To appear in the Journal of Computional Biology

Maintaining and Enhancing Diversity of Sampled Protein Conformations in Robotics-Inspired Methods

Jayvee R. Abella, Mark Moll, and Lydia E. Kavraki*

Department of Computer Science, Rice University, Houston, TX 77005, USA

Abstract

The ability to efficiently sample structurally *diverse* protein conformations allows one to gain a high-level view of a protein's energy landscape. Algorithms from robot motion planning have been used for conformational sampling and several of these algorithms promote diversity by keeping track of "coverage" in conformational space based on the local sampling density. However, large proteins present special challenges. In particular, larger systems require running many concurrent instances of these algorithms, but these algorithms can quickly become memory intensive because they typically keep previously sampled conformations in memory to maintain coverage estimates. Additionally, robotics-inspired algorithms depend on defining useful perturbation strategies for exploring the conformational space, which is a difficult task for large proteins because such systems are typically more constrained and exhibit complex motions. In this paper, we introduce two methodologies for maintaining and enhancing diversity in robotics-inspired conformational sampling. The first method addresses algorithms based on coverage estimates and leverages the use of a low-dimensional projection to define a global coverage grid that maintains coverage across concurrent runs of sampling. The second method is an automatic definition of a perturbation strategy through readily available flexibility information derived from B-factors, secondary structure, and rigidity analysis. Our results show a significant increase in the diversity of the conformations sampled for proteins consisting of up to 500 residues when applied to a specific robotics-inspired algorithm for conformational sampling. The methodologies presented in this paper may be vital components for the scalability of robotics-inspired approaches.

Keywords: protein conformational sampling, robotics-inspired sampling, perturbation strategies, concurrent sampling.

1 Introduction

The function of a protein is related to its three-dimensional structure and its associated structural changes (Wei *et al.*, 2016). Detailed understanding of protein function and how diseases disrupt function can eventually lead to treatment or prevention (Carlson, 2002). One typical starting point is to probe a given protein's conformational space. This is done through experimental techniques, such as X-ray crystallography, cryo-electron microscopy, or nuclear magnetic resonance (Powell, 2016; Xu *et al.*, 2015; Marion, 2013), which can provide necessary structural information that is refined by (or used as constraints for) computational techniques for conformational sampling, such as molecular dynamics (Paquet and Viktor, 2015). Conformational sampling can provide high-resolution information about a protein's conformational space (Maximova *et al.*, 2016). However, molecular dynamics methods have difficulty with rapid *exploration*

*Corresponding author. Prof. Lydia Kavraki, Dept. of Computer Science, MS 132, Rice University, 6100 Main Street, Houston, TX 77005, USA. E-mail: kavraki@rice.edu

of conformational space. Typical timescales for major biomolecular events are normally on the order of microseconds or greater while the timesteps of these methods are on the order of femtoseconds. *Enhanced* sampling methods can provide an initial exploration of conformational space that can be used to bootstrap more detailed molecular dynamics simulations.

Algorithms from robot motion planning (Gipson *et al.*, 2012; Al-Bluwi *et al.*, 2012) constitute one such class of methods for enhanced exploration of the conformational space. For a great introduction to these kinds of methods, we refer the interested reader to (Shehu and Plaku, 2016). Robotics-inspired methods have been used for conformational sampling in a variety of cases, including protein folding (Amato *et al.*, 2003; Thomas *et al.*, 2007; Tapia *et al.*, 2010), loop sampling (Yao *et al.*, 2008; Shehu and Kavraki, 2012; Stein and Kortemme, 2013), identifying low-energy transitions between known conformations (Raveh *et al.*, 2009; Haspel *et al.*, 2010; Al-Bluwi *et al.*, 2013; Gipson *et al.*, 2013), exploring conformational space (Jaillet *et al.*, 2011; Shehu and Olson, 2010; Gipson *et al.*, 2013; Luo and Haspel, 2013), and improving fit to experimental data (Devaurs *et al.*, 2016, 2017). These methods are characterized by their *geometric reasoning* to bias the exploration to unexplored regions of the space. Thus, the focus of these methods is on generating a *diverse* set of conformations.

Robotics-inspired sampling algorithms can be characterized by how they make two critical decisions. First, each algorithm must make a decision on *where* to sample in conformational space. These algorithms often estimate *coverage* based on the local sampling density to bias exploration towards less densely sampled regions of the conformational space. This is mainly how several robotics-inspired approaches promote structural diversity in their exploration. Second, once a selected region has been chosen, each algorithm must make a decision on *how* to generate a new conformation. Usually the proposed conformation is generated by perturbing a previously sampled conformation from the selected region and determining if the proposed conformation is valid (typically by checking the energy or for steric clashes). In this work, we will use the term *perturbation strategy* to refer to how the algorithm generates new conformations from previously sampled ones. So starting from an initial conformation, a robotics-inspired algorithm iterates over deciding where to sample, often using coverage estimates (first decision), then generating a new conformation based on its perturbation strategy (second decision).

While robotics-inspired approaches have seen many initial successes, larger proteins present special challenges that complicate how these algorithms make the two critical decisions and hence, hinder their ability to efficiently generate diverse conformations. First, larger proteins have more residues, or degrees of freedom, so sampling for such systems takes more time and memory and often requires running these algorithms concurrently across multiple cores. However, running many concurrent instances of robotics-inspired algorithms becomes memory intensive because of their frequent storage and use of previously sampled conformations as described by the first decision. In other words, robotics-inspired methods must be run for a shorter time as the size of the protein increases. As more concurrent instances are used, the memory usage rate increases since each instance needs to access all of the previous samples. The naive solution to this memory issue is to write the conformations to disk and restart conformational sampling at a randomly selected conformation, but we will demonstrate that this compromises the coverage estimates to the point where these algorithms lose the ability to promote diversity. So in addition to the memory issue, there is also the problem of *coordination* across the concurrent runs of sampling.

Second, defining useful perturbation strategies becomes complicated for larger proteins because such systems are typically more constrained and exhibit complex motions. Larger proteins usually have high correlation among distant residues that results in more intricate movements. Perturbation strategies are then less likely to capture these movements, and the probability of

proposing a valid conformation diminishes greatly (Vitalis and Pappu, 2009). Therefore, the ability to generate conformations highly different from the starting or reference conformation becomes challenging. This has led to algorithms that aim to somehow encode flexibility information into the perturbations (Shehu and Plaku, 2016). NMA-RRT samples conformations using normal mode analysis (Al-Bluwi *et al.*, 2013). KGS uses sampling in the nullspace of the Jacobian of constraints to generate conformations that satisfy constraints introduced by hydrogen bonds (Pachov and van den Bedem, 2015). PCA-EA uses perturbations in a principal component space defined from wildtype and mutant structures, which are then translated to the original conformational space using a combination of reconstruction algorithms (Clausen and Shehu, 2015). Note that defining perturbation strategies is also a problem outside of robotics-inspired methods as PCA-EA is an evolutionary algorithm. Our approach instead makes use of informed *moves*, which may be less computationally expensive because it does not require the use of reconstruction algorithms, computing Hessians (for NMA-RRT), or computing Jacobians.

In this paper, we introduce two methodologies for maintaining and enhancing diversity in robotics-inspired conformational sampling. The first method addresses algorithms based on coverage estimates and leverages the use of a low-dimensional projection to define a global coverage grid that maintains coverage across concurrent runs of sampling. This global coverage grid keeps statistics about previously generated samples across different runs, which means that each run no longer needs to access *all* of the conformations. This approach solves not only the memory issue associated with robotics-inspired sampling methods that rely on coverage estimates, but also the coordination problem of sampling across multiple cores. Our approach allows robotics-inspired methods to *maintain* the ability to efficiently decide *where* to sample conformations. The second method is an automatic definition of a perturbation strategy derived from B-factors, secondary structure, and rigidity analysis. We also use the B-factor information to define the low-dimensional projection and compare to prior work on defining projections (Novinskaya *et al.*, 2017). This method *enhances* diversity by focusing *how* our algorithm perturbs conformations using readily available flexibility information. Our results show that our methodology leads to a significant increase in the diversity of the conformations generated as well as the number of conformations generated for proteins consisting of up to 500 residues when applied to a specific robotics-inspired algorithm for conformational sampling, the Structured Intuitive Move Selector (SIMS) framework for conformational sampling (Gipson *et al.*, 2013).

The rest of the paper is organized as follows. In the next section, we will describe our methodologies in detail, which are implemented into SIMS. In section 3, we show how our new methodologies result in a significant improvement in the structural diversity of the sampled conformations. We also provide a discussion of the relative importance of each methodology. Finally, we conclude with a brief summary and a discussion of directions for future work.

