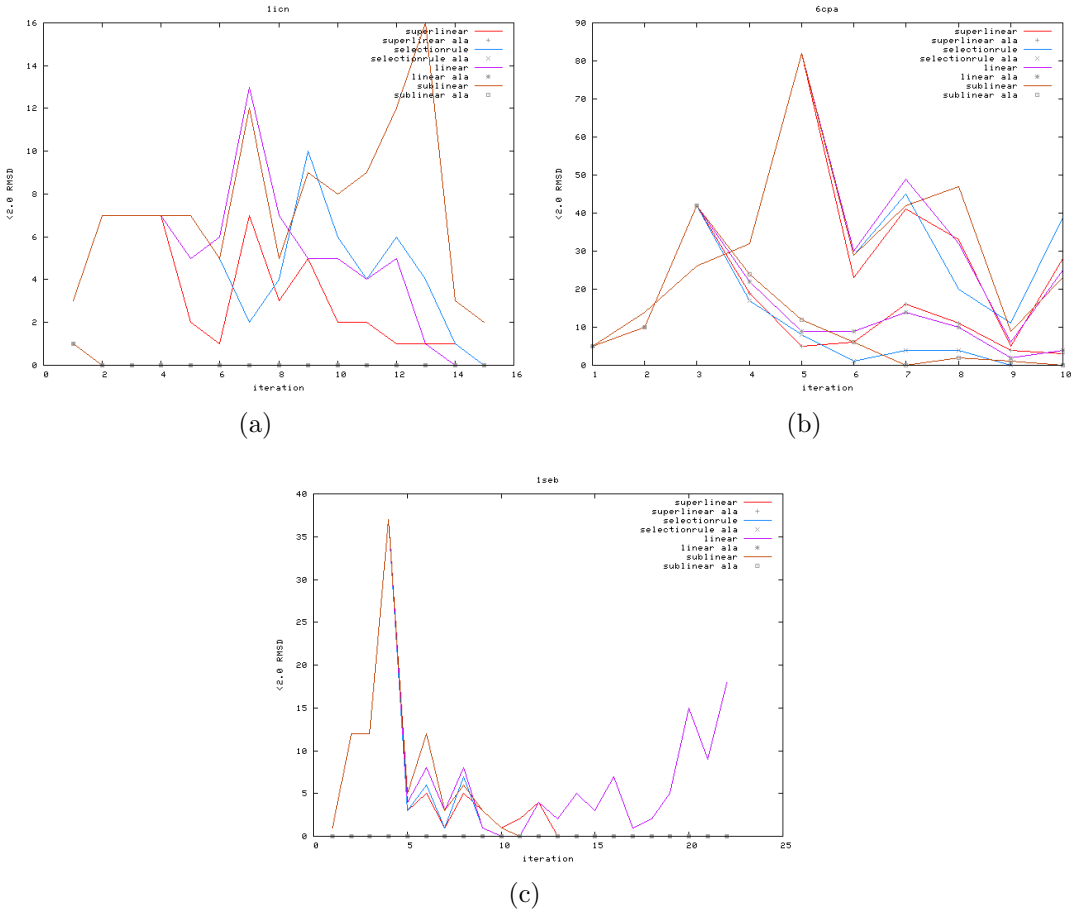
Several approaches for approximating the RMSD between conformations were explained in Section **??**, including **vectorOnly**, **vectorCOMPrior**, **vectorAllPrior** and **all**. Unreported results support the intuitive estimate of **vectorAllPrior** being the most accurate approximation, **vectorCOMPrior** the second best and **vectorOnly** being the worst.

The effectiveness of the available approximations are summarized in Table 4.7. **all** offers the best performance, although **vectorAllprior** is comparable. The inferior **vectorOnly** description substantially degrades the rate of successful prediction. Since the position and orientation of the bond constrains the positions of the downfield clusters, this vector-based RMSD approach is an effective measure for discriminating between growth directions. However, this measure is not unique, as different parent cluster positions can yield similar, if not exact, positions for the same bond. As such, this metric at times is unable to readily distinguish between structurally dissimilar conformations, especially for large diversity values. Thus, by including some additional information about the parent clusters, the RMSD estimate can be improved considerably.

#### 4.4.2.3 Burial-weighted diversity

The sub-linear weighting scheme for diversity gives optimal results for most cases, although for very loose (1seb) and very tight binding interactions (6cpa), the performance can vary. The former cases benefit from the linear weighting scheme, which enforces a stronger dependence on burial. Tightly bound ligands, however, tend to prefer sub-linear dependence, which balances diversity with burial. Under no circumstances were the super-linear weighting or unweighted diversity preferred, which suggests the integral role of diversity and burial in reducing the conformation pool. These conclusions are supported for representative cases in Fig. 4.14. Moreover, this method appears to address the primary drawback of uniform diversity-based sorting, that is, that poorly buried structures tend to increase the minimum diversity

threshold, which in turn reduces the density of solutions in critical regions of the LBD. Lastly, informal studies have demonstrated that **burialAvgNum** and **burialTotalNum** are related by a scale factor unique to each ligand, thus the above conclusions are expected to hold for either parameter.




(a) (b)

(c)

**Figure 4.14**: Plots of the number of good conformations obtained for the various diversity weighting schemes applied to **(a.)** 1icn, **(b.)** 6cpa, and **(c.)** 1seb

#### 4.4.2.4 Sorting filters

**burialFilter**: The first step in initializing the **burialFilter** is choosing an optimal value, based on the total and cluster-based burial percentages exhibited by the reference ligand. As shown in Fig. 4.15, the burial percentages are heavily skewed toward completely buried structures, although there is an appreciable number of partially

buried examples. Therefore, a lenient value of 0.85 was selected. In practice, however, the parameter may need to be adjusted on a case-by-case basis, since overly large values can result in grossly unburied structures, while smaller values may restrict the solutions to unrealistic areas of the protein.

Alternative approaches recently developed included computing the density of protein atoms around a given ligand atom, as opposed to relying on a binary measure. This metric gives higher weight to conformations that bind inside a groove on the protein exterior as opposed to merely hugging the protein surface, as the total number of nearby atom would be greater for the former scenario. The binary test described above cannot distinguish between these two scenarios. Ultimately the burial approach utilized in the **burialFilter** will be replaced with a newer metric, but at the time of publication, this has yet to be completed.

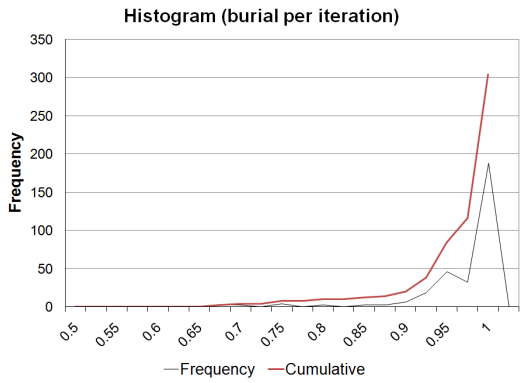



**Figure 4.15**: A histogram of burials measured for partially constructed reference ligands

**selfClashFilter**: The **selfClashFilter** is most effective for ligands with at least five rotatable bonds, since the absence of intramolecular energies in the sampling scoring function often leads to conformations with clashing substituents. Trial cases without this filter were overwhelmed with conformations that appeared to be coiled. Initially it was hoped that the anchor-to-base distance would provide a sufficient description, however, this still yielded a large number of physically unreasonable conformations. Therefore, this filter utilizes on the more expensive option of computing the ligand

nonbond intramolecular interaction energies.

