EVOLUTIONARY TECHNIQUES FOR DE NOVO PROTEIN CONFORMATION ENSEMBLE GENERATION

by

Ahmed Bin Zaman

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computer Science

Committee:

_____	Dr. Amarda Shehu, Dissertation Director
_____	Dr. Kenneth De Jong, Committee Member
_____	Dr. Alexander Brodsky, Committee Member
_____	Dr. Wanli Qiao, Committee Member
_____	Dr. David Rosenblum, Department Chair
_____	Dr. Kenneth S. Ball, Dean, The Volgenau School of Engineering

Date: _____ Summer Semester 2021
George Mason University
Fairfax, VA

Evolutionary Techniques for De Novo Protein Conformation Ensemble Generation

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Ahmed Bin Zaman
Master of Science
George Mason University, 2020
Bachelor of Science
Shahjalal University of Science and Technology, 2012

Director: Dr. Amarda Shehu, Professor
Department of Computer Science

Summer Semester 2021
George Mason University
Fairfax, VA

Dedication

I dedicate this dissertation to the people who are broken inside.

Acknowledgments

I would like to thank my advisor Dr. Amarda Shehu for all her help, support, and guidance for my research work and graduate studies. She has been kind and patient enough to bear with a highly idiosyncratic student. In batman terms, she is a vocal guardian, a watchful protector, a colorful knight for me. I would also like to thank a few of the past and current members of Shehu Lab for collaborating with me. I also thank all the other members of the lab for being my friends and helping me through the stressful life of a PhD student by providing teeth to laugh along, shoulders to fight along, and food to steal (or snatch) when I am hungry.

The ARGO cluster and the people associated with maintaining it deserve a big thank you on behalf of me for making it possible to catch the deadlines and on behalf of my personal computer for saving its life. I also thank the researchers within the community whose contributions through papers, reports, and software have helped me in my research and knowledge. I would also like to thank all the department/university staff (especially, the cleaning staff) for all their contributions in making me feel at home.

I would like to thank Dr. Kenneth De Jong for his collaboration and insightful feedback from time to time. I would also like to thank Dr. Wanli Qiao for thoughtful discussions and his suggestions for a project. I am also thankful to Dr. Alexander Brodsky for serving in my comprehensive exam committee and providing useful pointers for my development. I thank all three of them for their willingness to serve in my PhD dissertation committee and for providing valuable insights about my work.

I am eternally grateful to my friends and family for keeping me going for all these years, they are absolutely incredible.

Table of Contents

List of Tables

List of Figures

Abstract

EVOLUTIONARY TECHNIQUES FOR DE NOVO PROTEIN CONFORMATION ENSEMBLE GENERATION

Ahmed Bin Zaman, PhD

George Mason University, 2021

Dissertation Director: Dr. Amarda Shehu

The conformations in which a protein molecule arranges its amino-acid chain are primary determinants of its ability to interact with other molecules in the living cell. Discovering the functionally-relevant conformations of a protein is crucial to elucidating its functional repertoire and even further our understanding of diseases driven by mutations that affect the ability of a protein to assume specific conformations.

Discovering the possibly diverse set of biologically-relevant conformations of a protein from knowledge of its amino-acid sequence alone remains an outstanding challenge. While progress has been made in this direction, most notably by AlphaFold2, in discovering what is often referred to as a native structure, methods based on machine learning, including deep learning, are limited in their ability to see the entirety of the conformation space of a protein. While obtaining one conformation is sufficient for some proteins, many others are involved in many cellular reactions in the cell, and harness their ability to assume different conformations to achieve functional plasticity. Nowhere is the ability of proteins to assume different conformations more visible in the public eye than nowadays; we are all familiar with images that show the spike protein in the SARS-CoV2 virus switching between an open and a closed conformation to elude our immune system and strike at just the right

moment by binding with the ACE receptors in its closed conformation.

This dissertation presents a way forward on exploring the conformation space of a given protein molecule, when the only information available is the sequence of amino acids. We refer to this problem as de-novo protein conformation ensemble generation. In this dissertation, we present several novel stochastic optimization algorithms that operate under the umbrella of evolutionary computation. We show that these algorithms are able to balance between the known challenges of exploration and exploitation and capture meaningful representations of the conformation space, despite its size and complexity. In particular, we show that they are able to capture the presence of significantly different functionally-relevant conformations in metamorphic proteins, which we also provide to the community as a benchmark to further advance research on this problem. The work presented in this dissertation represents important groundwork for researchers aiming to improve protein conformation sampling in order to better understand the structural and functional plasticity of protein molecules in all their exquisite complexity.

Chapter 1: Introduction

Living cells use proteins as molecular instruments to accomplish biological functions. The spatial arrangements or conformations of the three-dimensional/tertiary structures of a protein molecule in which the amino acid units of the protein organize themselves under physiological conditions are key to determining its array of activities in the cell. Proteins assume different biologically-active/functional conformations to interface with other molecular partners and modulate complex biological activities [2]. Many diseases can occur due to proteins failing to adopt appropriate functional conformation, such as cancers, Alzheimer's, and Huntington's disease [3, 4].

To understand the conformation-function relationships of proteins, a vast body of work in molecular biology has been devoted to determining the biologically-active conformations. Early on, experimental techniques, such as X-ray crystallography, revealed static snapshots of protein molecules, capturing a protein in one conformation. Motivated in part by the inability of X-ray crystallography to generalize over different protein molecules, computational approaches stepped in. They leveraged a narrow formulation of the problem, where the goal was the determination of a single such conformation, also referred to as the native conformation, from a given protein amino-acid sequence (de novo) [5]. Impressive computational advances instigated via the "Critical Assessment of protein Structure Prediction" (CASP) competition were made over the years [6]. In December 2020, they culminated in the AlphaFold2 method, which, contrary to what the name suggests, presented a major advance in protein conformation determination. Reports from CASP14 suggest AlphaFold2 can now obtain a high-quality native conformation given an amino-acid sequence for possibly a large number of proteins [7].

However, in a largely detached thread in computational molecular biology, various researchers have advanced theory, experiment, and methods to reveal significant additional

protein complexity; that is, proteins are inherently dynamic systems using often large motions to switch between different stable and semi-stable conformations with which to bind to different molecular partners in the cell [8]. The dynamic view of proteins was evident in the early experimental structures obtained via Nuclear Magnetic Resonance (NMR); however, NMR is limited to reveal small conformational fluctuations. Then came cryo-electron microscopy, which revealed the diversity of native conformations assumed by a protein molecule [9].

Many researchers over the years have argued for broadening our computational treatment of proteins to account for the multiplicity of native conformations [10]. A complete protein functional conformation model should include all the possible active conformational states accessible by a protein molecule. The traditional view of an unique functional state of a protein undermines the fact that the "native" state of a protein is potentially a large number of conformational states. To capture all the active conformational states, we require methods that produce an ensemble of conformations that are functionally-relevant instead of a single conformation. This thesis focuses on designing and improving such methods.

However, the problem of generating ensembles that contain the native conformations of a given protein presents outstanding challenges, as it necessitates exploring a vast, high-dimensional space in search of possibly a very large number of functionally-relevant conformations. The majority of computational methods that can reveal possibly multiple active conformations for a given protein leverage deep insight about a specific protein of interest. For instance, a line of work leverages experimentally-known conformations of a protein to reveal latent coordinates over which to generate more conformations [11–16]. Other work is strictly limited to generating conformations that mediate the transition between two given conformations [17–23]. Several methods leverage collective coordinates to expedite numerical simulation (such as Molecular Dynamics simulation) [24, 25]. While beyond the scope of this thesis, adaptations to the classic Molecular Dynamics continue to be pursued to reveal the motion between two given conformations or enhance the exploration of the conformation space when starting from a known conformation rather than just the amino-acid

sequence [26]. This thesis truly explores the *de-novo* setting; where given an amino-acid sequence alone, it seeks to reveal various functionally-relevant conformations available to a protein. This setting is of supreme interest as most proteins do not have experimentally-known conformations revealed yet.

Generating conformation ensemble that includes the native conformations of a protein de novo is naturally posed as an optimization problem. It has been revealed theoretically and experimentally that the conformations representing the stable or semi-stable, long-time populated structural states observed in wet laboratories [27] occupy the deep and broad basins of the interatomic energy surfaces [28]. Thus, finding the native conformations correspond to finding the diverse minima in the corresponding energy landscape. However, this is very challenging as even finding just one such conformation has been shown to be NP-hard [29]. Moreover, while many semi-empirical energy functions have been devised in computational laboratories that attempt to approximate the interatomic energy of the conformations, many studies show how Rosetta [30], Amber [31], and other state-of-the-art energy functions [32] contain inherent inaccuracies that result in wildly rugged energy landscapes and steer the optimization process towards very low-energy conformations that are significantly different from the known native states (sometimes more than 10 Angstroms (Å) away in conformation space) [33–37]. This makes generating conformation ensembles possibly containing native conformations an extremely difficult task.

In addition to inaccurate energy models, the protein conformation space is vast and high-dimensional. If Cartesian coordinates of the atoms are considered as the underlying variables, the dimensionality would be in the thousands for a medium-size protein not exceeding 150 amino acids. If other representations are employed, such as using only dihedral angles as underlying variables (leaving bond lengths and bond angles in equilibrium/ideal values), the dimensionality goes down into the hundreds. Sampling in such high-dimensional and multimodal energy landscapes full of artifact minima poses a tremendous challenge for the conformation ensemble generation algorithms. Therefore, such algorithms try to ensure a broad, sample-based representation of the conformation space (and in turn the associated

energy surface) and not miss low-energy near-native conformations. The recommendation from developers is to generate as many conformations as can be afforded in order to increase the likelihood that some generated conformations reside near the unknown native conformations.