2 Methods

2.1 Structured Intuitive Move Selector (SIMS)

We apply our methods within the Structured Intuitive Move Selector (SIMS) framework for conformational sampling (Gipson *et al.*, 2013), which exemplifies the operation of several robotics-inspired methods through the combined use of the Open Motion Planning Library (OMPL) (Şucan *et al.*, 2012) along with Rosetta (Leaver-Fay *et al.*, 2011). In this section, we provide a high-level overview of SIMS that will introduce the components needed to describe our new methodologies. A detailed description of SIMS can be found in (Gipson *et al.*, 2013).

SIMS makes use of OMPL to determine *where* to direct exploration through the use of coverage

estimates, which are encoded inside a data structure called the *coverage grid*. The coverage grid data structure relies on a *projection* as input, which is used to map conformations to the coverage grid. The coverage grid consists of cells from the discretization of the mapped space, and each cell contains pointers to the conformations that are mapped to it. Deciding where to direct exploration at any given iteration is a two step process where 1) a cell is chosen and 2) a conformation is chosen within the cell. The probability a cell is chosen is proportional to its computed *importance* value, which is a function of various *grid cell statistics* such as the number of conformations mapped to it. Essentially, cells that are less densely populated are chosen more often than cells that are more densely populated. Finally, a conformation is randomly chosen from the cell. More details are provided on the coverage grid in Section 2.2 and on the projection in Section 2.3.

SIMS determines *how* new conformations are generated through a so-called *schema*. The schema encodes how to repeatedly apply small perturbations called *moves* to previously generated conformations. SIMS uses an internal coordinate representation, where only dihedral angles are manipulated (bond angles and bond lengths are kept constant). SIMS' perturbation strategy, defined in the *schema*, specifies what type of moves to use, how they are applied, and how often to apply them. The moves are applied to sets of residues called *fragments*, where each move-fragment pair is assigned a *weight* to reflect how often to apply the move-fragment pair. The probability that a particular move-fragment pair is chosen at any given iteration is proportional to the weight. Ideally, the schema captures which fragments of residues might be involved in coordinated motion and how flexible they are (through the use of the weights). Previous work showed that secondary structure can be used to partition the protein into flexible loops, which should be perturbed more often (given higher weight) than relatively rigid helices and sheets (given lower weight) (Gipson *et al.*, 2013). For each type of fragment, different moves can be defined (e.g., loop sampling, random perturbation, energy minimization). SIMS makes use of Rosetta for implementations of the moves (Leaver-Fay *et al.*, 2011). Once the move is applied to a fragment, side chain positions are determined by Rosetta's side chain minimization protocol (Das and Baker, 2008). Since SIMS uses Rosetta for the move implementations, SIMS' perturbation strategy can be easily extended to include advances in Rosetta's own robotics-inspired approaches to sample new conformations such as in (Stein and Kortemme, 2013). More details are provided in Section 2.3 on how a perturbation strategy in SIMS is constructed.

Each proposed conformation is automatically rejected if the computed energy of the proposed conformation is above a user-defined threshold. In this work, we use Rosetta's "score12_full" all-atom energy function for our smaller-sized proteins and Rosetta's "score3" energy function in "centroid" mode for our larger-sized proteins, although other energy functions could be used as well. Centroid mode computations in Rosetta are faster because side chains are approximated as a single atom of varying size, which provides additional computational benefit for larger proteins while still maintaining molecular detail. Energy thresholds are chosen to filter out conformations with steric clashes and other highly unfavorable interactions. Energy thresholds for this work are set to the value 0 because past experiments tend to show that conformations with a positive Rosetta score have some degree of steric clashes. One could always lower the energy threshold or filter out high-energy conformations in a post-processing step to obtain sampled conformations with lower energy.

2.2 Global Coverage Grid

The first critical decision that robotics-inspired methods have to make is *where* to direct the exploration. Many robotics-inspired methods, such as SIMS, base this decision on the computed

coverage estimates. Coverage estimates measure where the less-densely sampled regions of the conformational space are located. Based on the coverage estimates, robotics-inspired approaches incorporate a bias towards the unexplored regions of the space (Shehu and Plaku, 2016). Computing coverage estimates in robotics-inspired methods becomes complicated for larger systems, so in this section we describe a method that can maintain their ability to compute coverage.

Conformational sampling generally becomes a harder problem as the size of the considered system increases. Unless the system is highly stable and only exhibits small-scale movements (like side-chain rearrangements), more computational resources are needed. Energy computation takes longer so we must run simulations longer in order to sample a given number of conformations. The conformational space is also larger so we may need more conformations to accurately characterize the space. These complications give rise to the need to run multiple robotics-inspired sampling across many cores. However, keeping all of the sampled conformations in memory means that the rate of memory usage increases. For the rest of this paper, we will refer to this as the "memory issue." While there has been work on parallelizing robotics-inspired sampling techniques (Devaurs *et al.*, 2013; Ichnowski and Alterovitz, 2012; Plaku *et al.*, 2005), these approaches do not address the fact that memory-use becomes a bottleneck for large proteins. So in this work, we are addressing the problem of running multiple instances of SIMS concurrently in an efficient manner that can also handle the memory issue.

Initially, one may consider keeping all of the sampled conformations on disk and running separate, concurrent instances of SIMS across each computing core. But this means each core must have access to all of the sampled conformations because any previously sampled conformation could be perturbed in a given iteration. This may be expensive if every iteration involves reading and writing to disk. We could also write all the conformations to disk periodically and restart the exploration from randomly chosen points. However, as mentioned in Section 1, this results in losing the vital coverage estimates. After a restart, the algorithm is likely to re-explore parts of conformational space that have been densely sampled in previous runs.

Keeping all of the sampled conformations on disk appears to be unavoidable when running multiple instances of SIMS for a long time. So in order to prevent excessive reading and writing to disk, each instance of SIMS must work with its locally generated set of conformations. The question becomes how each instance of SIMS can "get informed" of the work other instances of SIMS are performing. We solve this by leveraging SIMS' use of a low-dimensional projection to keep coverage estimates and implementing a *global* coverage grid, whose scope reaches across all the instances of SIMS.

In SIMS, sampled conformations are added to the coverage grid data structure through the use of a projection (detailed in Section 2.3). The grid contains cells with conformations mapped to them and by counting how many conformations map into each cell, we can estimate the sampling density or coverage. Different robotics-inspired techniques such as Expansive Space Trees (Hsu *et al.*, 1999) and Kinodynamic Motion Planning by Interior-Exterior Cell Exploration (KPIECE) (Sucan and Kavraki, 2009) use this information to guide the sampling towards less-densely sampled parts of the conformational space. In this work we use KPIECE since it has been shown to significantly outperform EST (Sucan and Kavraki, 2009).

KPIECE keeps track of various statistics for each grid cell and uses these statistics to compute a heuristic called *importance* for each cell. Conformations in cells with higher importance are perturbed more often. Importance is computed for each cell using four statistics:

(1) The number of projected conformations mapped into the cell.
(2) The number of times the cell has been chosen for expansion.
(3) The iteration in which the cell had its first conformation mapped to it.
(4) The number of cell neighbors that have conformations mapped to it.

An increase in (1), (2), and (4) produce lower importance while a high value for (3) produces greater importance. Indeed, these are the statistics that are lost when conformations must be written to disk. Fewer conformations are accounted for in the coverage grid, and the importance heuristic loses the ability to differentiate cells based on sampling density.

Our method saves *global* grid cell statistics into a central database along with conformations sampled from each SIMS instance. In this work, we used a MySQL database to handle the read/write requests from the multiple SIMS instances, and created tables to simply hold the conformations and global coverage grid. When a SIMS instance is started, a subset of conformations is loaded along with "summarized" coverage statistics about the global coverage grid. These "summarized" coverage statistics are an indirect way of accounting for the sampling done by other SIMS instances. The SIMS instance then proceeds for a specified amount of time. The SIMS instance maintains its own *local* coverage grid (using the conformations generated by the run) enriched with the summarized coverage statistics (taken from the global coverage grid). When a SIMS instance is ready to write conformations, the new conformations are written to the database, and the global grid cell statistics are then updated. Thus, our method provides the coordination across the SIMS instances that effectively maintains coverage estimates.

Grid cell statistics on the global coverage grid are maintained centrally in a similar manner to how an individual SIMS instance computes grid cell statistics on a local coverage grid. Each grid cell computes an importance heuristic that determines how often conformations from that cell are perturbed. In the context of the global coverage grid, (3) is no longer used to compute importance because there is no meaningful way to define iteration when multiple cores are sampling simultaneously. However, (1), (2), and (4) are still used.

When a SIMS instance is finished, the new conformations are written to the database and the global grid cell statistics are updated. This is done by computing the change in (1) and (2) during the course of the SIMS run. These values are then added to the global values of (1) and (2) in the database. (4) is subsequently updated based on the new grid cell statistics. These statistics are global in the sense that all the instances provide an update when their run is finished.