**strainFilter**: It was anticipated that strain would be a sufficient statistic to discriminate between closely related conformations. However, the distribution of conformation energy for the generated ensembles was quite diffuse and therefore not conducive to yielding a statistically significant cut-off value.

**hydrogenBondFilter**: This parameter stipulates that some percentage of a ligand's hydrogen bond donors or acceptors establish bonds with complementary receptor atoms. In practice, the percentages reflected for various co-crystal cases varied considerably and thus a reasonable value for this parameter could not be reasonably determined. As such, blind application of this filter to the validation set had little impact or even worsened the results, as demonstrated in Table 4.7. The primary reason that a more reliable estimate for this parameter could not be obtained was that no attempt was made to quantify the ligand hydrogen bond interactions with co-crystallized waters or the surrounding solvent. These interactions can be especially important for compounds that bind along the protein surface, such as peptide-based ligands. Therefore, this filter may be better suited for ligands that have minimal exposure to the solvent, or for cases in which the positions of cocrystallized waters are well known.

#### 4.4.2.5 Sorting miscellanea

**Sampling density**: Several values of this parameter were tested and the results are summarized in Table 4.7. Based on these data, a value of 1000 offers a suitable balance between accuracy and computational expense. It was also observed that **numRotamers** and **numIntermediateFamilies** should be chosen such that at least four iterations of conformations are stored before sorting. This is a direct consequence of the inability to discriminate viable candidates from the conformation pool when only a few segments have been sampled. A further discussion of this concept in the context of non-bond energies is provided in the subsection below.

**SortingModes**: Both **diversityOnly** and **traditionalSort** utilize diversity for con-

formationSort, although the latter approach includes energy as a selection criterion. Smaller or tightly bound ligands tend to benefit from the **traditionalSort** approach, as diversity is less important for these cases. On the other hand, loosely bound ligands or those with a large number of degrees of freedom (DOF) require ample diversity to adequately probe the sampling space.

The most compelling argument for not relying too heavily on energy scores during sampling is that the scores of partially constructed conformations are generally not well correlated with those of full ligands. While in Section 4.2.1 it was shown that the co-crystallized ligand conformation is at or near the global energy minimum, this statement only holds for fully-constructed ligands. In Fig. 4.16 the probability densities as a function of energy are plotted for the set of conformations generated at iterations four through six of a representative co-crystal. The energies of the reference ligand at each iteration is displayed as points along y = 0.4. These data demonstrate that the reference structure energy is found below the distribution mean, but there is a significant density of solutions at or better than this energy. Specifically, the best conformations fall in the top 5% at the fourth iteration and within the top 3% for the subsequent iterations. This illustrates the risk in not retaining the lowest RMSD structure, when an energy-based or number-based cutoff is used to select a subset of partially constructed solutions. Therefore, metrics such as diversity and burial are more reliable at retaining plausible leads.

**Diversity by Family**: Using the **sortByParents** feature improved sampling somewhat, as a larger number of low RMSD conformations were retained in the final conformation set. This is demonstrated in Fig. 4.17, where the distributions of final conformations with and without this parameter are shown. The key result is the larger population of structures below 2.5 RMSD when diversity sorting by parents is used. This effect is expected to be more pronounced for increasing values of **numRotamers**.

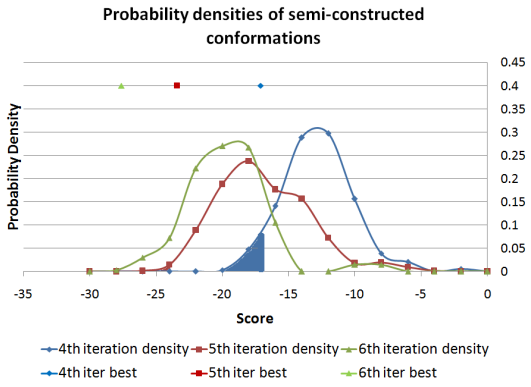



**Figure 4.16**: Probability densities as a function of energy for a set of partially constructed ligands

## 4.4.3 Scoring

### 4.4.3.1 Coarse-grain versus fine-grain scoring

In Fig. 4.18 the coarse-grain (CG) and fine-grain (FG) scores for a simulated set of atom pairwise distances are plotted. Since the piecewise functions return 0.0 for atom pairs that are above the cutoff distance, the scores for these pairs will deviate from the full-grain representation. For the non-zero entries, $R^2 = 0.83$, which suggests that the CG values are indeed a good approximation up to a scale factor. Since the emphasis in rotamer generation is on filling the open space of a protein and capturing hydrogen bonds, this coarse-grain representation is appears to be sufficient.

### 4.4.3.2 van der Waals functions

**vdwRadiiScale**: Reduced VDW radii were used to expand the available search space within the receptor. This tolerance is crucial, as the exact positions of atoms from X-ray data is not always well resolved and moreover, the resolved positions are statistical averages based on thermal fluctuations. Moreover, the coarseness of the search can lead to non-optimal placement of solutions that might ordinarily be precluded with full-scale radii.

Trials for this study used a scaling factor of 0.85 for heavy atoms and 0.50 for

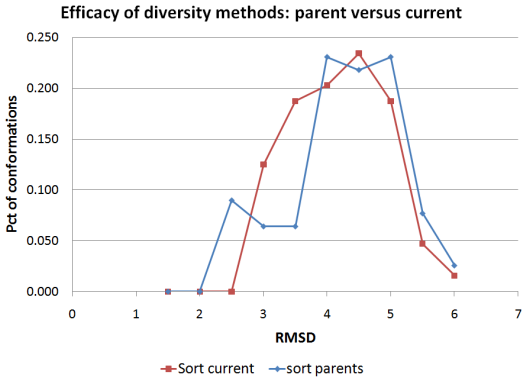



**Figure 4.17**: The distribution of conformations for 1icn with and without **sortByParents** enabled. A higher density of good conformations are retained with this parameter activated.

hydrogens, which yielded reasonable solutions for the majority of test cases. As shown in Table 4.7, the fitness of the predicted conformations improved with increasing VDW radii. However, it is worth noting that while there is a significant jump in accuracy when using radii about 0.75, larger values lead to only a marginal improvement.

Initially, the radii for hydrogens were completely eliminated, as their positions can depend greatly on the presence and type of bound ligand. However, ignoring the hydrogen VDW contributions often yielded ligand structures that were ultimately incompatible with the binding site once the hydrogens were restored. As an example, ignoring the hydrogens on a benzene ring often yielded placements that were too buried to permit restoration of the hydrogens.

It may ultimately be necessary to adjust the sizes of radii according to the nature of the atom, for example, assigning full VDW radii to set of rigid atoms like the protein backbone or proline residues and reduced radii to residues with larger configuration flexibility. The latter measure confers additional conformational freedom as the static positioning of receptor hydrogens can artificially restrict solutions to certain regions. Such a restriction may be dubious, as rotational flexibility at ambient temperature and changes in pH tend to give rise to an ensemble of possible hydrogen orientations.