A critical challenge in stochastic search and optimization on such a complex, multimodal energy landscape is to attain a proper balance between exploration (seeing more of the search space) and exploitation (getting to better-scoring regions of the space) of the search space. Too much exploitation generally results in premature convergence where the optimization algorithm gets stuck in a suboptimal region in the search space. On the other hand, too much exploration (i.e. little exploitation) can cause the algorithm to search in wide range of regions without proper investigation of the promising regions in the landscape to find no optimal solution at all. This core issue of balancing exploration and exploitation is not addressed by the popular Simulated Annealing Metropolis Monte Carlo (SA-MMC) based de novo conformation sampling algorithms such as Rosetta [30] and Quark [38], that require multiple-restarts to obtain a conformation ensemble. Evolutionary Algorithms (EAs) are inherently better suited for tuning this balance in optimization problems [39] and have been shown to be effective for conformation ensemble generation [40, 41].

The growing evidence that existing energy functions are not reliable indicators of *nativeness* and often make for poor guides towards native conformations is prompting the community to rethink the proper role and utilization of energy functions in conformation ensemble generation. An increasing realization in the computational structural biology community at large and the de novo conformation sampling community in particular is that the quality of the energy function is perhaps as much if not more important than the quality of the sampling of the conformation space [35, 42, 43]. Some research has explored splitting an energy function into groups of terms and pursuing conformation ensemble generation in a multi-objective rather than a single-objective optimization setting; this line of work has shown improvements over single-objective optimization [41, 44–46] of energy functions.

However, achieving proper balance between multiple competing objectives remains a challenge. Some recent work has investigated doing away with existing energy functions and constructing new ones based on predicted contacts or distances of pairs of amino acids [47]. The latter line of research has been prompted by ever more powerful machine learning models capable of leveraging existing native conformation of proteins deposited in databases such as the Protein Data Bank (PDB) [48] to predict distances or contacts between amino acids in native conformations given an amino-acid sequence [49].

This dissertation focuses on tackling a wide range of challenges that arise in de novo conformation ensemble generation aiming to find multiple functional conformations of a protein using Evolutionary Computation (EC) techniques. The description of the challenges and our efforts to address them are discussed in Section 1.1. Briefly, the dissertation first attempts to overcome the shortcomings of the existing energy functions. In doing so, it explores ways to balance multiple energetic objectives and investigates the utility of incorporating contact information to guide the optimization process. The dissertation then focuses on reducing the size of the conformation ensemble generated by the conformation ensemble generation algorithms without sacrificing the ensemble quality to promote efficiency and practical use of such algorithms. Finally, the dissertation focuses on attaining a proper balance between exploration and exploitation of the conformation space to enhance sampling of the native conformations.

The rest of the dissertation is organized as follows. Chapter 2 provides an overview of the important concepts and related work for de novo conformation ensemble generation. The domain-specific knowledge leveraged for this work as well as the evaluation datasets and metrics are presented in Chapter 3. Chapter 4 describes the work for minimizing the shortcomings of energy functions. Chapter 5 presents the details for the work in reducing the generated conformation ensemble size. Chapter 6 provides the details for the work in balancing the exploration and exploitation to improve the optimization process. Finally, Chapter 7 concludes the dissertation with a summary and directions for the future work.

1.1 Challenges and Contributions

Improving the complex optimization process in the vast, high-dimensional, and multimodal protein conformation space to sample near-native conformations requires overcoming several obstacles. Sampling low-energy conformations from a broad range of regions in the energy landscape is necessary to produce an ensemble of functionally-relevant conformations. Therefore, a balance between the exploration and exploitation needs to be in place for an effective optimization algorithm. The rugged and multimodal nature of the fitness/energy landscape can cause a simple optimization algorithm to get stuck in a local optimum. Hence, the optimization algorithm should have some mechanism to avoid getting stuck and explore potential basins of attraction or niches. Also, utilizing the energy function as the sole optimization objective can be problematic because of their inaccuracies and unreliability, and optimizing multiple objectives has the potential to help sampling better quality conformations. Thus a multi-objective optimization algorithm needs to choose and manage the objectives in an effective way to guide the search towards promising regions in the landscape.

Practical use of the conformation ensemble generation algorithms is a concern that is often overlooked by the researchers. Numerous conformations are generated by the algorithms in an attempt to have a better chance that the final reported conformation ensemble (which consists of all the conformations generated by the algorithm before termination) is diverse enough to cover a sufficient number of minima possibly housing near-native conformations. A vast body of research beyond the scope of this dissertation utilize various selection schemes to tease out the conformations that are near-native among those in the generated conformation ensemble. The large size of the generated conformation ensemble means selection algorithms tasked with analyzing this ensemble to extract functionally relevant conformations have to additionally deal with a data size issue. Moreover, conformation ensemble generation algorithms generally sample in a simplified conformation space because of the vastness and dimensionality of the original space. Therefore, the generated conformations are refined (added the removed atomistic detail) prior to analysis. Adding atomistic

detail on a conformation is computationally expensive, as the energy function employed has to handle a large number of atoms per conformation (that includes all side-chain atoms and all hydrogen atoms per amino acid). In addition, a lot of the conformation in the generated ensemble are similar in spatial arrangement and are indicators that the conformation ensemble generation algorithms often sample from similar regions in the landscape rather than avoiding already explored spaces.

The dissertation addresses the following major questions by employing evolutionary computation techniques. The corresponding contributions are also discussed below each question.

- To generate better ensembles that contain functionally-relevant conformations, how can we mitigate the limitations of the energy functions?
  - To address this question, we first develop a hybrid/memetic multi-objective EA that decomposes the energy function into multiple objectives and balances those objectives utilizing Pareto dominance and non-dominated sorting to select conformations that survive for the next generation. Unlike the other attempts for multi-objective EAs in conformation ensemble generation, it does not require an archive of solutions and it does not use the total energy of a conformation as a basis for selection at all which defeats the purpose of decomposing the energy function in the first place. Section 4.1 describes this approach in detail. This approach achieves good results but begs the question, "can we do better using information other than energy as objective?"

    Consequently, we explore utilizing sequence-predicted contact information. Previous approaches that attempt to utilize such information either completely replace the energy function with a contact-based scoring one, or devise a new, aggregate scoring function that adds derived contacts as restraints in a new term added to an energy function. While it is instructive to determine the separate impact of energy versus contact-based scoring in improving generated conformation ensemble, aggregation of terms into a pseudo-energy function may be

problematic and may result in an overly rugged search space with many sub-optimal minima. Aggregation is additionally problematic, as one cannot know a priori the relative importance of one scoring function over the other. Instead, optimization research advocates for keeping the various scoring criteria as separate optimization objectives, which not only avoids introducing unnecessary parameterization (term weights), but also is shown to lead to better and diverse optima [39]. We address this question by investigating the separate and combined roles and guidance of energy-based and sequence-predicted contact-based scoring. What makes this possible is our ability to leverage single- and multi-objective EAs as vehicles that intrinsically allow a variety of combinations and optimization settings. The details are presented in Section 4.2.

- Can the size of the ensemble generated by the conformation ensemble generation algorithms be reduced to improve feasibility of such algorithms?

  - To address this, we first demonstrate that a reduced size conformation ensemble can represent the originally generated conformation ensemble. We do so via a clustering-based approach to significantly reduce the size of the generated ensemble without sacrificing conformation quality. We first focus on representative clustering algorithms and conduct a rigorous analysis to determine the optimal settings for these algorithms. We then cluster the conformations based on their shape similarity and select conformations from the clusters to populate a reduced ensemble that contains similar quality conformations as the original conformation ensemble. This is presented in detail in Section 5.1. The success of this work prompts the question, "how can we get a conformation ensemble generation algorithm to generate such a reduced ensemble?"

    We do this by equipping a conformation ensemble generation algorithm with an evolving map of the conformation space it explores. The map utilizes low-dimensional representation of protein conformation and serves as a memory with

controllable granularity. The map has a considerably small storage requirement but provides similar quality of a map that would hold all the conformations ever generated by an algorithm. Section 5.2 provides the details for this approach. We then ask the question, "can we guide the conformation ensemble generation algorithm with the reduced-size map at the same time to enhance conformational sampling?"

We address this question in Section 5.3 where we guide the conformation ensemble generation algorithm by consulting the reduced-size map while selecting the regions in the landscape to explore. The idea is to search the unexplored parts of the landscape and avoid regions that are already explored.

- In the multimodal energy landscape of protein conformations, how to balance the exploration and exploitation of the landscape to explicitly sample diverse minima?
  - We address this question by first focusing on mapping the multimodal energy landscape by retaining diversity of the solutions/conformations in order to identify the diverse minima that correspond to different biologically-active conformations. The idea is to preserve the niches in the landscape by dividing the population of conformations into multiple explicit stable subpopulations where each subpopulation occupies a niche and is responsible for seeking solutions in the subspace around its niche. The diversity introduced through different subpopulations helps the exploration component of the search. The exploitation comes from the evolutionary process within a subpopulation and the competition for resources between the subpopulations. This helps the resulting EA to evolve and maintain multiple subpopulations at local minima while exploring new regions of the fitness/energy landscape. Section 6.1 provides the details for this approach. This approach samples lower energy conformations than existing conformation ensemble generation algorithms but does not perform significantly

better on proximity to the ground truths/native conformations in our evaluations. Therefore, we ask ourselves, how can we do better?

We investigate this by designing an adaptive algorithm that aims to tune its behavior towards exploration or exploitation as needed via an adaptive mechanism to obtain a better balance between exploration and exploitation. We demonstrate how selection pressure is useful for this purpose and present an adaptive EA that adjusts the EA selection pressure on the fly to properly control exploration and exploitation components of the search based on the characteristics of the population of conformations. This work is described in Section 6.2.