When a SIMS instance is restarted, the KPIECE sampling strategy is used to select new starting conformations based on the global coverage grid. Additionally, the local coverage grid is initialized to the current values in the global coverage grid. While each core is aware of the global coverage statistics, each core can only perturb conformations that are in memory (i.e., generated since the start of the SIMS instance). However, when a conformation is generated in a cell, the sampling density is determined not only by the conformations generated from the SIMS instance but also the sampling density loaded at the start of the run (Fig. 1). The presence of all other sampled conformations is thus accounted for indirectly.

We now claim that this algorithm also solves the memory issue. Each core will cycle through three steps: reading conformations and statistics from the database, running a SIMS instance, then writing conformations and updating the global statistics. The frequency in which each core does this process is a user-defined parameter called the *restart frequency*. The memory issue is avoided through the use of the summarized coverage and a restart frequency that is not too low. That is, we must restart often enough such that a core will not run out of memory from sampling too many conformations. Interestingly, there is incentive to restart often as this is the mechanism in which the database is updated with the work that each SIMS instance has done. On the other hand, saving conformations to a central database and restarting has some computational overhead associated with it. Our experiments use a restart frequency of 6 per hour (restart every 10 minutes). We leave it as future work to determine an optimal value for this parameter.

Figure 1: Pictorial representation of the effect that the use of summarized coverage statistics has on a local coverage grid. Each blue dot represents a previously sampled conformation. Before hitting memory limits, the exploration is proceeding in some direction depicted by the black arrow. When the exploration restarts, summary coverage statistics are maintained and depicted in shades of gray. Darker shades indicate more densely sampled areas that the algorithm can use to reduce sampling redundancy.

Finally we note that when an instance syncs with the database, then the information in the coverage grid could not be as up-to-date as possible. There could be other instances that have explored parts of conformational space that the coverage grid is not yet aware of. Our results in Section 3 indicate that this is not a cause for concern. Simply having some notion of the work done by other instances is enough to see an improvement in the diversity of conformations generated using our methodologies. The reason for this is because the original KPIECE method (Sucan and Kavraki, 2009) is inherently adaptive. If at some point of execution multiple cores are working in the same region of the conformational space, the global coverage grid will eventually get 'informed' of the work when the cores synchronize with the database. Then when new SIMS instances are created, these instances are less likely by design to work in the same region again because of the importance values in the coverage grid.

2.3 Defining a Perturbation Strategy using Flexibility Information

The second critical decision that robotics-inspired methods have to make is *how* to generate new conformations. In other words, these methods have to define a perturbation strategy to obtain new conformations from previously sampled ones. Defining perturbation strategies for larger proteins is more difficult because of the high correlation among residues, and naive/uninformed strategies will result in high rejection rates. In this section, we will describe how to generate informed perturbation strategies through readily available flexibility information. In the context of SIMS, this translates to finding a definition of the *schema*. Note that while we focus the construction of the new perturbation strategy on the schema used in SIMS, the same ideas can also be incorporated into other conformational sampling frameworks. We show how to automatically generate a schema that biases perturbations towards fragments that are more flexible using a combination of B-factors, secondary structure, and rigidity analysis.

The new perturbation strategy incorporates *global* structure information into the schema. In previous iterations of SIMS, the default schema made use of only secondary structure information. Alpha helices and beta sheets were made more stable than loops. However, secondary structure is essentially *local* information because every secondary structure element consists of a few consecutive residues. While in general helices and sheets are more stable than loops, helices

and sheets from different parts of the protein may have vastly different flexibility (Novinskaya *et al.*, 2017). The old default schema essentially lacked *tertiary* structure information describing *global* flexibility.

While secondary structure is readily computed or available in the PDB, tertiary structure information is not available directly from experiment. However, an approximation can be derived from the atom coordinates using rigidity analysis. This is done with KINARI web, a suite of tools for computing rigidity and flexibility of biomolecules (Fox *et al.*, 2011). KINARI web uses the pebble game algorithm to compute clusters of residues that are expected to move together (Lee and Streinu, 2008). A PDB file is inputted into the web server to get the residue clusters. All default parameters are used in the computation. KINARI outputs a file that specifies residue clusters. Each cluster consists of a set of residue intervals. We use *each* residue interval from each cluster as a separate residue grouping. These residue groupings will be used by the schema to model parts of the protein that are supposed to move together. We could have chosen to instead use the cluster of residues (or a set of intervals) as the residue groupings for a simpler rigidity model, but KINARI may only detect one or two residue clusters. Our definition of the residue groupings allows for a more fine-grained decomposition of the protein.

The schema used in SIMS specifies *fragments*, which consist of groups of residues, and *moves*, which define perturbations on the fragments. Each fragment is assigned a *weight* which describes how frequently the fragment is chosen to be perturbed. SIMS currently has 5 major moves that are briefly defined as follows:

1. *Minimization* involves a few steps of an energy minimization protocol on the fragment. We use the "dfpmin_armijo_nonmonotone" protocol [1] and run until a tolerance of 0.01.
2. *Loop sampling* involves sampling a random loop conformation with the constraint that the endpoints remain in the same position.
3. *Rigid-body sampling* involves rotating and translating one part of a domain relative to another. This is done by a displacement of one loop endpoint relative to the other.
4. *Random single perturbation* involves randomly perturbing a single residue's dihedral angles within a given fragment.
5. *Randomize all* involves perturbing all the dihedral angles in a given fragment.

In addition to the rigidity analysis, which has been used before in robotics-inspired sampling (Thomas *et al.*, 2007; Luo and Haspel, 2013; Andersson *et al.*, 2016), our method assigns a weight to each fragment using B-factors. SIMS relies on a starting conformation which is derived from experiment. These experiments will have some measure of uncertainty, which is usually correlated with flexibility or movement. For X-ray crystallography experiments, B-factors (also known as temperature factors) describe the displacement of the atomic positions from their mean values (Trueblood *et al.*, 1996). These B-factors can be easily extracted from a PDB file (coordinates of a protein conformation derived from experiment) to generate the projection. B-factors can also be generated from prediction tools (Yuan *et al.*, 2005). As a reminder, the probability a move-fragment pair is chosen is proportional to its assigned weight. Thus, using B-factors as the weights naturally biases the fragments that are more flexible.

For a system with n residues, n B-factors corresponding to the alpha carbon atoms in the backbone are extracted from the PDB file. Then for each factor b_i, $1 \leq i \leq n$, a user-defined

[1] https://www.rosettacommons.org/docs/latest/rosetta_basics/structural_concepts/minimization-overview

range $[b_{low}, b_{high}]$ of the B-factors is imposed using the following transformation.

$$t_i(b_i) = \begin{cases} b_{low}, & \text{if } b_i < b_{low} \\ b_{high}, & \text{if } b_i > b_{high} \\ b_i, & \text{otherwise} \end{cases} \quad (1)$$

This transformation is done because we are only interested in the relative flexibility of the fragments to each other. B-factor data may contain noisy values for fragments that may overly dominate and cause the method to over sample this region. For our experiments, we normally set $b_{low} = 20$ and $b_{high} = 50$. Next, the following transformation is applied to each t_i:

$$f(t_i) = \exp(t_i/\alpha) \quad (2)$$

This transformation essentially spreads the values farther apart from one another. The amount of spreading can be controlled using another user-defined parameter α. The use of an exponential function here is to space the larger B-factors away even further from the smaller B-factors such that the flexible residues are sampled more often. All of our experiments use $\alpha = 10$. The weight of each fragment is then computed by summing each $f(t_i)$ in the fragment.

Using KINARI, secondary structure information, and B-factors, the schema can be automatically generated. We propose to generate a schema that consists of three major classes of fragments. The first is a class containing only a single fragment that is defined over the whole protein. This class is sampled 10% of the time and is used to occasionally generate structures with a lower energy (*minimization*) or try disruptive whole protein perturbations (*rigid-body sampling*). When this fragment is sampled, *minimization* is chosen 90% of the time and *rigid-body sampling* is chosen 10% of the time.

The second major class of fragments is generated using secondary structures. This class of fragments constituted the majority of the fragments in previous iterations of SIMS. We place less overall influence on this class since we only sample this set 40% of the time (compared to 90% previously). The secondary structure information is extracted from the PDB file. Alpha helices and beta sheets are treated as loops if they are less than 5 residues in length. A fragment is then defined for each consecutive interval of residues with the same secondary structure classification. Loops can be perturbed using *loop sampling* (10%), *random single perturbation* (30%), *randomize all* (30%), or *rigid-body sampling* (30%). Helices and sheets are generally more stable so we use *random single perturbation* (50%) or *rigid-body sampling* (50%). In previous work, SIMS manually defined loops to be sampled with greater weight than helices and sheets. Fragments are sampled with a weight computed from the B-factors described earlier.