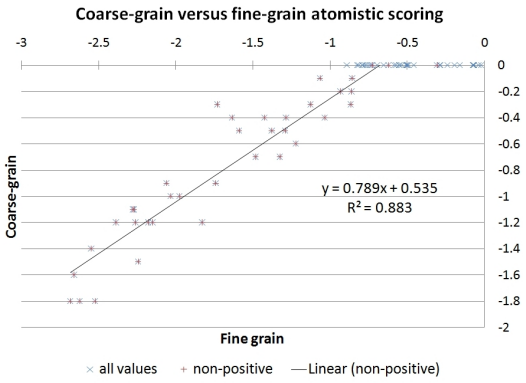



**Figure 4.18**: Comparison of fine-grain scores versus the coarse-grain representations. The linear relationship demonstrates that coarse-grain scores are a fair approximation to fine-grain energies

#### 4.4.3.3 Hydrogen bond functions

As shown in Table 4.7, including hydrogen bonds is necessary for strong performance. Moreover, the hydrogen correction term appears to offer a further improvement of the overall results. While in general diverse ensembles of ligand conformations are obtained in the absence of these terms, often these are unable to recover native hydrogen bonds despite extensive post-sampling minimization. Therefore, it is best to enforce this constraint during the growth process to ensure the proper establishment of hydrogen bond pairs. Nevertheless, one potential drawback of this approach is that by definition there is bias toward ligand-protein hydrogen bonds, thus solvent-exposed or solvent-mediated hydrogen bonds are neglected altogether.

#### 4.4.3.4 Electrostatics

The results in Table 4.7 demonstrate that disabling the Coulomb energy expression during sampling improves overall performance. The long-range persistence of charge interactions are pernicious to torsion sampling as they tend to strongly influence the position of partially charged ligand fragments. Possibly methods for overcoming this include applying a cut-off function or considering integral charges on polar atoms

only. However, neither of these have been investigated.

### 4.4.4 Overall sampling method

Traditional versus **wagSampling** refers to the manner in which the sorting, scoring and sampling approaches are combined. The **traditionalSampling** mode utilizes the fine-grain forcefield and includes scoring during the conformationSort stage. As shown Table 4.7, this method performed poorly compared to the default **wagSampling** approach. The poor behavior is likely due to two reasons. One is that a bias toward local minima is introduced by using continuous nonbond potentials during sampling. The second reason is that the search is further biased by the focus on retaining the lowest energy conformations during diversity, as opposed to seeking diverse solutions. Additionally, the use of a fine-grain FF greatly increases the computational expense of sampling. Therefore, the **wagSampling** approach is not only cheaper, but the wide, flat potential wells guide the sampling away from unfavorable pairwise interactions while imposing no additional bias.

### 4.4.5 Alanization of the receptor binding site

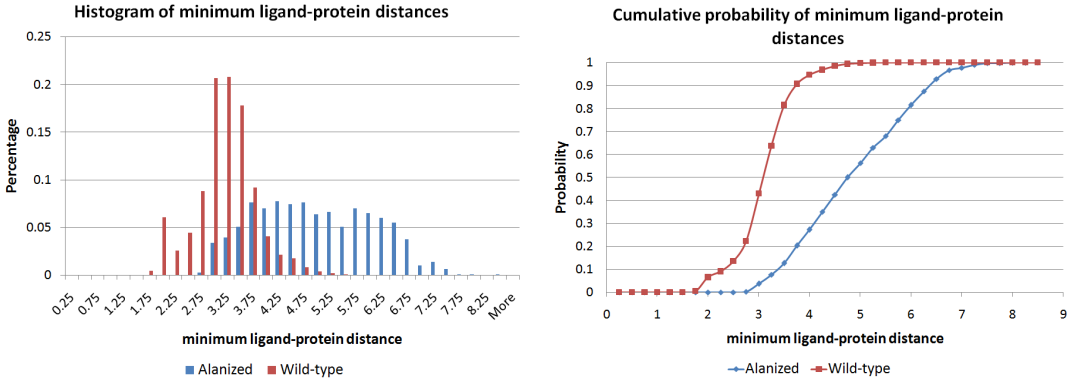
#### 4.4.5.1 Overall success

It was expected that accuracy of the predicted conformations would deteriorate in an alanized site, as the native side chains ordinarily impose spatial constraints on the ligand position. However, the results in Table 4.7 suggest that alanization did not severely impact overall prediction success rate. In some instances, like 6cpa (Fig. 4.14), it is evident that hydrophobic alanization yields a higher density of good solutions in the initial stages of the search, but this advantage quickly dissipates thereafter. In general, it is likely that at least one solution from an alanized binding site is compatible with the original LBD, provided the sampling saturates the space available. .

#### 4.4.5.2 Burial considerations

Based on the data in Table 4.7, alanization did not appear to add an immediate and consistent advantage for all cases. Initially the performance was considerably worse than for the wild-type cases, but this was because the **burialDist** was not adjusted from its default value of 4.0 Å which is inappropriate for alanized cases. To provide a more robust estimate of this parameter, the distances between each ligand atom and the closest receptor atom was plotted for both alanized and dealanized cocrystals.

The histograms for these configurations are plotted in Fig. 4.19. For the wild-type cases, the nearly-Gaussian distribution is sharply peaked around 3.25 Å, with roughly 90 percent of all distances under 3.75 Å. After alanization of all residues, the distribution is much broader. The 90th percentile for this distribution is 6.3 Å. Based on these 90th percentile values, the **burialDist** was set to 4.0 and 6.5 Å for the wild-type and alanized cases, respectively. The **burialDist** set for the alanized case is an upper bound to the distribution of minimum ligand-protein distance. For the other flavors of alanization, only subsets of residues are affected and thus the optimal burial cutoff is likely to fall somewhere between the two extremes presented above.



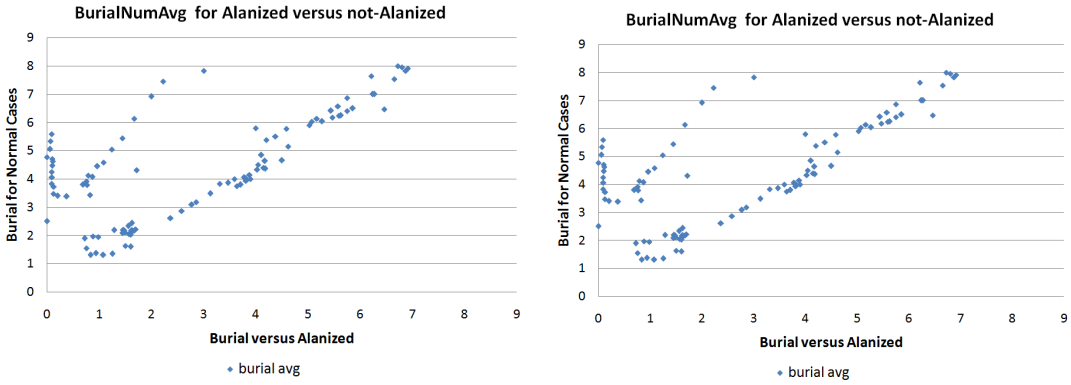

(a) (b)

**Figure 4.19**: Histogram of the minimum ligand-protein distance for all ligand atoms. Alanized binding site (red) is compared against the wild-type (blue)

Since alanization will inherently reduce the total burial of a ligand pose, the burial cutoffs used by *moleculeGL* must be relaxed accordingly. Unfortunately, initial data analysis does not reveal a simple relationship between the burial of a ligand in a wild-

type binding site and its alanized counterpart. As shown in Fig. 4.20, the cluster burial values can change considerably upon alanization. While for some clusters the burial values remain unchanged upon alanization, a sizeable percentage of cases have values that are less than half of their original percentage. Among those that change, there does not appear to be a consistent trend.