In brief, we look to tackle two primary challenges to sample diverse minima in the energy landscape and generate better ensembles that contain multiple near-native conformations. One is to subdue the shortcomings of energy functions and the other is achieving a proper balance between exploration and exploitation. We further tackle the problem of improving the efficiency and practical use of such conformation ensemble generation algorithms. Evaluations of our techniques mostly focus on proteins where one native conformation is known mainly because of the universal use of such evaluations in the community and the richness of both computational and experimental data in it. However, in Chapter 6, we construct and evaluate on a benchmark dataset with proteins where at least two native conformations are known to measure our ability to find multiple functionally-relevant conformations. The dataset that we present here will help researchers to further advance work on this problem and is another contribution of this thesis.

Chapter 2: Background and Related Work

The three main advances for de novo protein conformation ensemble generation can be categorized along what we refer to as *representation*, *sampling*, and *scoring*. We discuss these advances below.

2.1 Representation

A conformation refers to an assignment of values to underlying parameters representing a spatial arrangement of the chain of amino acids of a protein. Perhaps the most revolutionizing progress in protein conformation ensemble generation was due to the conformational representation of a protein structure as a series of fragment configurations [50] that serve to discretize the conformation space available to a chain of amino acids and simplify the computation of novel conformations as a fragment assembly process. In this process, also known as molecular fragment replacement, known native conformations of proteins in the PDB are excised into short fragments of covalently-bound amino acids of length f, and the resulting fragment configurations are organized in a fragment configuration library indexed by fragment amino-acid sequences. What is stored for each fragment are the Cartesian coordinates of backbone atoms or the torsion/dihedral angles that can be defined over bonds connecting consecutive backbone atoms in a fragment.

Molecular fragment replacement can be used to introduce variation in a given conformation as follows: an amino acid index i is selected at random over $[1, l - f + 1]$, where l is the number of amino acids in a given protein sequence, and f is the length of fragments in the pre-compiled fragment library. The configuration of the fragment composed of amino acids $[i, i + f - 1]$ in the given conformation is then replaced with a fragment configuration selected at random among those available for the fragment with the same or similar

amino-acid sequence in the library. This replacement can be considered a "move" in a local search technique, and bias can be introduced to obtain better-scoring conformation via an iterative process.

The conformation space is also simplified and reduced in dimensionality through a coarse-grained/centroid representation. The atoms of the side chain in each amino acid are compressed into a pseudo-atom, and the conformation variables are dihedral angles (ϕ, ψ, and ω) on bonds connecting modeled backbone atoms and side-chain pseudo-atoms. Note that even this representation yields hundreds of dihedral angles (thus, a conformation space of hundreds of dimensions) even for protein chains not exceeding 150 amino acids.

Fragement lengths employed in Rosetta are 9 and 3 [30]. Many conformation ensemble generation methods follow similar coarse-grained representations and fragment-based assembly as Rosetta [40, 41]. Quark conformation sampling algorithm makes use of longer fragments [38]. AlphaFold [47] is a single conformation prediction method that augments fragment configuration libraries with novel fragments generated from a generative recurrent neural network; according to the AlphaFold team, the novel fragments contributed significantly to the team's superior performance in CASP13.

2.2 Sampling

Other variations among the de novo conformation ensemble generation methods arise in the actual sampling algorithm employed. Among the dominant algorithms, Rosetta and Quark use Simulated Annealing Metropolis Monte Carlo (SA-MMC) and produce a single conformation on one run. These methods are run multiple times on a protein sequence to obtain an ensemble of conformations. Other works use single-objective or multi-objective EAs to generate an ensemble and enhance sampling over SA-MMC methods [41, 46, 51–53]. We briefly describe Rosetta as a representative SA-MMC based algorithm and HEA [41] as a representative EA below. The choice of Rosetta reflects the fact that Rosetta is considered a benchmark conformational sampling algorithm in the literature and a lot of the evaluations

in the literature and in this dissertation involve comparison with Rosetta. HEA contains the basic evolutionary operators and a lot of the techniques in this dissertation either build over HEA or adopt a few of its operators.

2.2.1 Rosetta Conformation Sampling Algorithm

Rosetta conformation sampling algorithm operates over 4 substages. Each substage is a single trajectory MMC search and the final conformation found on each substage is used as the starting conformation for the next substage. Each move in the MMC search is a Molecular Fragment Replacement. Substage 1 first constructs an extended chain from the amino-acid sequence by setting the backbone dihedral angles to characteristic values. It then uses $2,000$ MMC moves of fragment length 9 and Rosetta $score0$ score function from Rosetta energy function suite to evaluate each move. Substage 2 uses the same fragment length and runs for the same number of moves, but uses $score1$ energy function for evaluation. Substage 3 runs for $20,000$ moves, uses $score2$ energy function and also has the identical fragment length. Substage 4, however, switches to fragment length 3 and uses $score3$ energy function. This substage is run for $12,000$ moves to optimize the conformation at the coarse-grained level.

In an MMC search, each move (a molecular fragment replacement in this context) is accepted with a probability given by the Metropolis Criterion, $p = \exp(-\delta \mathrm{E}/\alpha)$, where δE is the difference in energy from the proposed to the current conformation, and α is a unitless parameter mimicking temperature and serving to scale the change in energy. In Rosetta, to avoid getting stuck and allow exploring a minimum, all the substages use a variable temperature scheme. Whenever a number of successive moves fail, temperature is increased. The initial value of α is 1. α increases by 1 after every 150 successive failures and resets to 1 when a move is accepted. Combining each substage, a total of $36,000$ moves are performed that results in the same number of energy evaluations. Rosetta uses multi-start or random-restart to obtain an ensemble of conformations.

2.2.2 HEA

HEA is a hybrid, population-based EA. As other population-based EAs, it evolves a population of individuals (conformations in our case) over a number of generations. The population is initialized via an initialization operator described below. In HEA, all individuals in the population are selected to serve as parents. Each parent produces an offspring via a variation operator also described below. Following the principle of natural selection, the parents and offspring compete for survival. The HEA uses a fixed-size population; that is, out of N parents and N offspring, only N individuals survive in the population for the next generation. What makes the employed HEA hybrid is its employment of an improvement operator to improve offspring before they compete with parents. This operator is also described below.

Initial Population Operator: The initial population operator is first invoked on a given amino-acid sequence of a target protein to obtain the initial population. The operator first constructs p identical extended chains by setting the backbone dihedral angles to characteristic values; p is the size of the population. The extended chains are then randomized via two consecutive stages of local search. Each one is implemented as an MMC, but the stages use different scoring functions and different values for the scaling parameter α that controls the acceptance probability in the Metropolis criterion. In both stages, each move is a fragment replacement of length 9. The first stage seeks to resolve steric clashes (self collisions) and so employs the Rosetta $score0$ scoring function that encourages steric repulsion. This stage is greedy and performs 200 moves on each extended chain. The second stage employs the Rosetta $score1$ to encourage the formation of secondary structures and uses the Metropolis criterion by setting the scaling parameter α to 2. The stage continues until l consecutive moves (l is number of amino acids in a given protein sequence) fail per the Metropolis criterion.

Variation and Improvement Operators: The variation operator utilizes molecular fragment replacement with a fragment length of 3 amino acids to introduce a small conformational change over a parent in order to obtain an offspring. The obtained offspring is then subjected to an improvement operator. The operator seeks to improve the quality of an offspring as measured by the interaction energy, evaluated with the Rosetta *score3* scoring function. The operator implements a greedy local search that accepts only configuration replacements of randomly chosen fragments in the offspring (of length $k = 3$) that improve the score (lower energy) until l consecutive moves fail; l is number of amino acids. The goal of the improvement operator is to map an offspring to a nearby local minimum in the energy surface. The improved offspring has a better chance of survival when competing against parents[39, 40].

Selection Operator: The selection operator implements what is known as elitism-based truncation selection. Essentially, all individuals (parents and improved offspring) are first evaluated using Rosetta's full centroid scoring function *score*4. Then, the top-scoring $r\%$ individuals from the parents are combined with the improved offspring to compete for survival; r is the elitism rate. The competing individuals are sorted in increasing order of their *score*4, and the top p individuals are selected to represent the population for the next generation.

Termination Criterion: As is common among conformational sampling algorithms [54], the termination criterion for the HEA is set to a total budget of energy/score evaluations. When the budget is exhausted, the algorithm terminates. This is more reasonable over setting an arbitrary number of generations, as it recognizes the fact that the energy evaluation of a conformation is the most computationally-extensive operation in a conformational sampling algorithm. This termination criterion also allows for a fair comparison among conformation ensemble generation algorithms realizing different stochastic search/optimization frameworks.

2.3 Scoring

Conformational sampling methods also vary in the scoring function employed to bias sampling. As mentioned in Chapter 1, energy functions are often an inadequate proxy of nativeness. Hence, a growing body of research is considering alternative or additional scoring functions based on predicted torsion angles, secondary structures, and/or contacts. For instance, in AlphaFold, a scoring function based on pairwise amino-acid distances learned over known native conformations guides a gradient descent algorithm (that makes use of the expanded fragment library) towards a near-native conformation [47]. There is now renewed interest in employing predicted distances or contacts (which record whether two amino acids are spatially proximal or not based on a characteristic distance threshold of 8Å between the beta carbon atoms of two amino acids) as alternative or additional scoring functions in conformational sampling methods.