Finally, the third major class of fragments is generated using the residue groupings from KINARI. Since each residue grouping is predicted to moved together, a fragment is defined for each interval of residues *between* the residue groupings (intervals at the ends are not counted). In other words, the parts of the protein that we wish to perturb are the residues that lie in between tertiary structures, which we call *hinges* in this work. Note that this work uses 'hinges' in a different manner as is used in the rigidity analysis community. When these hinges are perturbed, the surrounding parts move together as a rigid body. Each fragment is weighted using B-factors and can be perturbed using *loop sampling* (10%), *random single perturbation* (35%), *randomize all* (20%), or *rigid-body sampling* (30%). This class of fragments is sampled 50% of the time.

The increased emphasis on tertiary structures is more aligned with the intended use of SIMS for sampling large backbone motions. All of the percentages given above were determined empirically and further research may fine tune these values.

Finally, recall that SIMS uses a projection that maps a conformation to the coverage grid. The projection can be automatically generated using B-factors. For a system with n residues, the projection will be of dimension $d \times 4n$, where d is the dimension of the projection. SIMS uses the sine and cosine of each dihedral angle in the system (two per residue) for a total of $4n$ degrees of freedom. The sine and cosines are done to embed the angles to a Euclidean space. We can use the B-factors to define the projection used by SIMS for the coverage grid. Again we extract n B-factors corresponding to the alpha carbon atoms in the backbone. The B-factors are processed to produce $f(t_i)$ for each residue. A vector of size $1 \times 4n$ is created by replicating each B-factor four times consecutively. This operation is done because every four elements in a single row of the projection correspond to a single residue.

The other $d-1$ dimensions are generated randomly. For each extra dimension, we generate a $1 \times 4n$ random vector, where each element is drawn from a standard Normal distribution. Finally, the vectors are made orthonormal using the Gram-Schmidt process. The full projection is constructed vertically with the B-factor row at the top and the other randomly generated rows below to get a $d \times 4n$ matrix. Conformational sampling using this projection is compared with the automatically generated projection used in (Novinskaya *et al.*, 2017).

3 Results

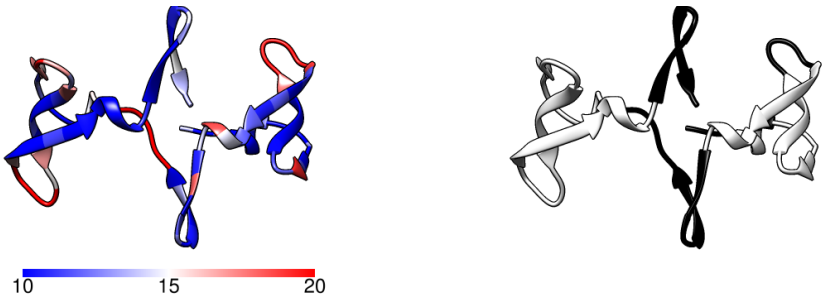
Our main objective was to determine the effect our new methodologies had on the diversity of the conformations sampled. All of our experiments are run on a single compute node with two Intel E5-2650v2 Ivy Bridge EP processors for a total of 16 cores, where each core runs an instance of SIMS. All runs are done for 100 minutes and write conformations to disk every 10 minutes (restart frequency is 6 restarts per hour). Energy thresholds for all the experiments are set to the value 0 because past experiments tend to show that conformations with a positive Rosetta score have some degree of steric clashes. All the projections used are 2-dimensional. In the discussion below, we call the version of SIMS with the new methods, "SIMS 2.0," which includes the global coverage grid implementation along with the new perturbation strategy defined in the schema as described in the previous section. The version without the new methods is called "Naive SIMS."

3.1 Proteins Used in Experiments

We illustrate the benefits of our new methods on four proteins of varying sizes: Cyanovirin-N (CVN) (Botos *et al.*, 2002), Calcium-loaded Calmodulin (CaM) (Anthis *et al.*, 2011), Ribose-Binding Protein (RBP) (Björkman *et al.*, 1994; Björkman and Mowbray, 1998), Maltodextrin-Binding Protein (MBP) (Quiocho *et al.*, 1997), and a single subunit of GroEL (Skjaerven *et al.*, 2011, 2012). CVN, CaM, RBP, and MBP are smaller sized proteins that we have previously studied in the context of evaluating random projections (Novinskaya *et al.*, 2017). The GroEL subunit is a larger and more constrained system consisting of about 500 residues. We also depict the B-factors and the KINARI information onto the structures to give a sense of how the new method is defining in the schema.

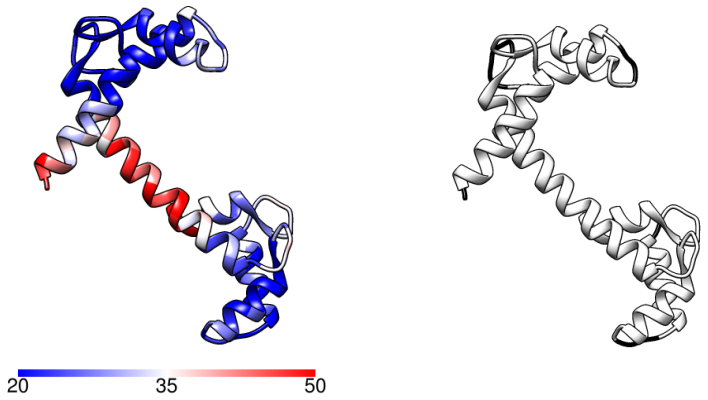
3.1.1 Cyanovirin-N

CVN is a 101 residue bacterial protein (PDB 3EZM) that exhibits antiviral activity towards the human immunodeficiency virus. CVN shows large scale motions from the correlated activity of three loop regions at residues 24–28, 50–55, and 75–80. These same loop regions are found as



(a) (b)

Figure 2: Cyanovirin-N. a) Colored by B-factors; b) Colors based on KINARI web output: clusters (white), hinges (black). The B-factors and hinges reflect the flexibility of the three loop regions at residues 24–28, 50–55, and 75–80.



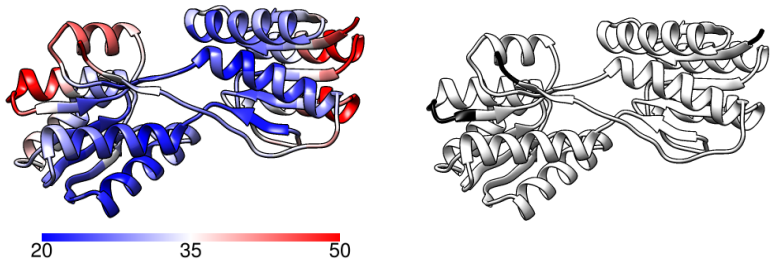
(a) (b)

Figure 3: Calmodulin. a) Colored by B-factors; b) Colors based on KINARI web output: clusters (white), hinges (black). Note that the B-factors reflect the flexibility of the middle alpha helix which SIMS 2.0 could exploit.

flexible from Fig. 2a. From Fig. 2b, KINARI classifies two of these these loop regions (residues 24–28 and 40–54) as hinges (which we defined in the previous section as the residues in between residue groupings). When constructing the schema as described in the previous section, the range of the B-factors are 10–20, instead of the default 20–50.

3.1.2 Calmodulin

CaM is a 144 residue protein (PDB 1CLL) involved with interactions between calcium ions and other proteins. B-factors show the flexible parts of the protein are found in residues 5–20, 35–41, 52–57, 67–80, 87–93, 107–116, and 126–129, which is represented in Fig. 3a. Note that the flexible helix at residues 67–80 would have been treated as a more stable part of the structure (and hence, not perturbed frequently) if only secondary structure information was used. The computed hinges in Fig. 3b are loops located at residues 41–43, 57–62, and 114–116.



(a) (b)

Figure 4: Ribose-Binding Protein. a) Colored by B-factors; b) Colors based on KINARI web output: clusters (white), hinges (black). The hinges computed with KINARI do not correspond to the three loop regions in the center. However the B-factors for these loop regions are indeed weighted more heavily than the two domains.

3.1.3 Ribose-Binding Protein

RBP is a 271 residue protein (PDB 1URP, chain A) that consists of two domains connected by three loop regions located at 91–104, 226–237, and 253–269. The first two regions are more constrained and have to move in a coordinated way. Interestingly, the B-factor distribution in Fig. 4a shows that the most flexible parts are mainly the alpha helices at the end. The KINARI output in Fig. 4b also predicted that most of the protein move together (residues 3–205 and 211–268), leaving a single hinge at residues 206–210 that is not part of the three main loop regions. Nevertheless, the B-factors for the main loop regions indeed show greater flexibility than the surrounding two domains.