Overall, the data suggest that a burial cutoff of 0.4 would retain all clusters for the non-alanized cases, while no such cutoff is clear for alanized cases. Interestingly, strong linear scaling trends appear for the **burialAvgNum** values, which suggests there may be an exploitable feature for this term. At the present, however, the clusters contributing the groupings remain to be analyzed.




(a) (b)

**Figure 4.20**: **burialPercent** and **burialAvgNumber** of clusters in normal versus alanized cases

Instead, an estimate for **burialPercent** can be obtained by incrementing its value for the *all*, *hydrophobic*, and *wag* alanization modes. These results are summarized in Table 4.8. Based on the these data, optimal values for each of the alanization modes can be determined. Amongst the tested options, *wag* gives optimal performance amongst the possible alanization modes, as suggested in the Table 4.7. This is understandable, given that a smaller set of residues are alanized than the other alanization modes, which implies that the search domain is more constrained. In a similar fashion, a smaller **burialPercent** value tends to give superior results than a more strigent burial test.

| Alanization | **burialPercent** | | | |
|---|---|---|---|---|
| | 0.25 | 0.50 | 0.75 | 1.00 |
| all | 0.62 | 0.56 | 0.53 | 0.58 |
| hydrophobic | 0.64 | 0.65 | 0.62 | 0.58 |
| wag | 0.67 | 0.69 | 0.65 | 0.62 |

**Table 4.8**: Results for various **burialPercent** increments for the available alanization options

At first glance, these results suggest that alanization of the binding site leads to a substantial drop in the ability to predict a reliable binding pose. It is important to keep in mind, however, the sampling within the alanized site is only the first step in determining ligand poses in an modeled or otherwise inexact LBD configuration. Typically, alanized sampling is followed by a side-chain replacement step that identifies the rotamer conformations most compatible with the ligand pose ensemble. An example of this strategy applied to a modeled GPCR is provided in the following section.

For the purpose of validation testing, the native positions of the wild-type (WT) residues were simply replaced at the completion of sampling. As no attempts were made to optimize the side-chain conformations, a large number of otherwise acceptable conformations are very likely to be eliminated due to bad contacts. .

## 4.5 Prediction of F-M-R-F-$NH_2$ Bound to Mouse MrgC11

To demonstrate the ability to generate reliable ligand conformations in an alanized binding site, the neuropeptide F-M-R-F-$NH_2$ bound to mouse MrgC11 GPCR [34] was predicted (*See* Fig. 4.21). The Mas-related gene (MRG) receptors are localized to the dorsal root ganglia [47, 48] in mice and are thus believed to be implicated in pain modulation .

This case presents two interesting dilemmas. As opposed to the validation cases

for which the native positions of the LBD residues are known, the lack of a crystal structure for MrgC11 necessitates the use of modeling to recreate the LBD. While these methods have made significant advances in recent years [49, 50, 27], fine details of the receptor are likely to be non-optimal. Therefore, to expand the possible search space, sampling was performed in an alanized LBD. The resulting ligand conformations were then paired with a side-chain replacement algorithm to regenerate positions for the native residues.

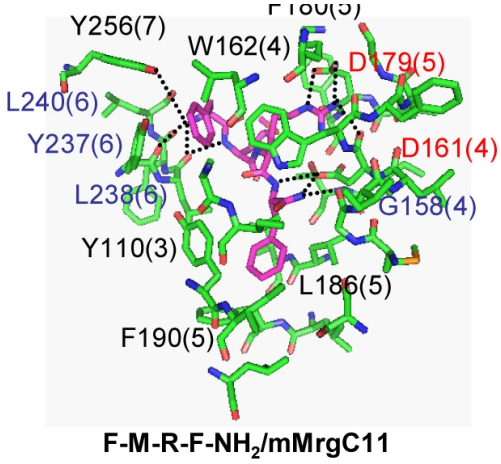



**Figure 4.21**: F-M-R-F-$NH_2$ bound to the MrgC11 receptor

The second challenge regards the unique manner in which the ligand binds. The range of pharmacological binding data suggest binding could involve two spatially disjoint regions of the protein. The efficacy of a given compound would thus be partially determined by the nature of the chemical group linking the two components. To find an adequate pose, the method must exhaustively sample along the protein surface to link the primarily and secondary binding sites. Traditional approaches that focus on scores to drive the sampling engine could be expected to fail in this regard. Since *moleculeGL* emphasizes diversity over exclusively using energies, it identified poses that could explain the peptide's pharmacology.

The original *moleculeGL* implementation identified a pose for the FRMF neuropeptide that could adequately explain experimental findings. While the details

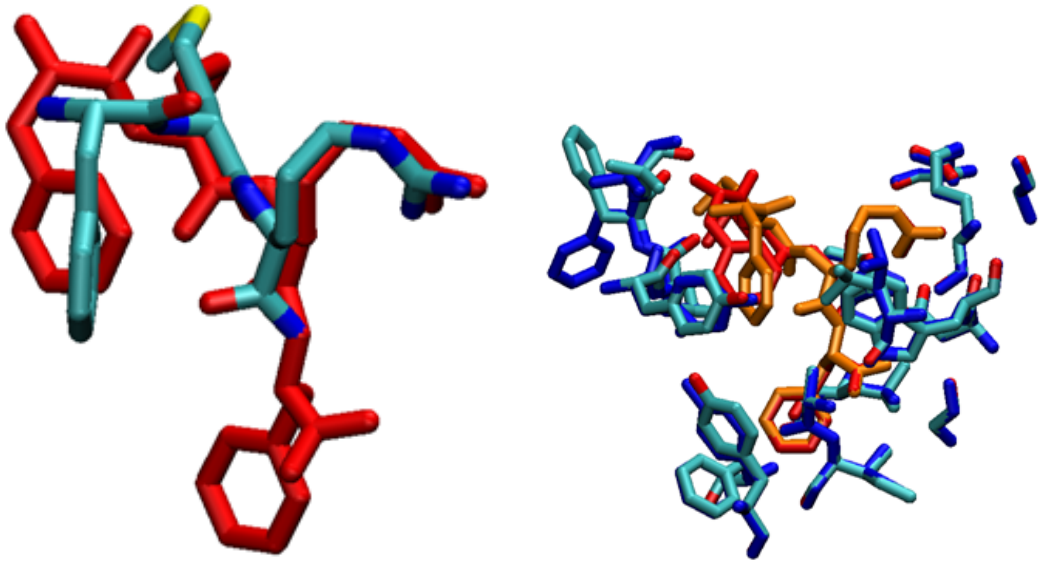
of the predicted binding mode are explored elsewhere [34], discussed here and in Table 4.9 is a trial using the current *moleculeGL* implementation with the refined structure from the Heo study as a reference.

| Alanized | | | Dealanized | | |
|---|---|---|---|---|---|
| Rank | RMSD | Energy | Rank | RMSD | Energy |
| 1 | 10.47 | -327.9 | | | 1248.46 |
| 31 | 1.62 | -257.2 | 1 | 1.28 | -235.70 |
| 23 | 2.80 | -268.4 | 2 | 2.806 | -77.01 |

**Table 4.9**: Summary of predicted FRMF neuropeptide results for alanized and dealanized MrgC11

According to the procedure outlined in Section 3.1.5 a total of 1385 conformations were returned, among which the lowest RMSD structure at 1.615 Å was ranked $31^{st}$ in energy with a score of -257.235 kcal/mol. A comparison of its conformation with the reference structure is shown in Fig. 4.22(a). The top-ranked structure had a score of -327.904, which was not unexpected given that the native LBD residues were not present.