The growth in known native conformations of proteins has allowed data-driven methods to predict contacts with increasing accuracy; some methods use evolutionary coupling analysis [55], and others leverage supervised machine learning [56] over a wide variety of features. We note that the top performer in the CASP12 contact prediction category was RaptorX-Contact [57], which employs a deep residual neural network for contact prediction. However, it remains unclear how to best exploit steady improvements in contact prediction for improved conformational sampling [58]. Research is active and existing approaches either completely replace the energy function with a contact-based scoring one, or devise a new, aggregate scoring function that adds derived contacts as restraints in a new term added to an energy function [53, 58–62]. It also remains unclear how many contacts are needed for improved performance, as not all contacts are predicted with the same confidence from a machine learning model. When evaluating contact prediction methods, it is common practice to consider a reduced list of the most confident 10, $l/5$, or $l/2$ contacts, where l is the number of amino acids in a target sequence [58]. Various contact-based scores are devised. Precision, Recall, F1, and Coverage are some of the most commonly used ones for evaluation

in the CASP contact assessment category. Due to the fact that these measures are highly correlated on the reduced lists, precision is typically the primary choice for evaluation.

Chapter 3: Employed Domain-specific Knowledge and Evaluation

3.1 Representation

A conformation is the result of instantiating variables selected to represent the spatial arrangement of a tertiary structure. The dissertation employs Rosetta's centroid (CEN) representation for a conformation. In this representation, the side-chain of each amino acid is first reduced to a pseudo-atom, which marks the location of the side-chain and interpreted as the centroid over the side-chain atoms. The only atoms that are explicitly modeled for each amino acid are the heavy backbone atoms and the centroid of the side-chain atoms. The underlying variables modeled are the ϕ, ψ, ω dihedral backbone angles for each amino acid which basically determine the spatial arrangement of atoms that are covalently linked to form a chain that folds in different ways in three dimensions. This representation simplifies the search space and reduces its dimensionality compared to the coordinate representation where each atom in each amino-acid is modeled with the Cartesian coordinates. Through forward kinematics, we can go from this dihedral conformation representation to the coordinate representation.

In addition, the molecular fragment replacement technique is used to discretize the search space and add variation to a conformation by bundling the backbone dihedral angles together, as described in 2.1. We use the popular online Robetta fragment configuration library [30] which provides fragment configurations of length 3 and 9 given an amino-acid sequence of a protein. The provided fragments are organized in such a way that a query with the amino-acid sequence of a fragment returns 200 fragment configurations to choose from.

3.2 Energy functions

The conformation space available to a given amino-acid sequence is vast and high-dimensional. Some back-of-the-envelope calculations provide the context. Consider a short protein sequence of 60 amino acids. This results in 178 backbone dihedral angles (3 – ϕ, ψ, and ω – angles for each amino acid, save for the first and the last one that contain two angles each), thus giving rise to a 178-dimensional conformation space. Not all conformations correspond to energetically-favorable states. Some conformations are clearly unfavorable, containing steric clashes among portions of the chain. Others are not energetically-favorable for a variety of reasons, captured in a scoring function. The Rosetta suite of energy/scoring functions provides a variety of energy functions for a conformation that consider different energetic terms to calculate the energy scores. We use *score*0, *score*1, *score*3, and *score*4 energy functions that work with the centroid representation for the experiments in this thesis. The *score*0 energy function considers only the *Van der Waals* (*vdw*) energetic term which penalize steric clashes. *score*1 additionally considers five more energetic terms and rewards secondary structure formation. *score*3 adds four more energetic terms over *score*1 and rewards compact tertiary structure formation. *score*4 is the full centroid scoring function that additionally considers short-range hydrogen bonding, long-range hydrogen bonding, and chainbreak terms along with the Ramachandran score which assigns probability-based scores for residues based on the dihedral angles. The 14 energetic terms and their weight values for each of these energy functions are provided in Table 3.1.

3.3 Evaluation Datasets

The algorithms and techniques presented in this thesis are mainly evaluated on two monomorphic datasets that contain proteins with one known native conformation mostly because evaluation in such datasets is the norm in the literature for this problem. The first is a benchmark dataset of 20 target proteins of varying lengths (ranging from 53 to 146 amino acids) and folds (α, β, $\alpha + \beta$, and *coil*), listed in Table 3.2 by their PDB IDs.

Table 3.1: The 14 energetic terms considered in the Rosetta's centroid energy functions and their weight values for each energy functions used.

	Energy Function			
Energetic Term	score0	score1	score3	score4
environment (env)	0	1.0	1.0	1.0
residue pair (pair)	0	1.0	1.0	1.0
cbeta	0	0	1.0	1.0
Van der Waals forces (vdw)	0.1	1.0	1.0	1.0
radius of gyration (rg)	0	0	3.0	2.0
cenpack	0	0	1.0	1.0
helices-strands pair (hs_pair)	0	1.0	1.0	1.0
strand-strand pair (ss_pair)	0	0.3	1.0	1.0
rsigma	0	0	1.0	1.0
beta sheet formation (sheet)	0	1.0	1.0	1.0
long-range hydrogen bonding (hbond_lr_bb)	0	0	0	1.0
short-range hydrogen bonding (hbond_sr_bb)	0	0	0	1.0
Ramachandran score (rama)	0	0	0	1.0
chainbreak	0	0	0	1.0

This dataset was introduced in [63] and then complemented with more targets in later work [34, 41, 46, 52, 53, 64]. The second dataset consists of 10 hard, free-modeling target domains from CASP12 and CASP13, listed in Table 3.3 by their domain IDs. However, we expand our evaluation on Section 6.2 to include a third dataset, one that is metamorphic or consists of proteins with at least two known native conformations. We compiled this novel dataset of 13 proteins from various works [18, 65]. The dataset consists mostly of proteins with two known native conformations. Table 3.4 relates the dataset. The first 12 rows relate proteins where wet-laboratories have elucidated two very distinct conformations. The pairwise distance of the conformations are related in Column 4 in the form of lRMSD (described in Section 3.4). The last row relates Calmodulin, for which 4 distinct conformations are obtained from the PDB. The range of pairwise lRMSD is shown in this case.

Table 3.2: Targets in the Benchmark dataset.

PDB ID	**Length**	**Fold**
1ail	73	α
1aly	146	β
1aoy	78	α
1bq9	53	β
1c8c(A)	64	β
1cc5	83	α
1dtd(B)	61	$\alpha + \beta$
1dtj(A)	76	$\alpha + \beta$
1fwp	68	$\alpha + \beta$
1hhp	99	β
1hz6(A)	67	$\alpha + \beta$
1isu(A)	62	*coil*
1sap	66	β
1tig	88	$\alpha + \beta$
1wap(A)	68	β
2ci2	83	$\alpha + \beta$
2ezk	93	α
2h5n(D)	123	α
2hg6	106	$\alpha + \beta$
3gwl	106	β

Table 3.3: Targets in the CASP dataset.

Domain ID	**Length**	**CASP**
T0859-D1	129	12
T0886-D1	69	12
T0892-D2	110	12
T0897-D1	138	12
T0898-D2	55	12
T0953s1-D1	67	13
T0953s2-D3	93	13
T0957s1-D1	108	13
T0960-D2	84	13
T1008-D1	77	13

Table 3.4: Targets in the Metamorphic dataset

Protein Name	Length	PDB Ids of Known Conformations	lRMSD (Å)
SARA	127	1fzp(D), 2frh(A)	19
Calcium-bound EF-Hand protein	134	1jfk(A), 2nxq(B)	15.9
Yeast Matalpha2/MCM1	87	1mnm(C), 1mnm(D)	6.7
IscA	112	1x0g(A), 1x0g(B)	18
NF-kB RelB	110	1zk9(A), 3jv6(A)	16.5
Beta 2 Microglobulin	100	3low(A), 3m1b(F)	19.3
Protein Related to DAN and Cerberus	148	4jph(B), 5hk5(H)	6.9
Methanocaldococcus monomeric selecase	110	4qhf(A), 4qhh(A)	12.2
CopK	74	2k0q(A), 2lel(A)	9.1
SLAS-micelle bound alpha-synuclein	140	2kkw(A), 2n0a(D)	36.1
Human prion protein mutant HuPrP	147	2lej(A), 2lv1(A)	18.6
Cyanovirin-N	101	2ezm(A), 1l5e(A)	16
Calmodulin	148	1cfd(A), 1cll(A), 2f3y(A), 1lin(A)	4.3-13.4

3.4 Evaluation Metrics

In this dissertation, we measure the performance of a conformation ensemble generation algorithm by the lowest reached energy and the lowest reached distance to the known native conformation of the target sequence under consideration, as is practice in evaluations of EAs for conformational sampling [54]. The first provides information on the capability of an algorithm to explore the vast conformation space and the underlying energy surface of a given protein sequence. The second provides information on the ability of an algorithm to get to near-native regions of the space. Measuring the distance to the known native conformation is important because lower energies do not necessarily correlate with proximity to the native conformation. For the lowest reached energy, we use Rosetta *score*4 energy as in [40,41,46,66]. In conformational sampling, the proximity comparisons typically focus on the main carbon atoms or the Cα atoms. We measure the proximity to the native conformation via three popular metrics, least root-mean-squared-deviation (lRMSD) [67], Template Modeling Score (TM-score) [68,69], and Global Distance Test - Total Score (GDT_TS) [70].

We use lRMSD to perform evaluations in all the datasets as it is widely used in template-free PSP. We employ TM-score and GDT_TS for evaluations in the CASP dataset as they are standard similarity measures used in CASP competitions. We provide brief descriptions of these proximity metrics below.

lRMSD: RMSD is a dissimilarity metric based on Euclidean distance between two conformations. After a generated conformation and a given native conformation are optimally superimposed to remove differences due to rigid-body motions in 3D (rotations and translations), RMSD measures the Euclidean distance averaged over the atoms under comparison. If N is a given native conformation and S is a generated conformation, both containing M atoms, RMSD between them is given by, $\sqrt{\sum_{j=1}^{M} |P_j(N) - P_j(S)|^2}/M$, where $P_j(X)$ is the position of atom j in conformation X. The "least" term in lRMSD indicates that the conformations are optimally aligned to provide the lowest RMSD between the conformations. lRMSD is measured in Å; a lower score indicates a better proximity.