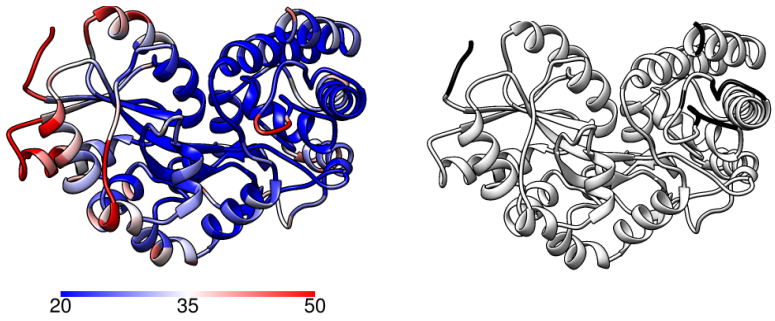
3.1.4 Maltodextrin-Binding Protein

MBP is a 370 residue protein (PRB 3MBP) that consists of two domains on each end terminal. MBP is known to exhibit protein-wide conformational changes between the open and bound forms that involves movement in nearly all the residues. The B-factor distribution in Fig. 5a show most flexibility at the extreme ends of the protein, which will allow the schema to focus on the two domains. However, KINARI predicted that most of the protein is rigid, so there is only a single major hinge shown in Fig. 5b.

3.1.5 GroEL Subunit

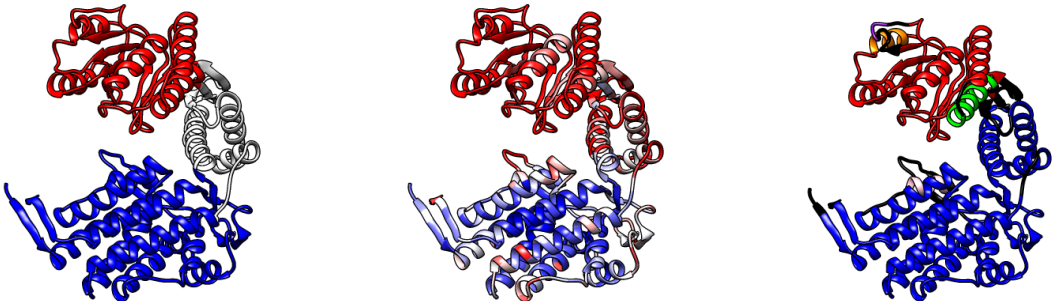
GroEL (PDB 1XCK) is a molecular chaperone consisting of 14 identical subunits forming two heptameric rings. We extract out chain A and use this as input to KINARI web. Each subunit consists of 524 residues arranged into three domains (Fig. 6a): equatorial (1–133, 409–524), intermediate (134–190, 377–408), and apical (191–376). The apical domain has the most movement, facilitated by hinges located in the intermediate domain (Skjaerven *et al.*, 2011, 2012). The equatorial domain remains mostly stable.

In Fig. 6b, notice that the high B-factors correlate to apical domain which is known to be the most flexible part. Fig. 6c shows the parts of the subunit from which we treat as hinges. Note how the hinges are mostly loops located between major helices/sheets in the system. Perturbing these hinges will in turn affect the alpha helices and beta sheets, analogous to a rigid body transform.



(a) (b)

Figure 5: Maltose-Binding Protein. a) Colored by B-factors; b) Colors based on KINARI web output: clusters (white), hinges (black). There was only a single major hinge detected by KINARI. However, the B-factor distribution will result in weighting the two domains highly.



(a) (b) (c)

Figure 6: A single subunit of GroEL. a) Colored by domain: equatorial (blue), intermediate (white), apical (red); b) Colored by B-factors; c) Colors based on KINARI web output: clusters (non-black), hinges (black). The B-factors reflect the flexibility of the apical domain while hinges roughly define regions between major domains.

3.2 Increased Number of Generated Conformations

In this section, we want to get a sense of how many conformations SIMS 2.0 can produce given a certain number of cores. With conformational sampling for larger systems the energy computation becomes more expensive, so the rate in which conformations are produced becomes vital. Producing more conformations is more efficient in the sense that the method is rejecting conformations less often. Note that all of the conformations produced are required to be below an energy threshold. We run experiments on a varied number of cores (1, 4, and 16) to assess the effect that our new methods have on scalability. Table 1 records the average number of conformations produced.

Table 1 clearly shows that as the number of cores used increases, the difference in the number of conformations produced by SIMS 2.0 compared to Naive SIMS increases. Note that all of the conformations produced are below the user-defined energy threshold. For a given number of cores, the rate in which conformations are being produced by SIMS 2.0 is greater

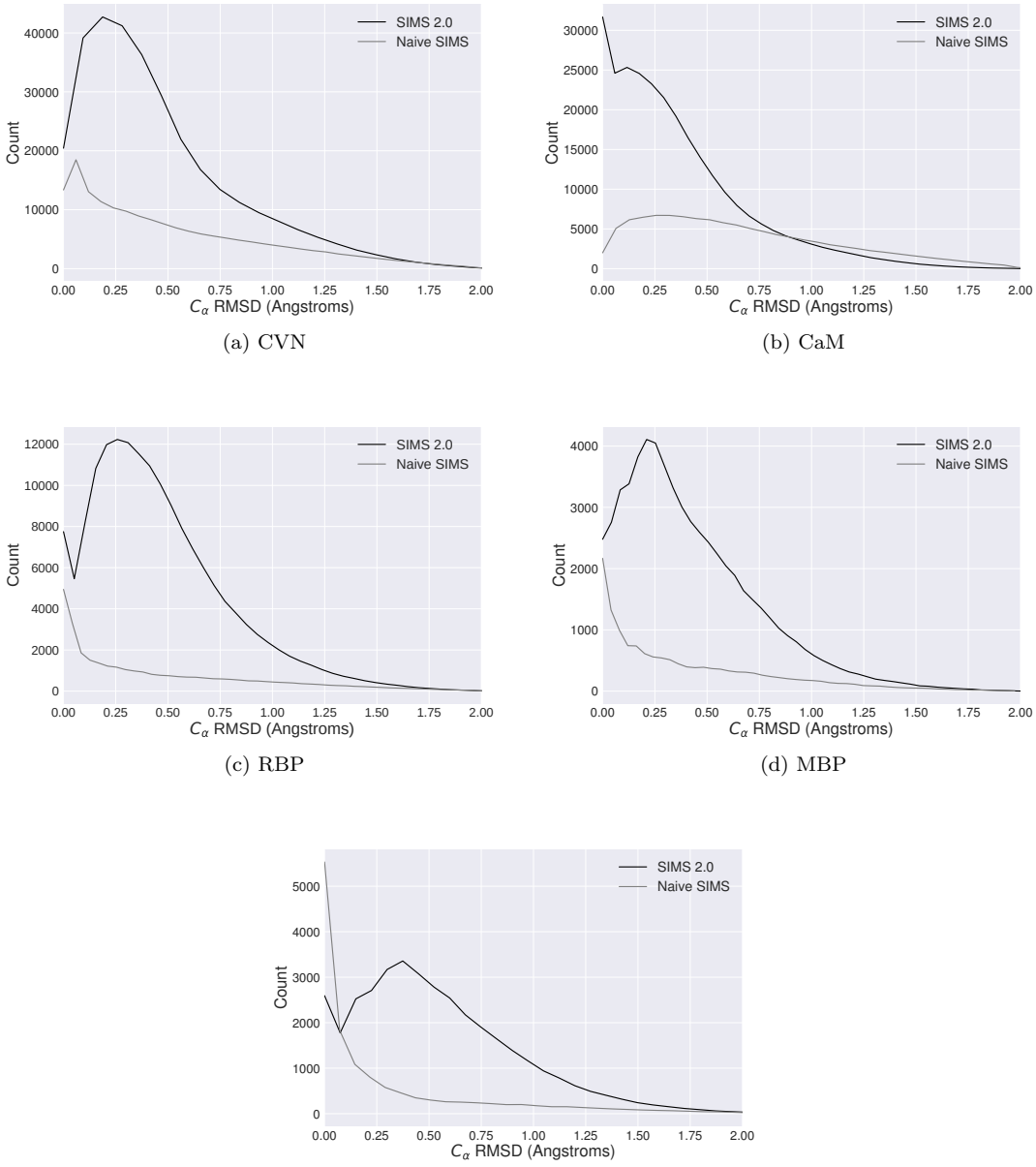
than naive SIMS. Hence, the rejection rate of the proposed conformations is lower in SIMS 2.0. As discussed earlier, high rejection rates in sampling for larger proteins was an issue preventing the scalability of robotics-inspired approaches (in *how* conformations are generated). SIMS 2.0 would make more efficient use of resources when the search requires up to 16 cores.

3.3 Improved conformational space coverage

The results from the previous section say nothing about the *diversity* of the conformations produced. In this set of experiments, we fix the number of cores to 16 and assess how well SIMS 2.0 generates diverse conformations. We use C_α RMSD to measure distances between conformations to emphasize the changes in protein backbone.