Upon restoring the native residues via SCREAM and minimizing, the lowest-RMSD structure retained an energy of -235.74 and reduced its RMSD to 1.28 Å, with most of the error stemming from the ill-positioned N-terminal F. Meanwhile, the top contenders from the previous step were severely penalized when the native residues were imposed, which restored the lowest-RMSD structure to the top ranked by energy. The next closest contender had an energy score of -77.01 and a 2.806 Å RMSD, while the residues within the LBD had an RMSD of 2.73 Å. The finalized structure is depicted in Fig. 4.22(b).



(a) (b)

**Figure 4.22**: **(a.)** original docked conformation (cyan) lowest RMSD with heavy atoms (red) [1.61 Å; the 516th by diversity; the 30th by energy] **(b.)** original receptor conformation (cyan), after de-alanization (blue), reference ligand (range), conf 515 (red)

# Chapter 5

# Summary and Conclusions

## 5.1 Summary

*moleculeGL* is a powerful tool for rapidly searching the torsion space of a flexible ligand within a binding site. The method features a coarse-grain energy function including electrostatics and hydrogen bonds, as well as steps to ensure sampling of multiple pathways in the recursive search. This protocol is further augmented by a filtering function that groups conformations into clusters to encourage diversity in the solutions. *moleculeGL* is proficient in predicting the conformations of small ligands and, with necessary revisions, may in the future be equipped to handle small polypeptides and other highly flexible, biologically significant molecules.

## 5.2 Future

This document demonstrates the performance of the *moleculeGL* protocol and presents an assortment of performance metrics to guage the futher improvement of the methodology. A variety of topics beckon to be addressed in subsequent releases of the program. A subset of these have been listed below.

### 5.2.1 Miscellaneous

This study focused on the data set from [29]. However, state of the art packages use a much larger data set for validation, of which only an excerpt was provided in Table 1.1.

To assess the advantage of *moleculeGL* over these approaches, a much larger database of co-crystals must be added to the data set. Moreover, the obvious next step is to compare *moleculeGL* directly against available docking packages. During the algorithm development stages there was was inadequate time to benchmark this suite, but as this program reaches maturity, there will be a shift towards testing.

### 5.2.2 Energy functions

A rigorous calibration of the scoring functions employed by *moleculeGL* is also direly needed. The simple, linear formulation of the coarse-grain functions lend themselves to optimization techniques such as the simplex [17] and amoeba methods [51], which can be performed without computing derivatives. This suggests that a coarse form of minimization could be built into the *moleculeGL* protocol. Additionally, coarse-grain expressions could be fit to the appropriate fine-grain equations at a predefined set of pairwise distances. This would both improve the accuracy of the coarse-grain approaches as well as enable the use of look-up tables to accelerate computation.

### 5.2.3 Bound waters

Studies have shown that bound water molecules are commonplace in X-ray crystal structures and frequently serve catalytic roles [46]. However, their positions are not always well resolved via crystal data and oftentimes are determined by the presence of a bound ligand. Determining the possible placement of a water molecule in the course of sampling adds another level of complexity that is not pursued at this time. As such, this strategy would likely fail for those cases in which a bound atom could augment ligand binding, as was observed amongst the trypsin cases.

### 5.2.4 Pose optimization

At the present time, further refinement of the predicted structures requires calling a rigorous molecular mechanics/dynamics program like MPSim [31] for minimization. This entails considerable computational and upon minimization, many of the

structures collapse to an equivalent pose. To reduce the number of conformations submitted for refinement, a coarse-grain minimization procedure will be implemented in *moleculeGL*. This would also help reconcile bad contacts upon restoring the VDW radii to the full scale, which oftentimes presents difficulties for normal, also for optimizing hydrogen-bonding contacts, since hydrogen positions are not sampled in the search.

### 5.2.5 Grid-based scoring

Currently, the scoring engine relies on explicitly computing pairwise interactions between ligand and protein atoms. This scales as $\mathcal{O}(n^2)$, which can place severe demands on the computational resources for a typical sampling run. Implementing a grid-based interpolation scheme would be a boon to this methodology, as this would reduce the computational expense to $\mathcal{O}(n)$ per conformation in addition to the amortized cost of initializing the grid. There are challenges, nevertheless, in handling the stored data efficiently, but simulation packages have routinely addressed this hurdle.

Additionally, the pairwise approach to scoring ligands leads to long-range artifacts in the electrostatic potential. A grid-based solution would require addressing the electrostatic field in the vicinity of the protein, which involves numerical solution of the Poisson-Boltzmann (PB) equation. This can be approached in a multigrid fashion [52] to accelerate numerical convergence. Therefore, a grid-based scoring method would substantially improve simulation time and offer a considerably more accurate electrostatic description.

### 5.2.6 Neutralized protein for ameliorating long-range electrostatic artifacts

The long-range decay of the electrostatic potential can lead to spurious energetics if not handled appropriately. This becomes especially obvious when dealing with proteins, for which salt-bridges and charged amino acids are commonplace. Although there are methods for handling such problems, such as an explicit solvent simulation

or numerical solution of the PB equation, these are much too expensive for rotamer sampling. *moleculeGL* presently supports neutralized proteins for which the protonation state is revised to reduce the net charge of each amino acid to zero. This eliminates the residual charge that would otherwise introduce long-range artifacts into the simulation. At the present, the user must perform the neutralization manually via an external program [27] and load the appropriate forcefield. A future edition of *moleculeGL* may include this as an automated feature.

### 5.2.7 Coupling with anchor search

A primary goal of *moleculeGL* is to couple the torsion sampling with an anchor search program like MSCDock [46]. In this fashion, *moleculeGL* could be called for each anchor position to yield an ensemble of orientation-substituent combinations. It is expected that the majority of anchor positions will not support the full construction of substituents, thus an effective approach to thinning out these red herrings will be a necessity. Clearly the success of the tandem approach rests on the accurate placement of the lead anchor, but this should not be an insurmountable issue, given the strong performance for the docking of small ligands listed in Table 1.1.