TM-score: TM-score is a similarity metric that weights shorter distances between corresponding residues of two conformations stronger than the longer distances. The goal is to achieve more sensitivity towards the global fold similarity than to the local conformation deviations. TM-score is given by, $Max[\frac{1}{L(N)} \sum_{j=1}^{L(A)} \frac{1}{1+(\frac{D_j}{D_0})^2}]$, where $L(N)$ and $L(A)$ are the lengths of the native conformation and the length of the aligned residues respectively. D_j is the distance between the j-th pair of residues and D_0 is a scaling factor that normalizes distances. Max denotes the maximum value after optimal superposition. TM-score provides a score in $[0, 1]$ with a higher score indicating a better proximity.

GDT_TS: GDT_TS is a similarity metric that utilizes 4 different distance thresholds. After superimposing two conformations, it measures the average of the largest set (as a percentage) of amino-acid's alpha carbon atoms in a native conformation falling within defined distance thresholds of their position in the generated conformation. GDT_TS is given

by, $(GDT\_P_1 + GDT\_P_2 + GDT\_P_4 + GDT\_P_8)/4$, where GDT_P_i denotes the percentage of residues under distance threshold iÅ. GDT_TS provides a score in $[0, 1]$, which is often interpreted as a percentage, with a higher score indicating a better proximity.

The comparative evaluations we relate in this dissertation are further strengthened by statistical significance tests. We use Fisher's [71] and Barnard's [72] exact tests over 2x2 contingency matrices keeping track of the particular performance metric under comparison. Fisher's exact test is conditional and widely adopted for statistical significance. Barnard's test is unconditional and generally considered more powerful than Fisher's test on 2x2 contingency matrices.

Finally, to provide a complete picture and measure how much better or worse performance is achieved on each target, we also employ performance profiles [73] for our works presented in Sections 5.3 and 6.2. Performance profiles show the cumulative distribution functions for different performance ratios for a evaluation metric that reveal major performance characteristics. Let us briefly summarize the concept of performance profiles, as they have never been employed in protein modeling research to the best of our knowledge. Performance profiles provide us with a way of depicting how frequently a particular algorithm is within some distance of the best algorithm for a particular problem instance/target. So, for each problem instance, we first compute the best method, and then for every other method, we determine how far they are from optimal. We vary the *performance ratio* (pr) over a range for this analysis. Specifically, for a given pr, *measure reached* means that an algorithm comes within a factor of pr of the best measure over all algorithms on a given target. The number of targets where an algorithm does this is tallied up, and this becomes indicative of its performance, also referred to as number of problems *solved*, at a given performance ratio. In our case, problem instances are our targets in the dataset in consideration.

Chapter 4: Mitigating Energy Function Limitations

As stated in Chapter 1, even state-of-the-art energy functions that quantify atomic interactions in a conformation are inherently inaccurate; they result in overly rugged energy surfaces (associated with protein conformation spaces) that are riddled with artifact local minima. In this chapter, we explore ways to mitigate the effects these inaccuracies have on the optimization process. First, we present our work on properly balancing multiple energetic objectives to generate better quality conformation ensembles in Section 4.1. Then, we explore utilizing sequence-predicted contact information as an additional optimization objective in Section 4.2. The algorithms we present below take the amino-acid sequence of a protein as input and provide an ensemble of conformations generated through the evolutionary process as output.

4.1 Balancing Multiple Objectives in Conformation Sampling

The work presented in this section has been published in [74]. Decomposing the energy function into multiple energetic objectives and optimizing these objectives together has been shown to generate better quality conformation ensemble than methods that optimize a single objective considering the energy function as a whole [41, 44–46]. In this work, we explore how to achieve a proper balance between multiple competing energetic objectives to ensure a diverse set of sampled conformation. To do this, we develop a multi-objective EA to directly control the diversity of the sampled conformation. We refer to the algorithm as Evo-Diverse. Evo-diverse balances the multiple objectives in a way that results in high exploration capability and is additionally able to access lower-energy regions of the energy landscape of a given protein with similar or better proximity to the known native conformation than state-of-the-art algorithms. Unlike existing, state-of-the-art multi-objective EAs,

the proposed algorithm circumvents issues related to the usage of an archive (described later), thus saving computational overhead, and avoids the usage of total energy in its optimization objectives altogether.

4.1.1 Summary of Evo-Diverse

Evo-diverse is a memetic EA that controls the diversity of the conformations it computes via the selection operator that determines individual survival. The algorithm builds over expertise in our laboratory on EAs for conformation sampling. Evo-diverse evolves a fixed-size population of N conformations over generations. The initial population is constructed as in HEA (described in Section 2.2.2). at the beginning of each generation, all individuals in the population are selected as parents and varied so that each yields one offspring conformation. To additionally improve exploitation (digging deeper into the energy surface), each offspring is further subjected to an improvement operator. The variation and improvement operators are employed as described in Section 2.2.2. After applying the variation and improvement operators, the algorithm has now computed N new (offspring) conformations that will fight for survival among one another and the N parent conformations. The winners constitute the population for the next generation.

4.1.2 Selection Operator

The selection operator is the mechanism leveraged to pursue a multi-objective optimization setting and directly control the diversity of computed conformations. We first describe how the selection operator allows a multi-objective optimization setting.

Multi-objective Optimization under Pareto Dominance

Let us consider that a certain number of optimization objectives is provided along which to compare conformations. A conformation C_a is said to *dominate* another conformation C_b if the value of each optimization objective in C_a is lower than the value of that same objective in C_b; this is known as strong dominance. If equality is allowed, the result is

soft dominance. The proposed algorithm makes use of strong dominance. Utilizing the concept of dominance, one can measure the number of conformations that dominate a given conformation C_b. This measure is known as *Pareto rank* (PR) or, equivalently, *domination count*. In contrast, the number of conformations dominated by a given conformation C_a is known as the *Pareto count* (PC) of C_a. If no conformation in a set dominates a given conformation C_b, then C_b has a domination count (PR) of 0 and is said to be *non-dominated*. Non-dominated conformations constitute the *Pareto front*.

The concept of Pareto dominance can be operationalized in various ways. In early proof-of-concept work [41, 46], the Rosetta *score*4 (which includes both short-range and long-range hydrogen bonding terms) was divided into three optimization objectives along which parents and offspring can be compared in the selection operator: short-range hydrogen bonds (objective 1), long-range hydrogen bonds (objective 2), and everything else (summed together in objective 3). This categorization recognizes the importance of hydrogen bonds for formation of native conformations [35]. Using these three objectives, work in [41] utilizes only PR in the selection operator, first sorting the N parent and N offspring conformations from low to high PR, and then further sorting conformations with the same PR from low to high *score*4 (total energy that sums all three objectives). PC can be additionally considered to obtain a sorted order, as in [46]. Conformations with the same PR are sorted from high to low PC, and conformations with the same PC are further sorted from low to high *score*4. The selection operator then selects the top N conformations (out of the combined $2N$ conformations of parents and offspring) according to the resulting sorted order.

Non-dominated Fronts Our algorithm truly considers a multi-objective setting and does not utilize an aggregate energy value (the sum of the objectives). Specifically, the algorithm considers non-dominated fronts in its selection operator. A fast, non-dominated sorting algorithm (originally proposed in [75]) is used to generate these fronts as follows. All the conformations in the combined parent and offspring population that have a domination count of 0 (thus, are non-dominated) make up the first non-dominated front F_1. Each

subsequent, non-dominated front F_i is generated as follows. For each conformation $C \in F_{i-1}$, the conformations dominated by C constitute the set S_C. The domination count of each member in S_C is decremented by 1. Conformations in S_C that have their domination count reduced to 0 make up the subsequent, non-dominated front F_i. This process of generating non-dominated fronts terminates when the total number of conformations over the generated fronts equals or exceeds the population size N. In this way, the selection operator is accumulating enough good-quality conformations from which it can further draw based on additional non-energy based objectives. Moreover, this allows generating Pareto-optimal solutions over the generations and achieving better convergence to the true, Pareto-optimal set.

Density-based Conformation Diversity

Borrowing from evolutionary computation research [75] on optimization problems of few variables ranging from 1 to 30 (as opposed to hundreds of variables in our setting), we leverage crowding distance to retain diverse conformations. Crowding distance estimates the density of the conformations in the population space and guides the selection process over generations towards less crowded regions [75]. We use the crowding distance assignment technique to compute the average distance of a conformation from other conformations in the same non-dominated front along each of the optimization objectives. First, the crowding distance of each conformation is initialized to 0. Then, for each objective, conformations are sorted based on their corresponding score (value of that objective) in ascending order and assigned infinite distance value to conformations with the highest and lowest scores; this ensures that conformations with the highest and lowest scores (effectively constituting the boundaries of the population space) are always selected. For all other conformations C, the absolute normalized difference in scores between the two closest conformations on either side of C is added to the crowding distance. Finally, when all the objectives are considered, the crowding distance of a conformation is the sum of the individual distances along each objective.

Putting it All Together: Conformation Diversity in a Multi-objective Optimization Setting

To obtain the next population, the selection operator selects r conformations from the non-dominated fronts $F_1, F_2, \ldots, F_t$ sequentially, where r is $\sum_{i \in \{1,2,\ldots,t\}} F_i$ until $r + |F_{t+1}|$ reaches or exceeds N. If $r < N$, which is usually the case, the crowding distance of conformations in F_{t+1} is computed and used to sort them in descending order. The selection operator then selects the top $N - r$ conformations in this order.