3.3.1 Nearest Neighbor Distances

We first measure the closeness of the conformations from each other. This is done by tracking the distance of each conformation to its nearest neighbor. This is a measure of how "spread out" the conformations are from each other. If the conformations are all similar to its neighbor, then the algorithm may not have produced a structurally diverse set of conformations. Another way to interpret a small value using this measure is that neighboring conformations are more likely to have been sampled next to each other (one conformation was perturbed to get the other). So if the average nearest neighbor distance is higher for one method, then the average effect of each perturbation was greater and hence, produce a more rapid exploration. From Fig. 7 we see that SIMS 2.0 indeed produces conformations that are farther apart from each other compared to "Naive SIMS."



(a) CVN

(b) CaM

(c) RBP

(d) MBP

(e) GroEL

Figure 7: Density of nearest neighbor distances (averaged over ten runs) for CVN (a), CaM (b), RBP (c), MBP (d), and GroEL subunit (e). SIMS 2.0 produces more conformations that are farther apart from each other.

3.3.2 Expansiveness

We now measure the expansiveness of the conformational search. A more expansive search means that farther parts of the conformational space are sampled given the same starting point. Fig. 8 shows a density plot of the distances of each conformation from the start conformation. The same start conformation was used for these experiments for a given protein. In addition to producing more conformations, SIMS 2.0 also produces more conformations that are farther from the start. Therefore, SIMS 2.0 produces a more expansive search than "Naive SIMS."

3.3.3 Isolating each improvement

These results taken together show that SIMS 2.0 produces more diverse conformations with the new methods compared to "Naive SIMS." We now end this section by investigating which specific method contributed most to the improved expansiveness. We focus the following experiments on the GroEL subunit system. Results for the other proteins were qualitatively similar.

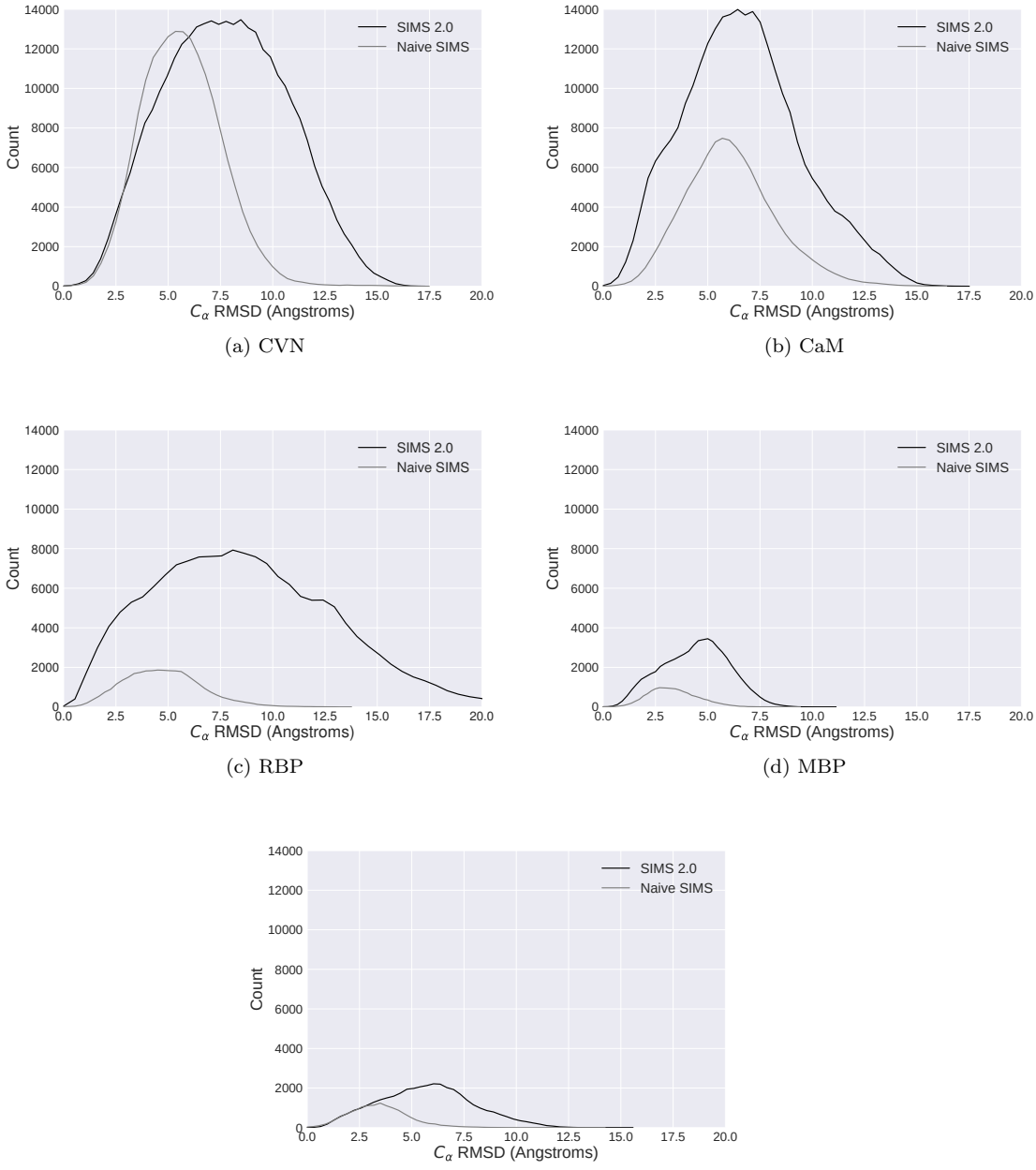
We first ran an experiment with SIMS 2.0, where the global coverage grid did not send summarized coverage estimates when a SIMS run restarted. This run is similar to "Naive SIMS " except with the new perturbation strategy. Fig. 9 shows the effect this had on expansiveness. It appears that since no synchronization was occurring, the exploration was not as expansive likely due to the increased amount of repeated work done amongst the cores.

Next we focus on the projection defined by B-factors. We ran two additional experiments. The first is SIMS 2.0 using a random projection (*randomProj*) (Gipson *et al.*, 2013). The second is SIMS 2.0 using a projection constructed from secondary structure (*ssProj*) as mentioned in (Novinskaya *et al.*, 2017). The results in Fig. 9 imply that the projection definition is not vital to the exploration. The exploration using a random projection only appears to be marginally worse than one from SIMS 2.0. Additionally, the projection using secondary structure information does not provide much improvement over the runs using a random projection. This is likely due to the fact that we are using a linear, 2-dimensional projection to represent a complex, high-dimensional conformational space. Thus, even though we use B-factors to incorporate flexibility into the projection definition, this translates to a relatively small improvement in how coverage is computed since the space is so simplified and much information is lost in the projection operation.

Finally, we focus on the schema improvement. We ran an experiment with SIMS 2.0 using a schema defined using only secondary structure information. The new schema appears to contribute the greatest since the density curve with a *Naive schema* is most similar to the one corresponding to "Naive SIMS " even though the other improvements are included. The density curve with a *Naive schema* also implies that the improvement in the number of conformations produced was also due to the new schema, since the height of the density curve of *Naive schema* is lower than the one from SIMS 2.0. This demonstrates how important the schema is to the exploration because it essentially encodes how SIMS 2.0 *explores* the space.

3.4 Discussion

We have shown that with the addition of the global coverage grid and a perturbation strategy enriched with flexibility information, SIMS 2.0 can more efficiently generate conformations that are also distributed more diversely as compared to Naive SIMS. All of the comparisons done were under the same computing budget, and the conformations generated were all under the same energy threshold. These new methodologies present promising steps toward making robotics-inspired conformational sampling better suited for larger proteins. While the example proteins



(a) CVN

(b) CaM

(c) RBP

(d) MBP

(e) GroEL

Figure 8: Density of distances to start conformation (averaged over ten runs) for CVN (a), CaM (b), RBP (c), MBP (d), and GroEL subunit (e). SIMS 2.0 produces more conformations that are farther from the start conformation.