# Bibliography

[1] B Kramer, G Metz, M Rarey, and T Lengauer. Ligand docking and screening with flexx. *MEDICINAL CHEMISTRY RESEARCH*, 9(7-8):463–478, 1999. (document), 1.1, 1.2

[2] M Rarey, B Kramer, and T Lengauer. Docking of hydrophobic ligands with interaction-based matching algorithms. *BIOINFORMATICS*, 15(3):243–250, 1999. (document), 1.1, 1.2

[3] G Jones, P Willett, RC Glen, AR Leach, and R Taylor. Development and validation of a genetic algorithm for flexible docking. *JOURNAL OF MOLECULAR BIOLOGY*, 267(3):727–748, 1997. (document), 1.1, 1.2, 3.1.1

[4] TJA Ewing, S Makino, AG Skillman, and ID Kuntz. Dock 4.0: Search strategies for automated molecular docking of flexible molecule databases. *JOURNAL OF COMPUTER-AIDED MOLECULAR DESIGN*, 15(5):411–428, 2001. (document), 1.1, 1.2

[5] N Brooijmans and ID Kuntz. Molecular recognition and docking algorithms. *ANNUAL REVIEW OF BIOPHYSICS AND BIOMOLECULAR STRUCTURE*, 32:335–373, 2003. 1.1, 1.2

[6] J Che. A simple method for improving torsion optimization of ligand molecules in receptor binding sites. *JOURNAL OF CHEMICAL THEORY AND COMPUTATION*, 1(4):634–642, 2005. ISSN 1549-9618. URL http://pubs3.acs.org/acs/journals/doilookup?in_doi=10.1021/ct0499433. 1.1

[7] ID Kuntz, JM Blaney, SJ Oatley, R Langridge, and TE Ferrin. A geometric approach to macromolecule-ligand interactions. *JOURNAL OF MOLECULAR BIOLOGY*, 161(2):269–288, 1982. 1.2

[8] B Kramer, M Rarey, and T Lengauer. Evaluation of the flexx incremental construction algorithm for protein-ligand docking. *PROTEINS STRUCTURE FUNCTION AND GENETICS*, 37(2):228–241, 1999. 1.2

[9] R Abagyan, M Totrov, and D Kuznetsov. Icm - a new method for protein modeling and design - applications to docking and structure prediction from the distorted native conformation. *JOURNAL OF COMPUTATIONAL CHEMISTRY*, 15(5):488–506, 1994. 1.2

[10] GM Morris, DS Goodsell, RS Halliday, R Huey, WE Hart, RK Belew, and AJ Olson. Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. *JOURNAL OF COMPUTATIONAL CHEMISTRY*, 19(14):1639–1662, 1998. 1.2

[11] TJA Ewing and ID Kuntz. Critical evaluation of search algorithms for automated molecular docking and database screening. *JOURNAL OF COMPUTATIONAL CHEMISTRY*, 18(9):1175–1189, 1997. 1.2

[12] G Klebe, T Mietzner, and F Weber. Different approaches toward an automatic structural alignment of drug molecules - applications to sterol mimics, thrombin and thermolysin inhibitors. *JOURNAL OF COMPUTER-AIDED MOLECULAR DESIGN*, 8(6):751–778, 1994. 1.2

[13] H Gohlke, M Hendlich, and G Klebe. Predicting binding modes, binding affinities and 'hot spots' for protein-ligand complexes using a knowledge-based scoring function. *PERSPECTIVES IN DRUG DISCOVERY AND DESIGN*, 20(1):115–144, 2000. 1.2, 3.1.1

[14] R Thomsen and MH Christensen. Moldock: A new technique for high-accuracy

molecular docking. *JOURNAL OF MEDICINAL CHEMISTRY*, 49(11):3315–3321, 2006. 0.1, 1.1

[15] HM Berman, J Westbrook, Z Feng, G Gilliland, TN Bhat, H Weissig, IN Shindyalov, and PE Bourne. The protein data bank. *NUCLEIC ACIDS RESEARCH*, 28(1):235–242, 2000. 1.3, 2.8.1

[16] SL Mayo, BD Olafson, and WA Goddard. Dreiding - a generic force-field for molecular simulations. *JOURNAL OF PHYSICAL CHEMISTRY*, 94(26):8897–8909, 1990. 2.3, 3.1.2, 3.1.4

[17] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001. 2.3.1, 5.2.2

[18] FL Markley. Unit quaternion from rotation matrix. *JOURNAL OF GUIDANCE CONTROL AND DYNAMICS*, 31(2):440–442, 2008. 2.3.3

[19] James Macqueen. K-optimal randomization tests for association in practical metric spaces using nearest neighbor methods. 1967. 2.5.1.1

[20] S Plimpton. Fast parallel algorithms for short-range molecular-dynamics. *JOURNAL OF COMPUTATIONAL PHYSICS*, 117(1):1–19, 1995. 2.6.2

[21] AD Mackerell, D Bashford, M Bellott, RL Dunbrack, JD Evanseck, MJ Field, S Fischer, J Gao, H Guo, S Ha, D Joseph-mccarthy, L Kuchnir, K Kuczera, FTK Lau, C Mattos, S Michnick, T Ngo, DT Nguyen, B Prodhom, WE Reiher, B Roux, M Schlenkrich, JC Smith, R Stote, J Straub, M Watanabe, J Wiorkiewicz-kuczera, D Yin, and M Karplus. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *JOURNAL OF PHYSICAL CHEMISTRY B*, 102(18):3586–3616, 1998. 2.6.2, 3.1.2

[22] Israelachvili J. *Intermolecular and Surface Forces*. Academic Press, Limited, 1992. 2.6.3

[23] J Zheng, R Balasundaram, SH Gehrke, GS Heffelfinger, WA Goddard, and SY Jiang. Cell multipole method for molecular simulations in bulk and confined systems. *JOURNAL OF CHEMICAL PHYSICS*, 118(12):5347–5355, 2003. 2.6.6

[24] M. P. Allen and D. J. Tildesley. *Computer simulation of liquids*. Clarendon Press, New York, NY, USA, 1989. ISBN 0-19-855645-4. 2.6.6

[25] AA Canutescu, AA Shelenkov, and RL Dunbrack. A graph-theory algorithm for rapid protein side-chain prediction. *PROTEIN SCIENCE*, 12(9):2001–2014, SEP 2003. ISSN 0961-8368. doi: {10.1110/ps.03154503}. 2.8.1

[26] Victor Kam. *METHODS IN COMPUTATIONAL PROTEIN DESIGN*. PhD thesis, California Institute of Technology, unk 2008. 2.8.1

[27] VWT Kam and WA Goddard. Flat-bottom strategy for improved accuracy in protein side-chain placements. *JOURNAL OF CHEMICAL THEORY AND COMPUTATION*, 4(12):2160–2169, 2008. 2.8.1, 3.1.5, 4.5, 5.2.6

[28] David M. Beazley, David Fletcher, Dominique Dumont, and Hewlett Packard. Perl extension building with swig. In *in O'Reilly Perl Conference 2.0*, 1998. 2.8.4

[29] MD Eldridge, CW Murray, TR Auton, GV Paolini, and RP Mee. Empirical scoring functions .1. the development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *JOURNAL OF COMPUTER-AIDED MOLECULAR DESIGN*, 11(5):425–445, 1997. 3.1.1, 5.2.1

[30] AK Rappe and WA Goddard. Charge equilibration for molecular-dynamics simulations. *JOURNAL OF PHYSICAL CHEMISTRY*, 95(8):3358–3363, 1991. 3.1.2

[31] K T Lim, S Brunett, M Iotov, R B McClurg, N Vaidehi, S Dasgupta, S Taylor, and W A Goddard. Molecular dynamics for very large systems on massively parallel computers: The mpsim program. *Journal of Computational Chemistry*, 18(4):501–521, 1997. Article WH731 J COMPUT CHEM. 3.1.2, 3.1.4, 5.2.4