It is worth noting that in our lab's earlier operationalizations [41, 46] of multi-objective optimization for template-free PSP, all conformations ever computed were retained to form an archive for the calculation of PR and PC values for each conformation. This introduces a significant computational overhead, which the proposed algorithm circumvents. The proposed algorithm instead uses only the current combined population of parents and offspring to perform selection, thus saving such overhead.

4.1.3 Implementation Details

The population size is $N = 100$ conformations, in keeping with earlier work on multi-objective EAs. Instead of imposing a bound on the number of generations, the algorithm is executed for a fixed budget of $10,000,000$ energy evaluations. The algorithm is implemented in Python and interfaces with the PyRosetta library. The algorithm takes $1-4$ hours on one Intel Xeon E5-2670 CPU with 2.6GHz base processing speed and 64GB of RAM. The range in running time depends primarily on the length of the protein. As further described in the next section, the algorithm is run 5 times on a test case (a target amino-acid sequence) to remove differences due to stochasticity.

4.1.4 Results

The evaluation is carried out on both the benchmark and the CASP datasets listed in Section 3.3. The Evo-Diverse algorithm is compared with Rosetta's conformation sampling

algorithm described in 2.2.1, a memetic EA that does not utilize multi-objective optimization [40], and two other memetic EAs that do so (one utilizing only Pareto Rank [41], and the other utilizing both Pareto Rank and Pareto Count [46], as described in Section 4.1.2). We will correspondingly refer to these algorithms as Rosetta, mEA, mEA-PR, and mEA-PR+PC. This comparison allows us to isolate the impact of the selection operator in Evo-Diverse over those in mEA-PR, and mEA-PR+PC, as well as point to the impact of the multi-objective setting (in comparison with mEA) and the evolutionary computation framework overall (in comparison with Rosetta). Each of these algorithms is run 5 times on each target sequence, and what is reported is their best performance over all 5 runs combined. Each run continues for a fixed computational budget of $10M$ energy evaluations.

In keeping with published work on EAs for conformational sampling [54], performance is measured by the lowest energy ever reached and the lowest distance ever reached to the known native conformation of a target under consideration. To carry out a principled comparison, we evaluate the statistical significance of the presented results using Fisher's and Barnard's exact tests.

Comparative Analysis on Benchmark Dataset

Fig. 4.1 shows the lowest energy obtained over combined 5 runs of mEA, mEA-PR, mEA-PR+PC, Rosetta, and Evo-Diverse for each of the 20 target proteins; the targets are denoted on the x axis by the Protein Data Bank (PDB) [48] identifier (ID) of a known native conformation for each target. Fig. 4.2 presents the comparison in terms of the lowest lRMSD achieved on each of the test cases. Color-coding is used to distinguish the algorithms from one another.

A summary of comparative observations is presented in Table 4.1. Table 4.1(a) shows that lowest energy is achieved by Evo-Diverse in 9/20 of the test cases over the other algorithms; in comparison, mEA-PR achieves the lowest energy in 4/20, mEA and mEA-PR+PC in 3/20, and Rosetta in only 1 case. In a head-to-head comparison, Evo-Diverse bests each of the other algorithms in a comparison of lowest energy. Table 4.1(b) shows that

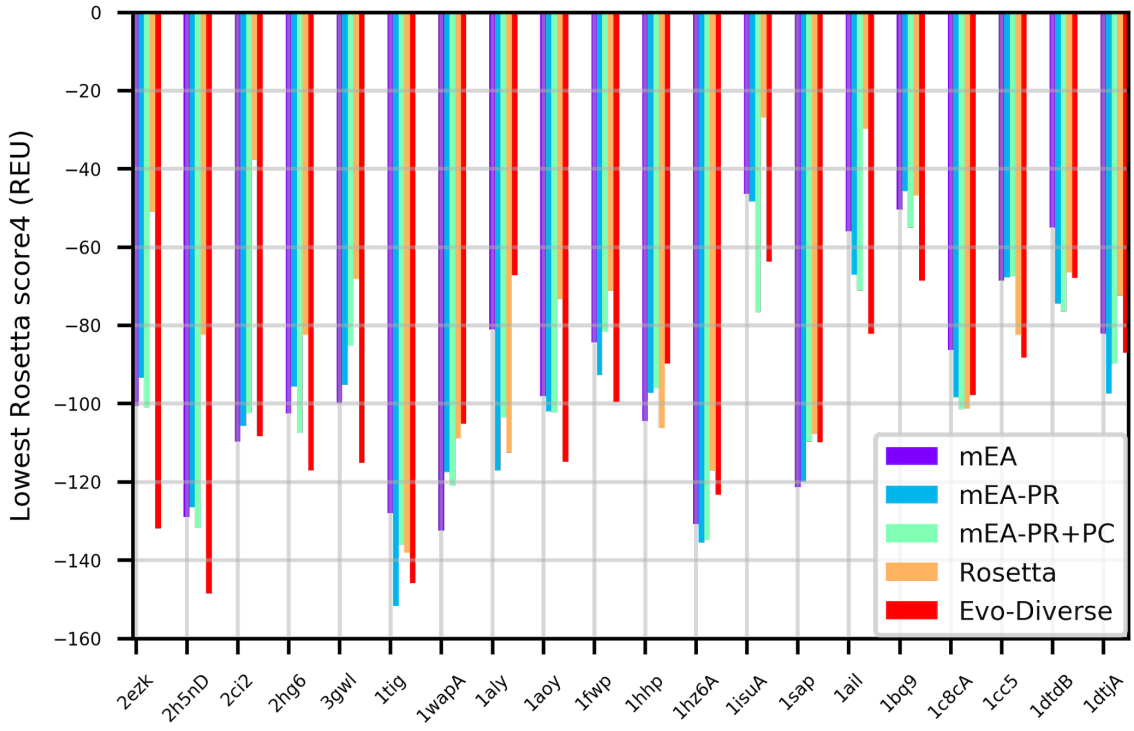


Figure 4.1: The lowest Rosetta *score*4 (measured in Rosetta Energy Units – REUs) to a given native conformation obtained over 5 runs of each algorithm on each of the 20 test cases of the benchmark dataset is shown here, using different colors to distinguish the algorithms under comparison.

lowest lRMSD is achieved by Evo-Diverse in 10/20 test cases over the other algorithms; in comparison, mEA-PR achieves the lowest energy in 2/20, mEA and mEA-PR+PC in 1/20, and Rosetta in 9 cases. In a head-to-head comparison, Evo-Diverse bests each of the other algorithms in a comparison of lowest lRMSD, as well.

The above comparisons are further strengthened via statistical analysis. Table 4.2(a) shows the p-values obtained in 1-sided statistical significance tests that pitch Evo-Diverse against each of the other algorithms (in turn), evaluating the null hypothesis that Evo-Diverse performs similarly or worse than its counterpart under comparison, considering two metrics, achieving the lowest energy in each test case, and achieving a lower (lowest)

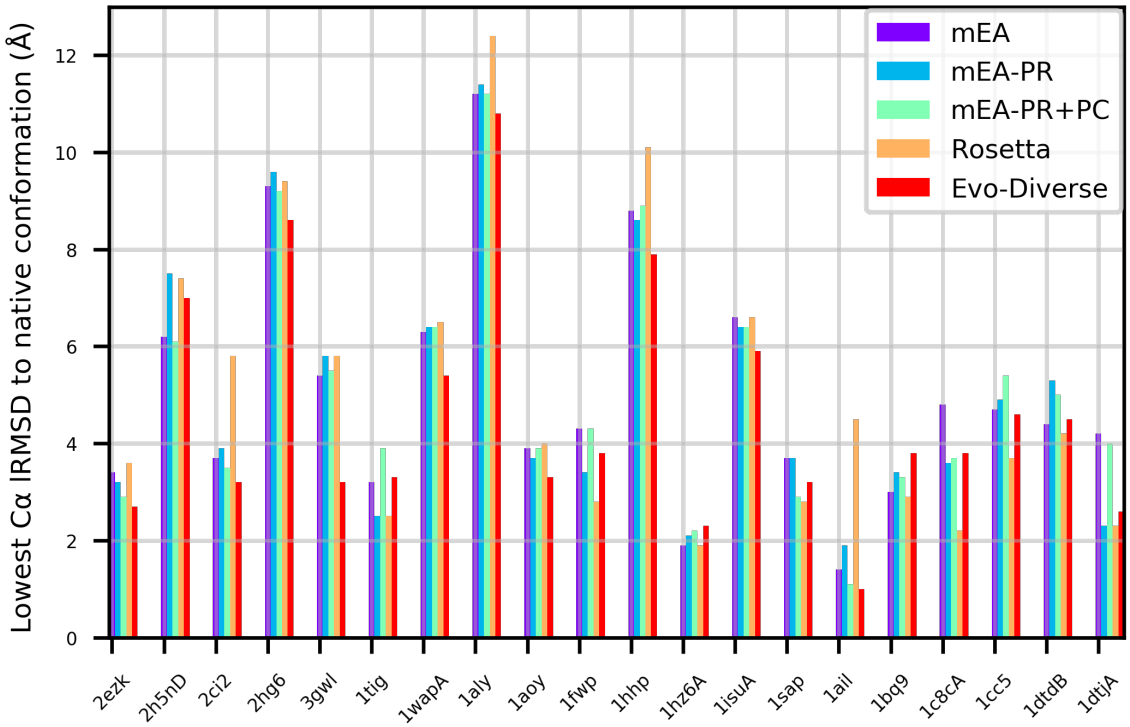


Figure 4.2: The lowest lRMSD (measured in Angstroms – Å) to a given native conformation obtained over 5 runs of each algorithm on each of the 20 test cases of the benchmark dataset is shown here, using different colors to distinguish the algorithms under comparison.

energy on each test case that its current counterpart. Both Fisher's and Barnard's test are conducted, and p-values less than 0.05 (which reject the null hypothesis) are marked in bold. Table 4.2(a) shows that the null hypothesis is rejected in most of the comparisons; Evo-Diverse performs better than mEA and Rosetta; the performance over mEA-PR and mEA-PR+PC is not statistically significant.