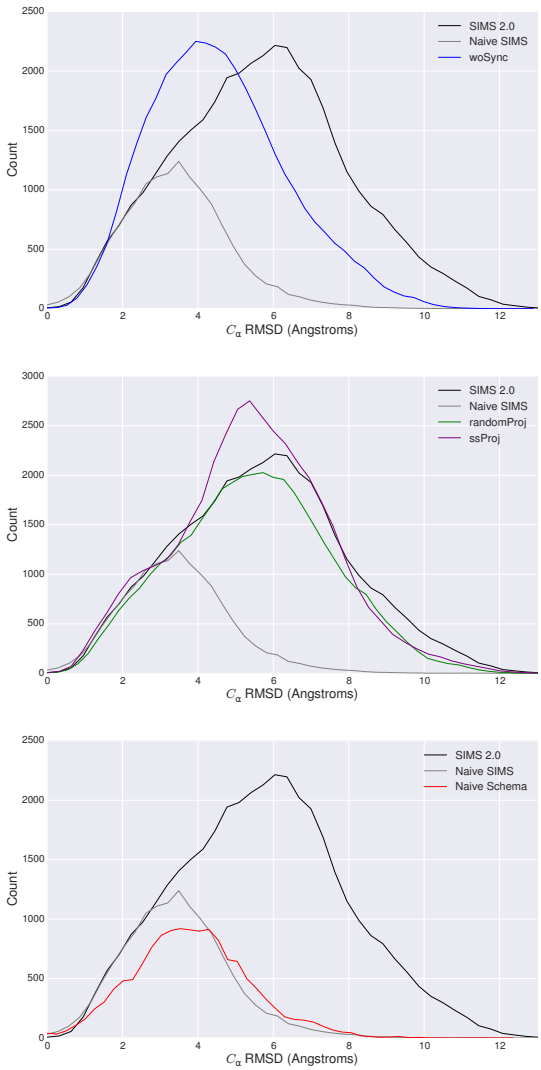


Figure 9: Density of distances to start conformation (average over ten runs) for the GroEL subunit. (TOP) *woSync* (blue) corresponds to the runs where the synchronization of coverage statistics was turned off. Since the distribution shifted to the left, the exploration was less expansive than SIMS 2.0. (MIDDLE) *randomProj* (green) corresponds to the runs where a random projection was used. *ssProj* (purple) corresponds to the runs where secondary structure information was used to construct the projection. Since the distributions barely shifted, the definition of the projection does not appear to have much impact on the expansiveness. (BOTTOM) The new schema appears to have the greatest impact since the run that uses a schema with only secondary structure information (*Naive Schema*, red) has a density curve that is most similar to "Naive SIMS " (*old*, gray).

ranged from about 100 to 500 residues, additional work may be required for proteins that are greater than 500 residues in length. Since our results show that the construction of the perturbation strategy contributed most to the improvement in structural diversity, we may have to adjust the parameters in the schema construction in order to handle even larger systems. For example, the KINARI results for RBP and MBP did not provide any significant decompositions of the protein that the schema could exploit, and perhaps adjusting the parameters in the KINARI software could provide a better set of residue groupings. Our results also show that for a given number of cores, SIMS 2.0 produces conformations at a faster rate than Naive SIMS. This points to the additional benefit of adding flexibility information to the perturbation strategy. Our experiments only went up to 16 cores to represent more modest computational resources like a high-end desktop. However, more work is needed to characterize how SIMS 2.0 performs in a large-scale setting with hundreds of cores.

4 Conclusion

Robotics-inspired approaches rely heavily on two critical decisions: *where* to focus sampling in the conformational space and *how* to sample new conformations. As the protein size increases, robotics-inspired methods that run across multiple cores become memory-intensive, coverage estimation becomes more expensive to maintain, and the definition of a useful perturbation strategy becomes difficult. In this paper, we introduced two methodologies to maintain and enhance the diversity of conformations sampled and implemented these in SIMS. First, we proposed to maintain a *global* coverage grid that eliminates the memory bottleneck for large proteins and enables efficient conformational sampling across multiple cores. Next, we presented a perturbation strategy using flexibility information from B-factors, secondary structure, and rigidity analysis.

For SIMS, our results show a significant improvement in the diversity of the conformations generated with our methods for proteins consisting of up to 500 residues. We also showed that our methods increased the number of conformations generated at faster rate as the number of cores increased. We demonstrated that the new perturbation strategy provided the most dramatic change in diversity in terms of expansiveness (as measured in terms of distance from the start conformation). Our methods solve both the memory problem associated with SIMS as well as the coordination problem of sampling across multiple cores. Our new perturbation strategy is also a practically free way to obtain informed moves that improve structural diversity over a simple, naive strategy.

For future work, we plan to apply these ideas to other robotics-inspired conformational sampling frameworks. The ideas introduced in Section 2.2 apply to robotics-inspired methods that keep track of coverage, while the ideas in Section 2.3 apply more generally to robotics-inspired methods. Also, we will work on improving SIMS 2.0 through a variety of directions. We can combine our global coverage grid with other existing perturbation strategies. Furthermore, we will also begin to investigate new ways to keep track of coverage. Our experiments showed that the currently-used projection definition does not greatly affect the expansiveness of the exploration so we have begun to look at non-linear projections. This work also showed that the most significant improvement in terms of diversity came from the new perturbation strategy. We will investigate the benefits of a dynamically changing perturbation strategy that perturbs conformations differently as a function of the currently chosen conformation.

Acknowledgments

This work is supported in part by NSF CCF 1423304, Rice University funds, and a training fellowship from the Gulf Coast Consortia on the Training Program in Biomedical Informatics, National Library of Medicine T15LM007093. Experiments were run on equipment that is supported in part by the Data Analysis and Visualization Cyberinfrastructure funded by NSF under Grant OCI 0959097, as well as on equipment that is supported by the Cyberinfrastructure for Computational Research funded by NSF under Grant CNS 0821727.

Author Disclosure Statement

No competing financial interests exist.

References

Al-Bluwi, I., Siméon, T., and Cortés, J. 2012. Motion planning algorithms for molecular simulations: A survey. *Computer Science Review* 6, 125–143. ISSN 15740137.

Al-Bluwi, I., Vaisset, M., Siméon, T., and Cortés, J. 2013. Modeling protein conformational transitions by a combination of coarse-grained normal mode analysis and robotics-inspired methods. *BMC Struct Biol* 13 Suppl 1, S2.

Amato, N. M., Dill, K. A., and Song, G. 2003. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. *J. Comp. Bio.* 10, 239–255.

Andersson, E., Hsieh, R., Szeto, H., Farhoodi, R., Jagodzinski, F., and Haspel, N. 2016. Assessing how multiple mutations affect protein stability using rigid cluster size distributions. In *proc. of IEEE-ICCABS (International Conference on Computational Advances in Bio and Medical Sciences)*.

Anthis, N. J., Doucleff, M., and Clore, G. M. 2011. Transient, sparsely populated compact states of Apo and calcium-loaded Calmodulin probed by paramagnetic relaxation enhancement: Interplay of conformational selection and induced fit. *J. Am. Chem. Soc.* 133, 18966–18974.

Björkman, A. J., Binnie, R. A., Zhang, H., Cole, L. B., Hermodson, M. A., and Mowbray, S. L. 1994. Probing protein-protein interactions. The ribose-binding protein in bacterial transport and chemotaxis. *J. Biol. Chem.* 269, 30206–11. ISSN 0021-9258.

Björkman, A. J. and Mowbray, S. L. 1998. Multiple open forms of ribose-binding protein trace the path of its conformational change. *J. Mol. Biol.* 279, 651–64.

Botos, I., O'Keefe, B. R., Shenoy, S. R., Cartner, L. K., Ratner, D. M., Seeberger, P. H., Boyd, M. R., and Wlodawer, A. 2002. Structures of the complexes of a potent anti-HIV protein Cyanovirin-N and high mannose oligosaccharides. *J. Biol. Chem.* 277, 34336–42.

Carlson, H. A. 2002. Protein flexibility is an important component of structure-based drug discovery. *Curr. Pharm. Des.* 8, 1571–1578.

Clausen, R. and Shehu, A. 2015. A data-driven evolutionary algorithm for mapping multibasin protein energy landscapes. *Journal of Computational Biology* 22, 844–860. ISSN 1066-5277.

Das, R. and Baker, D. 2008. Macromolecular modeling with Rosetta. *Annu Rev Biochem* 77, 363–82.

Devaurs, D., Antunes, D. A., Papanastasiou, M., Moll, M., Ricklin, D., Lambris, J. D., and Kavraki, L. E. 2017. Coarse-grained conformational sampling of protein structure improves the fit to experimental hydrogen-exchange data. *Frontiers in Molecular Biosciences* 4.

Devaurs, D., Papanastasiou, M., Antunes, D. A., Abella, J. R., Moll, M., Ricklin, D., Lambris, J. D., and Kavraki, L. E. 2016. Native state of complement protein C3d analysed via hydrogen exchange and conformational sampling. In *International Conference on Intelligent Biology and Medicine (ICIBM)*.

Devaurs, D., Simeon, T., and Cortes, J. 2013. Parallelizing RRT on large-scale distributed-memory architectures. *IEEE Trans. on Robotics* 29, 571–579.

Fox, N., Jagodzinski, F., Li, Y., and Streinu, I. 2011. Kinari-web: a server for protein rigidity analysis. *Nucleic Acids Res* 39, W177–83.

Gipson, B., Hsu, D., Kavraki, L. E., *et al.* 2012. Computational models of protein kinematics and dynamics: Beyond simulation. *Annu. Rev. Anal. Chem.* 5, 273–291.

Gipson, B., Moll, M., and Kavraki, L. E. 2013. SIMS: A hybrid method for rapid conformational analysis. *PLoS ONE* 8, e68826.

Haspel, N., Moll, M., Baker, M. L., Chiu, W., and Kavraki, L. E. 2010. Tracing conformational changes in proteins. *BMC Structural Biology* 10, S1.