[32] IV Kurinov and RW Harrison. Prediction of new serine proteinase-inhibitors. *NATURE STRUCTURAL BIOLOGY*, 1(10):735–743, 1994. 0.1, 3.1.4, 4.2.5, 4.5

[33] G Zamanakos. *A Fast and Accurate Analytical Method for the Computation of Solvent Effects in Molecular Simulations*. PhD thesis, California Institute of Technology, December 2001. 3.1.4

[34] J Heo, N Vaidehi, J Wendel, and WA Goddard. Prediction of the 3-d structure of rat mrga g protein-coupled receptor and identification of its binding site. *JOURNAL OF MOLECULAR GRAPHICS & MODELLING*, 26(4):800–812, 2007. 3.1.5, 4.5, 4.5

[35] V Kam, T Pascal, Y Liu, and WA Goddard. The dreiding generic force field for main group molecules, updated. *JOURNAL OF PHYSICAL CHEMISTRY, in preparation*, 2008. 3.1.5

[36] PA Boriack, DW Christianson, J Kingerywood, and GM Whitesides. Secondary interactions significantly removed from the sulfonamide binding pocket of carbonic-anhydrase-ii influence inhibitor binding constants. *JOURNAL OF MEDICINAL CHEMISTRY*, 38(13):2286–2291, 1995. 4.1.2.1

[37] H Brandstetter, D Turk, HW Hoeffken, D Grosse, J Sturzebecher, PD Martin, BFP Edwards, and W Bode. Refined 2.3-angstrom x-ray crystal-structure of bovine thrombin complexes formed with the benzamidine and arginine-based thrombin inhibitors napap, 4-tapap and mqpa - a starting point for improving antithrombotics. *JOURNAL OF MOLECULAR BIOLOGY*, 226(4):1085–1099, 1992. 4.1.2.1

[38] H Kim and WN Lipscomb. Crystal-structure of the complex of carboxypeptidase-a with a strongly bound phosphonate in a new crystalline form - comparison with structures of other complexes. *BIOCHEMISTRY*, 29(23):5546–5555, 1990. 4.1.2.1

[39] MA Holmes and BW Matthews. Binding of hydroxamic acid inhibitors to crystalline thermolysin suggests a pentacoordinate zinc intermediate in catalysis. *BIOCHEMISTRY*, 20(24):6912–6920, 1981. 4.1.2.1

[40] MNG James, AR Sielecki, K Hayakawa, and MH Gelb. Crystallographic analysis of transition-state mimics bound to penicillopepsin - difluorostatine-containing and difluorostatone-containing peptides. *BIOCHEMISTRY*, 31(15):3872–3886, 1992. 4.1.2.2

[41] J Eads, JC Sacchettini, A Kromminga, and JI Gordon. Escherichia-coli-derived rat intestinal fatty-acid-binding protein with bound myristate at 1.5 angstrom resolution and i-fabp(arg106-]gln) with bound oleate at 1.74 angstrom resolution. *JOURNAL OF BIOLOGICAL CHEMISTRY*, 268(35):26375–26385, 1993. 4.1.2.2

[42] JC Sacchettini, JI Gordon, and LJ Banaszak. Crystal-structure of rat intestinal fatty-acid-binding protein - refinement and analysis of the escherichia-coli-derived protein with bound palmitate. *JOURNAL OF MOLECULAR BIOLOGY*, 208(2):327–339, 1989. 4.1.2.2

[43] TS Jardetzky, JH Brown, JC Gorga, LJ Stern, RG Urban, YI Chi, C Stauffacher, JL Strominger, and DC Wiley. 3-dimensional structure of a human class-ii histocompatibility molecule complexed with superantigen. *NATURE*, 368(6473): 711–718, 1994. 4.1.2.2

[44] JT Su, X Xu, and WA Goddard. Accurate energies and structures for large water clusters using the x3lyp hybrid density functional. *JOURNAL OF PHYSICAL CHEMISTRY A*, 108(47):10518–10526, 2004. 4.2.1

[45] E Casale, C Collyer, P Ascenzi, G Balliano, P Milla, F Viola, M Fasano, E Menegatti, and M Bolognesi. Inhibition of bovine beta-trypsin, human alpha-thrombin and porcine pancreatic beta-kallikrein-b by 4',6-diamidino-2-phenylindole, 6-amidinoindole and benzamidine - a comparative thermodynamic

and x-ray structural study. *BIOPHYSICAL CHEMISTRY*, 54(1):75–81, 1995. 4.2.5

[46] AE Cho, JA Wendel, N Vaidehi, PM Kekenes-huskey, WB Floriano, PK Maiti, and WA Goddard. The mpsim-dock hierarchical docking algorithm: Application to the eight trypsin inhibitor cocrystals. *JOURNAL OF COMPUTATIONAL CHEMISTRY*, 26(1):48–71, 2005. 4.2.5, 5.2.3, 5.2.7

[47] XZ Dong, SK Han, MJ Zylka, MI Simon, and DJ Anderson. A diverse family of gpcrs expressed in specific subsets of nociceptive sensory neurons. *CELL*, 106 (5):619–632, 2001. 4.5

[48] PMC Lembo, E Grazzini, T Groblewski, MO , AND Roy, J Zhang, C Hoffert, J Cao, R Schmidt, M Pelletier, M Labarre, M Gosselin, Y Fortin, D Banville, SH Shen, P Strom, K Payza, A Dray, P Walker, and S Ahmad. Proenkephalin a gene products activate a new family of sensory neuron-specific gpcrs. *NATURE NEUROSCIENCE*, 5(3):201–209, 2002. 4.5

[49] RK Niemer, R Abrol, and WA Goddard. Computational studies of the structure and function of two lipid-activated gpcrs. *JOURNAL OF RECEPTORS AND SIGNAL TRANSDUCTION*, 28(1-2), 2008. 4.5

[50] J Heo, WW Ja, S Benzer, and WA Goddard. The predicted binding site and dynamics of peptide inhibitors to the methuselah gpcr from drosophila melanogaster. *BIOCHEMISTRY*, 47(48):12740–12749, 2008. 4.5

[51] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge (UK) and New York, 2nd edition, 1992. ISBN 0-521-43064-X. 5.2.2

[52] A Brandt and CW Cryer. Multigrid algorithms for the solution of linear complementarity-problems arising from free-boundary problems. *SIAM JOUR-*

*NAL ON SCIENTIFIC AND STATISTICAL COMPUTING*, 4(4):655–684, 1983. 5.2.5

# Appendix A

# Appendix

**Algorithm A.0.1:** RINGTEST($arguments$)

$leaf = a_ij$; is the lead of branch $a_i$
$E = \phi$
$Ring = \phi$
$while(a_{ij} != a_{i0})$
$\begin{cases} U = a_{ij} \\ V = a_{ij} - 1 \\ E' = E' \cup e_{uv} \\ \textbf{if } a_{ij}leaf \\ \quad \textbf{then } R = E' \\ \textbf{success} \end{cases}$
**return** ($vertices[]$)