Table 4.2(b) shows the p-values obtained in 1-sided statistical significance tests that pitch the performance of Evo-Diverse against each of the other algorithms (in turn), evaluating the null hypothesis that Evo-Diverse performs similarly or worse than its counterpart under comparison, considering two metrics, achieving the lowest lRMSD in each test case, and achieving a lower (lowest) lRMSD on each test case than its current counterpart. Both

Table 4.1: Comparison summary of benchmark dataset.

(a) Comparison of the number of test cases of the benchmark dataset on which the algorithms achieve the lowest energy value.

Evo-Diverse vs. others: 9 vs. 3 (mEA), 4 (mEA-PR), 3 (mEA-PR+PC), and 1 (Rosetta)
Evo-Diverse vs. mEA: 14 vs. 6
Evo-Diverse vs. mEA-PR: 11 vs. 9
Evo-Diverse vs. mEA-PR+PC: 12 vs. 8
Evo-Diverse vs Rosetta: 16 vs. 4

(b) Comparison of the number of test cases of the benchmark dataset on which the algorithms achieve the lowest lRMSD value.

Evo-Diverse vs. others: 10 vs. 1 (mEA), 2 (mEA-PR), 1 (mEA-PR+PC), and 9 (Rosetta)
Evo-Diverse vs. mEA: 15 vs. 5
Evo-Diverse vs. mEA-PR: 14 vs. 6
Evo-Diverse vs. mEA-PR+PC: 15 vs. 5
Evo-Diverse vs Rosetta: 11 vs. 9

Fisher's and Barnard's test are conducted, and p-values less than 0.05 (rejecting the null hypothesis) are in bold. Table 4.2(b) shows that the null hypothesis is rejected in most tests; Evo-Diverse outperforms all algorithms except for Rosetta.

Table 4.3(a) shows the p-values obtained in 2-sided statistical significance tests that pitch Evo-Diverse against each of the other algorithms (in turn), evaluating the null hypothesis that Evo-Diverse performs similarly to its counterpart under comparison, considering two metrics, achieving the lowest energy in each test case, and achieving a lower (lowest) energy on each test case than its current counterpart. Both Fisher's and Barnard's test are conducted, and p-values less than 0.05 (which reject the null hypothesis) are marked in bold. Table 4.3(a) shows that the null hypothesis is rejected in most of the comparisons; Evo-Diverse does not perform similarly to mEA and Rosetta; the dissimilarity of performance compared to mEA-PR and mEA-PR+PC is not statistically significant at 95% confidence level. Similarly, Table 4.3(b) shows the p-values obtained in 2-sided statistical significance

Table 4.2: 1-sided statistical significance tests for the benchmark dataset.

(a) Comparison of Evo-Diverse to other algorithms on lowest energy via 1-sided Fisher's and Barnard's tests on the benchmark dataset. Top panel evaluates the null hypothesis that Evo-Diverse does not achieve the lowest energy, considering each of the other four algorithms in turn. The bottom panel evaluates the null hypothesis that Evo-Diverse does not achieve a lower lowest energy value in comparison to a particular algorithm, considering each of the four other algorithms in turn.

Best Lowest Energy

Test	mEA	mEA-PR	mEA-PR+PC	Rosetta
Fisher's	**0.04118**	0.088	**0.04118**	**0.004181**
Barnard's	**0.02489**	0.05368	**0.02489**	**0.001879**

Better Lowest Energy

Test	mEA	mEA-PR	mEA-PR+PC	Rosetta
Fisher's	**0.01282**	0.3762	0.1715	**0.00018**
Barnard's	**0.008299**	0.3179	0.1341	**0.00009139**

(b) Comparison of Evo-Diverse to other algorithms on lowest lRMSD via 1-sided Fisher's and Barnard's tests on the benchmark dataset. Top panel evaluates the null hypothesis that Evo-Diverse does not achieve the lowest lRMSD, considering each of the other four algorithms in turn. The bottom panel evaluates the null hypothesis that Evo-Diverse does not achieve a lower lowest lRMSD value in comparison to a particular algorithm, considering each of the four other algorithms in turn.

Best Lowest lRMSD

Test	mEA	mEA-PR	mEA-PR+PC	Rosetta
Fisher's	**0.001671**	**0.006907**	**0.001671**	0.5
Barnard's	**0.000702**	**0.003284**	**0.000702**	0.4373

Better Lowest lRMSD

Test	mEA	mEA-PR	mEA-PR+PC	Rosetta
Fisher's	**0.001924**	**0.01282**	**0.001924**	0.3762
Barnard's	**0.001118**	**0.008299**	**0.001118**	0.3179

tests that now consider the lowest lRMSD instead of lowest energy. Table 4.3(b) shows that the null hypothesis is rejected in most tests; Evo-Diverse does not perform similarly to all algorithms except for Rosetta at 95% confidence level.

Taken altogether, these results indicate that Evo-Diverse has a high exploration capability, decidedly outperforming mEA and Rosetta in terms of its ability to wisely use a

fixed computational budget to reach lower-energy levels, and performing similarly or better than mEA-PR and mEA-PR+PC. The latter result is not surprising, as mEA-PR, mEA-PR+PC, and Evo-Diverse use a multi-objective optimization framework, which delays a premature convergence, thus allowing them to reach lower energies within the same computational budget provided to mEA and Rosetta. Interestingly though, the head-to-head lRMSD comparisons show that, while mEA-PR and mEA-PR+PC achieve lower energies than Rosetta, this does not help them achieve the same performance as Rosetta in terms of lowest lRMSDs. In contrast, Evo-Diverse effectively retains the best of both. It is able to reach lower energies than Rosetta and comparable or lower lRMSDs than Rosetta, thus constituting a clear advantage over the current state-of-the-art multi-objective optimization EAs.

When analyzing the performance of conformation sampling algorithms, it is additionally informative to visualize the energy landscape that they probe one conformation at a time. We do so by plotting conformation-energy pairs, representing a conformation with its lowest lRMSD coordinate to the known native conformation of each test case. Fig. 4.3 and Fig. 4.4 juxtapose such landscapes for two selected test cases, the protein with known native conformation under PDB ID 1ail, and that with known native conformation under PDB ID 1dtja, respectively.

The comparison is limited here to landscapes probed by Evo-Diverse, mEA-PR, and mEA-PR+PC, as prior work [41] comparing mEA-PR and mEA-PR+PC to Rosetta and mEA shows that these two algorithms achieve better funneling (better correlation between low energies and low lRMSDs to the native conformation), and that mEA-PR+PC does so the best for 1ail, while mEA-PR does so for 1dtja.

Fig. 4.3 shows that Evo-Diverse reveals better funneling of the landscape than mEA-PR+PC (higher correlation between low energies and low lRMSDs) and multiple non-native local minima, visually confirming its high exploration capability. Fig. 4.4 shows that Evo-Diverse and mEA-PR reveal similar correlation between low energies and low lRMSDs (higher than both Rosetta and mEA) and multiple non-native local minima.

Table 4.3: 2-sided statistical significance tests for the benchmark dataset.

(a) Comparison of Evo-Diverse to other algorithms on lowest energy via 2-sided Fisher's and Barnard's tests on the benchmark dataset. Top panel evaluates the null hypothesis that Evo-Diverse achieves similar performance on reaching the lowest energy, considering each of the other four algorithms in turn. The bottom panel evaluates the null hypothesis that Evo-Diverse achieves similar performance on reaching a lower lowest energy value in comparison to a particular algorithm, considering each of the four other algorithms in turn.

Best Lowest Energy

Test	mEA	mEA-PR	mEA-PR+PC	Rosetta
Fisher's	0.08236	0.176	0.08236	**0.008362**
Barnard's	**0.04977**	0.1074	**0.04977**	**0.003759**

Better Lowest Energy

Test	mEA	mEA-PR	mEA-PR+PC	Rosetta
Fisher's	**0.02564**	0.7524	0.3431	**0.00036**
Barnard's	**0.0166**	0.6358	0.2682	**0.0001828**

(b) Comparison of Evo-Diverse to other algorithms on lowest lRMSD via 2-sided Fisher's and Barnard's tests on the benchmark dataset. Top panel evaluates the null hypothesis that Evo-Diverse achieves similar performance on reaching the lowest lRMSD, considering each of the other four algorithms in turn. The bottom panel evaluates the null hypothesis that Evo-Diverse achieves similar performance on reaching a lower lowest lRMSD value in comparison to a particular algorithm, considering each of the four other algorithms in turn.