Hsu, D., Latombe, J.-C., and Motwani, R. 1999. Path planning in expansive configuration spaces. *Intl. J. of Computational Geometry and Applications* 9, 495–512.

Ichnowski, J. and Alterovitz, R. 2012. Parallel sampling-based motion planning with superlinear speedup. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 1206–1212. ISSN 2153-0858.

Jaillet, L., Corcho, F. J., Pérez, J.-J., and Cortés, J. 2011. Randomized tree construction algorithm to explore energy landscapes. *J Comput Chem* 32, 3464–74.

Leaver-Fay, A., Tyka, M., Lewis, S. M., Lange, O. F., Thompson, J., Jacak, R., Kaufman, K. W., Renfrew, P. D., Smith, C. A., Sheffler, W., Davis, I. W., Cooper, S., Treuille, A., Mandell, D. J., Richter, F., Ban, Y.-E. A., Fleishman, S. J., Corn, J. E., Kim, D. E., Lyskov, S., Berrondo, M., Mentzer, S., Popović, Z., Havranek, J. J., Karanicolas, J., Das, R., Meiler, J., Kortemme, T., Gray, J. J., Kuhlman, B., Baker, D., and Bradley, P. 2011. Rosetta3: An object-oriented software suite for the simulation and design of macromolecules. In M. L. Johnson and L. Brand, editors, *Computer Methods, Part C*, volume 487 of *Methods in Enzymology*, chapter 19, pages 545 – 574. Elsevier.

Lee, A. and Streinu, I. 2008. Pebble game algorithms and sparse graphs. *Discrete Mathematics* 308, 1425 – 1437. ISSN 0012-365X. Third European Conference on CombinatoricsGraph Theory and ApplicationsThird European Conference on Combinatorics.

Luo, D. and Haspel, N. 2013. Multi-resolution rigidity-based sampling of protein conformational paths. In *CSBW (Computational Structural Bioinformatics Workshop), in proc. of ACM-BCB (ACM International conference on Bioinformatics and Computational Biology)*, pages 787–793.

Marion, D. 2013. An introduction to biological nmr spectroscopy. *Molecular and Cellular Proteomics* .

Maximova, T., Moffatt, R., Ma, B., Nussinov, R., and Shehu, A. 2016. Principles and Overview of Sampling Methods for Modeling Macromolecular Structure and Dynamics. *PLoS Computational Biology* 12, 1–70. ISSN 1553-7358.

Novinskaya, A., Devaurs, D., Moll, M., and Kavraki, L. E. 2017. Defining Low-Dimensional Projections to Guide Protein Conformational Sampling. *Journal of Computational Biology* 24(1), 79–89.

Pachov, D. V. and van den Bedem, H. 2015. Nullspace sampling with holonomic constraints reveals molecular mechanisms of protein g?s. *PLOS Computational Biology* 11, 1–24.

Paquet, E. and Viktor, H. L. 2015. Molecular Dynamics , Monte Carlo Simulations , and Langevin Dynamics : A Computational Review. *BioMed Research International* 2015.

Plaku, E., Bekris, K. E., Chen, B. Y., Ladd, A. M., and Kavraki, L. E. 2005. Sampling-based roadmap of trees for parallel motion planning. *IEEE Trans. on Robotics* 21, 597–608.

Powell, D. R. 2016. Review of x-ray crystallography. *Journal of Chemical Education* 93, 591–592.

Quiocho, F. A., Spurlino, J. C., and Rodseth, L. E. 1997. Extensive features of tight oligosaccharide binding revealed in high-resolution structures of the maltodextrin transport/chemosensory receptor. *Structure* 5, 997 – 1015. ISSN 0969-2126.

Raveh, B., Enosh, A., Schueler-Furman, O., and Halperin, D. 2009. Rapid sampling of molecular motions with prior information constraints. *PLoS Comput Biol* 5, e1000295.

Shehu, A. and Kavraki, L. E. 2012. Modeling structures and motions of loops in protein molecules. *Entropy* 14, 252–290. ISSN 1099-4300.

Shehu, A. and Olson, B. 2010. Guiding the search for native-like protein conformations with an ab-initio tree-based exploration. *The International Journal of Robotics Research* 29, 1106–1127.

Shehu, A. and Plaku, E. 2016. A Survey of Computational Treatments of Biomolecules by Robotics-Inspired Methods Modeling Equilibrium Structure and Dynamics. *Journal of Artificial Intelligence Research* 57, 509–572.

Skjaerven, L., Grant, B., Muga, A., Teigen, K., McCammon, J. A., Reuter, N., and Martinez, A. 2011. Conformational Sampling and Nucleotide-Dependent Transitions of the GroEL Subunit Probed by Unbiased Molecular Dynamics Simulations. *PLoS Computational Biology* 7, e1002004. ISSN 1553-734X.

Skjaerven, L., Muga, A., Reuter, N., and Martinez, A. 2012. A dynamic model of long-range conformational adaptations triggered by nucleotide binding in groel-groes. *Proteins: Structure, Function, and Bioinformatics* 80, 2333–2346. ISSN 1097-0134.

Stein, A. and Kortemme, T. 2013. Improvements to robotics-inspired conformational sampling in rosetta. *PLOS ONE* 8, 1–13.

Sucan, I. A. and Kavraki, L. E. 2009. *Kinodynamic Motion Planning by Interior-Exterior Cell Exploration*, volume 57, pages 449–464. Springer.

Şucan, I. A., Moll, M., and Kavraki, L. E. 2012. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine* 19, 72–82. http://ompl.kavrakilab.org.

Tapia, L., Thomas, S., and Amato, N. M. 2010. A motion planning approach to studying molecular motions. *Communications in Information and Systems* 10, 53–68.

Thomas, S., Tang, X., Tapia, L., and Amato, N. M. 2007. Simulating protein motions with rigidity analysis. *J Comput Biol* 14, 839–855.

Trueblood, K. N., Bürgi, H.-B., Burzlaff, H., Dunitz, J. D., Gramaccioli, C. M., Schulz, H. H., Shmueli, U., and Abrahams, S. C. 1996. Atomic Dispacement Parameter Nomenclature. Report of a Subcommittee on Atomic Displacement Parameter Nomenclature. *Acta Crystallographica Section A* 52, 770–781.

Vitalis, A. and Pappu, R. V. 2009. Methods for Monte Carlo simulations of biomacromolecules. *Annual reports in computational chemistry* 5, 49–76. ISSN 1574-1400.

Wei, G., Xi, W., Nussinov, R., and Ma, B. 2016. Protein Ensembles: How Does Nature Harness Thermodynamic Fluctuations for Life? The Diverse Functional Roles of Conformational Ensembles in the Cell. *Chemical Reviews* 116, 6516–6551. ISSN 0009-2665.

Xu, X., Yan, C., Wohlhueter, R., *et al.* 2015. Integrative modeling of macromolecular assemblies from low to near-atomic resolution. *Comput. Struct. Biotechnol. J.* 13, 492–503.

Yao, P., Dhanik, A., Marz, N., Propper, R., Kou, C., Liu, G., van den Bedem, H., Latombe, J.-C., Halperin-Landsberg, I., and Altman, R. B. 2008. Efficient algorithms to explore conformation spaces of flexible protein loops. *IEEE/ACM Trans Comput Biol Bioinform* 5, 534–545.

Yuan, Z., Bailey, T. L., and Teasdale, R. D. 2005. Prediction of protein b-factor profiles. *Proteins: Structure, Function, and Bioinformatics* 58, 905–912. ISSN 1097-0134.

Table 1: Number of Conformations Produced (averaged over 10 runs) for four different proteins. Standard deviations shown in parentheses. As the number of cores increase, the new version of SIMS produces conformations at a faster rate.

		1 Core	4 Cores	16 Cores
CVN	SIMS 2.0	28294 (2203)	122067 (10903)	315989 (17105)
	Naive SIMS	10710 (1501)	43136 (3064)	176813 (6721)
	Difference	**+17584**	**+78931**	**+139176**
CaM	SIMS 2.0	25977 (2950)	106321 (9415)	272604 (6154)
	Naive SIMS	6822 (789)	27383 (1314)	111247 (4348)
	Difference	**+19155**	**+78938**	**+161357**
RBP	SIMS 2.0	13078 (1385)	50693 (1689)	164588 (6032)
	Naive SIMS	2183 (177)	8729 (803)	32125 (1024)
	Difference	**+10895**	**+41964**	**+132463**
MBP	SIMS 2.0	4721 (194)	19483 (2836)	60921 (6846)
	Naive SIMS	997 (74)	4747 (1891)	14699 (2016)
	Difference	**+3724**	**+14736**	**+46222**
GroEL Subunit	SIMS 2.0	3026 (162)	11120 (623)	38763 (2729)
	Naive SIMS	850 (39)	3495 (223)	13739 (392)
	Difference	**+2176**	**+7625**	**+25024**