**Algorithm 1**: Algorithm for identifying rings from a list of bonds

**Algorithm A.0.2:** CLUSTEREDGES($arguments$)

$T = MinimumSpanningTree()$;
**for** $V_{end} \leftarrow 0$ **to** $|V_{end}|$
**do**
$R_i = RingTest(V_{end})$;
$R = R \cup R_i$
**for** $V \leftarrow 0$ **to** $n$
**do** ;
**if** $E \in R$
**then** $E = n_{cluster}$

**Algorithm 2**: Algorithm for clustering a ligand into a set of rotatable elements. The first step generates a minimum spanning tree in $\mathcal{O}(E\,lg\,V)$ time. The second step identifies rings for each MST branch in $\mathcal{O}(E\,lg\,E)$ time. The final step assigns cluster numbers

1. let $C = \cup_{k=n}^{N} clusters$
2. find $R, \mathbf{t}$ s.t. $[\mathbf{x}'_f, \mathbf{x}_f, \mathbf{x}_r]_{ref} = [R|\mathbf{t}][\mathbf{x}'_f, \mathbf{x}_f, \mathbf{x}_r]_{conf}$
3. for $[\mathbf{x}'_f, \mathbf{x}_f, \mathbf{x}_r, \mathbf{x}'_r]_{conf}$, compute $\theta_{torsion}$
4. compute $R(\theta, \mu)$
5. Compute $\mathbf{x}' = R(\theta) \cdot \mathbf{x} \quad \forall x \in C$

**Algorithm 3**: Algorithm for rotating the bond between clusters $(n-1)$ and $n$ about an arbitrary axis

Step 1. Build bonds, $B$

1. define $E$ and $V$
2. $B \equiv E$

Step 2. Build torsions, $T$

1. Find set $V' \subset V$, which has $degree(v) \geq 2$
2. Find set $E' \subset E$, which contains $e'_{ij} : v'_i, v'_j \in V'$
3. Find set $T$
   - For each $e'_{ij}$, find all $e_{ki}, e_{jl} \in E$
   - $T = \{t | t = \{e_{ki}, e'_{ij}, e_{jl}\}\}$

Step 3. Build angles, $A$

1. For each $t_{kijl} \in T$, $A_l = \{a | a = \{e_{ki}, e'_{ij}\}\}$, $A_r = \{a | a = \{e'_{ij}, e_{jl}\}\}$
2. A=$A_l \bigcup A_r$

**Algorithm 4**: Protocol for determining all bonds, angles and torsions from an atom connectivity list.

1. Define branches
   - find all primary clusters (heavy atoms connected to only one other heavy atom)
   - remove clusters upfield from lead cluster
   - create array of upfield paths back to lead cluster
2. Rank paths according to length to find primary chain, $P$
3. Rank remaining paths according to one of the following to define secondary chains, $B$
   - shortest chain to second longest chain
   - longest chain to shortest
   - chain closest to tip, then 2nd closest and so forth

**Algorithm 5**: Protocol for defining the sampling path, which is the order in which the ligand clusters are sampled. The longest chain is defined as the primary branch, $P$, while all other branches, $B$, form the set of secondary branches

Sample:

1. Sample primary chain, $P$, to find ensemble $\{P\}$
2. Repeat the following for all $B_i \in B$
   - Disable all clusters not part of chain $B_i$
   - Sample $B_i$ and retain best $n$ solutions to form $\{B_i\}$
   - Sort $B_i$: $\hat{B}_i = \mathsf{sort}(B_i)$
3. Select best combinations based on total energy
   - Use best solution from each $B_i$ to form ground state solution: $C_0 \equiv P^0 \bigcup_{m=0}^{M} B_m^0$
   - Repeat the following until desired number of combinations is reached or all $B_i$ are exhausted
     - Draw next lowest member of $\hat{B}$: $\hat{b}_m^i \in \hat{B}$
     - Remove the corresponding branch from the last combination: $C_{j+1} = C_j \bigcap b_m$
     - add new branch to form new conformation: $C_{j+1} \leftarrow \hat{b}_m^i$

**Algorithm 6**: Protocol describing the generation and combination of branch ensembles to form completely constructed ligand solutions

**Algorithm A.0.3:** MERGE($arguments$)

$vertices[]$
//n number of conformations
**for** $i \leftarrow 1$ **to** $N$
**do** $\begin{cases} closestDist[i] = min(d_{ij}) \forall j \in N \\ closestMember[i] = j \\ weights[i] = 1 \end{cases}$
$Sort(closestDist[])$
**for** $i \leftarrow N$ **to** $1$
**do** $\begin{cases} \textbf{if } weight[i] \rangle weight[j] \\ \quad \textbf{then} \begin{cases} weight[i] += weight[j] \\ vertices[j] \leftarrow 0 \\ weight[j] \leftarrow 0 \end{cases} \\ \quad \textbf{else if } weight[i] \langle weight[j] \\ \quad \textbf{then} \begin{cases} weight[j] += weight[i] \\ vertices[i] \leftarrow 0 \\ weight[i] \leftarrow 0 \end{cases} \\ \quad \textbf{else} \\ \quad \textbf{then} \begin{cases} \textbf{if } rand() > 0.5 \\ \quad \textbf{then} \begin{cases} x \leftarrow j \\ y \leftarrow i \end{cases} \\ \quad \textbf{else} \\ \quad \textbf{then} \begin{cases} x \leftarrow i \\ y \leftarrow j \end{cases} \\ weight[x] += weight[y] \\ vertices[y] \leftarrow 0 \\ weight[y] \leftarrow 0 \end{cases} \end{cases}$
**return** ($something[]$)

**Algorithm 7**: Process for iteratively consolidating a set of conformations into diverse clusters according to the hierarchical clustering protocol

**Parameter listing**

| Parameters | |
|---|---|
| SetBondRotamers | SetHBFilter |
| SetFixedBond | SetStrainFilter |
| SetMoleculeConformation | SetSelfClashFilter |
| SetMoleculeColorRGB | SetBurialCumulative |
| SetAtomColorRGB | SetBurialFilter |
| SetMoleculeColor | SetBurialPercent |
| SetAlanizationMode | SetSortingMode |
| SetDiversityVariable_useBurialWeight | SetIntermedChildren |
| SetParameterFile | SetMCMaxSteps |
| SetDiversityCutoff | SetMCAcceptanceMode |
| SetNumFocusRotamers | SetMCTemp |
| SetVDWMode | SetFinalFamilies |
| SetVDWRadiiScale | SetFinalChildren |
| SetDiversityMode | SetRMSDComparison |
| SetSortByParents | SetMoleculeName |
| SetHVDWRadiiScale | SetICConfRotatedCluster |
| SetNonbondCutoff | SetICConfAngle |
| SetFineScoring | SetAtomResNum |
| SetFocusSampling | SetAtomResName |
| SetSamplingMode | SetAtomAtomName |
| SetSelfInteractions | SetAtomCoords |
| SetRecursionDepth | SetDiversityVariable_usePrimaryWeights |
| SetIntermedFamilies | SetDiversityVariable_useSecondaryWeights |
| SetNumRotamers | SetDiversityVariable_useCumulativeWeights |
| SetNumExploreRotamers | SetOpenGLFocus |
| SetHBMode | SetOpenGLFocusPoint |
| SetHBondCutoff | SetDisplaySize |
| SetBurialDistance | SetBGColor |

**Table A.1**: Parameter options