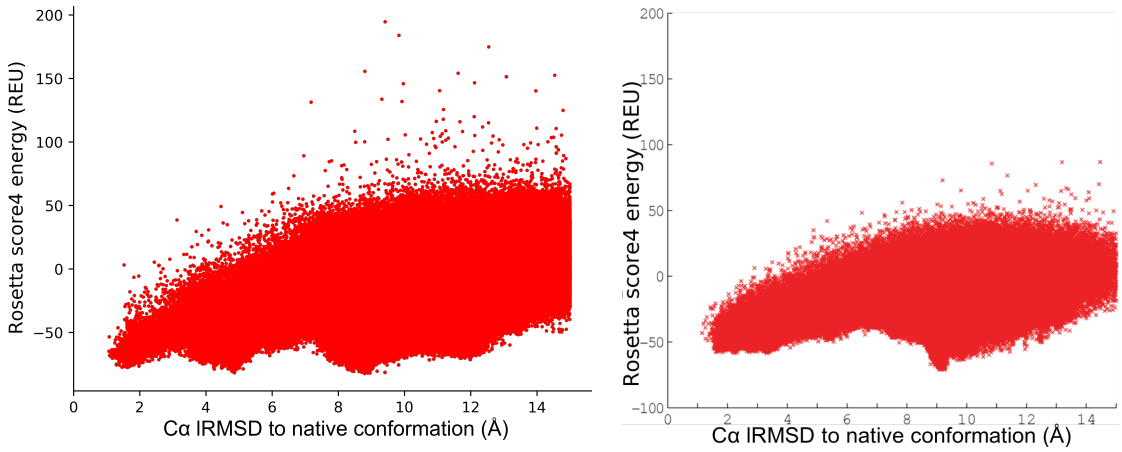
Best Lowest lRMSD

Test	mEA	mEA-PR	mEA-PR+PC	Rosetta
Fisher's	**0.003342**	**0.01381**	**0.003342**	1
Barnard's	**0.001404**	**0.006567**	**0.001404**	0.8746

Better Lowest lRMSD

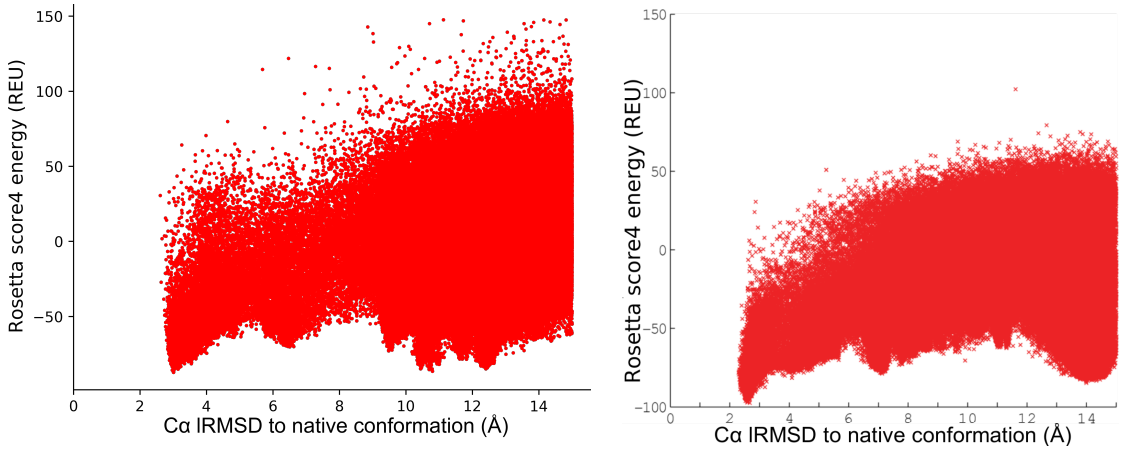
Test	mEA	mEA-PR	mEA-PR+PC	Rosetta
Fisher's	**0.003848**	**0.02564**	**0.003848**	0.7524
Barnard's	**0.002236**	**0.0166**	**0.002236**	0.6358

Fig. 4.5 superimposes the best conformation (lowest lRMSD to the known native conformation) over the known native conformation for three selected proteins (PDB IDs 1ail, 1dtja, and 3gwl). Rendering is performed with the CCP4mg molecular graphics software [1]. In the case of 1ail, Evo-Diverse obtains the lowest lRMSD to the native conformation (1Å).



(a) Evo-Diverse (1ail) (b) mEA-PR+PC (1ail)

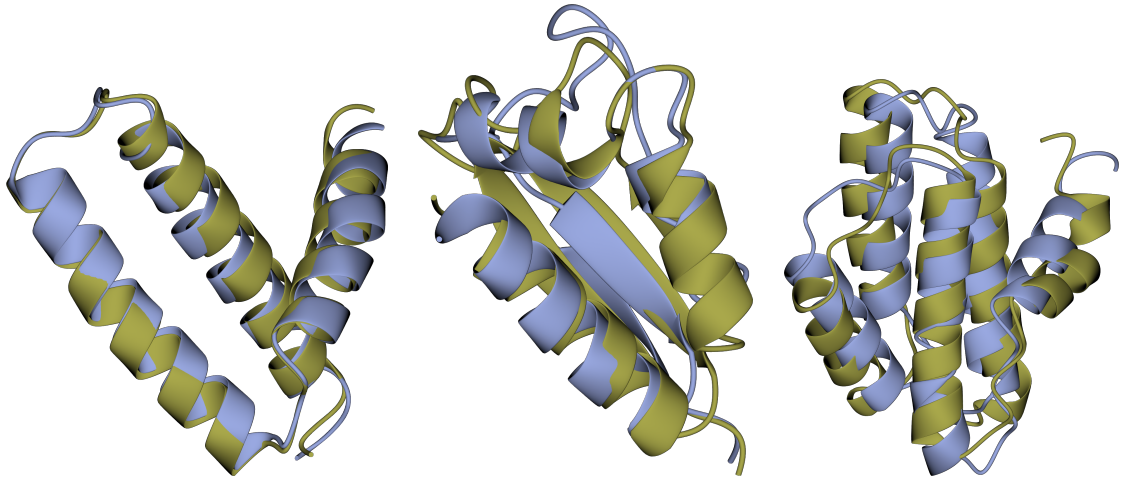
Figure 4.3: Conformations are shown by plotting their Rosetta *score*4 vs. their Cα lRMSD from the native conformation (PDB ID in parentheses) to compare the landscape probed by different algorithms for the target with known native conformation under PDB id 1ail.



(a) Evo-Diverse (1dtja) (b) mEA-PR (1dtja)

Figure 4.4: Conformations are shown by plotting their Rosetta *score*4 vs. their Cα lRMSD from the native conformation (PDB ID in parentheses) to compare the landscape probed by different algorithms for the target with known native conformation under PDB id 1dtja.

On 1dtja, Evo-Diverse reaches a similar lowest lRMSD (2.6Å) as Rosetta and mEA-PR



1ail (lRMSD = 1Å) 1dtja (lRMSD = 2.6Å) 3gwl (lRMSD = 3.2Å)

Figure 4.5: The conformation obtained by Evo-Diverse that is closest to the native conformation is shown for three selected cases, the protein with known native conformation under PDB ID 1ail (left), 1dtja (middle), and 3gwl (right). The Evo-Diverse conformation is in blue, and the known native conformation is in olive.

(confirmed in Fig. 4.2). On 3gwl, Evo-Diverse achieves a dramatic improvement of lowest lRMSD to the native conformation over all other algorithms; while none of the other algorithms reach below 5Å, Evo-Diverse reaches 3.2Å, almost a 2Å improvement.

Comparative Analysis on CASP 12-13 Dataset

Table 4.4 shows the lowest energy and the average energy of the 10 best conformations obtained by Evo-Diverse and Rosetta on each of the 10 target domains denoted by their identifiers in column 1. The lower energy values between the two algorithms on each target domain are marked in bold. Table 4.4 shows that lower energy values are obtained by Evo-Diverse in 7/10 cases compared to Rosetta's 3/10 cases. When the average of the best 10 conformations is considered instead, Evo-Diverse achieves lower energy values in 8/10 cases compared to Rosetta's 2/10 cases.

The above comparisons are further strengthened via statistical analysis. Table 4.8(a)

Table 4.4: Comparison of energy of the lowest energy conformation and average energy of the 10 best conformations (measured in Rosetta Energy Units – REUs) obtained by each algorithm on each of the 10 CASP domains.

		Lowest Energy		**Avg. of the best 10**	
Domain	Length	Rosetta	Evo-Diverse	Rosetta	Evo-Diverse
T1008-D1	77	−164.2	**−166.4**	−162	**−166.3**
T0957s1-D1	108	**−121.5**	−112.6	**−115**	−112.6
T0892-D2	110	−101.8	**−112.3**	−94.1	**−112.3**
T0953s2-D3	93	−53.1	**−67.6**	−49.8	**−66.3**
T0960-D2	84	−79.7	**−82.3**	−79.4	**−82.3**
T0898-D2	55	−65.5	**−66.7**	−62.8	**−66.7**
T0859-D1	129	**−99.5**	−85.6	**−90.7**	−85.6
T0897-D1	138	−141.4	**−147.4**	−137.4	**−147.4**
T0886-D1	69	**−89.2**	−85.4	−84	**−85.4**
T0953s1-D1	67	−51.8	**−59**	−49.1	**−59**

shows the p-values obtained in 1-sided statistical significance tests that pitch Evo-Diverse against Rosetta, evaluating the null hypothesis that Evo-Diverse performs similarly or worse than Rosetta. Both Fisher's and Barnard's test are conducted, and p-values less than 0.05 (which reject the null hypothesis) are marked in bold. Table 4.8(a) shows that the null hypothesis is rejected when the average of the best 10 conformations is considered, and Evo-Diverse performs significantly better than Rosetta with 95% confidence. When the focus is on the lowest energy reached, the performance improvement of Evo-Diverse over Rosetta is not statistically significant at 95% confidence level, although the p-values are very close to the 0.05 threshold.

Table 4.5 shows the lowest lRMSD to the native conformation and the average lRMSD of the 10 best conformations obtained by Evo-Diverse and Rosetta on each of the 10 target domains denoted by their identifiers in column 1. The lower lRMSD values between the two algorithms on each target domain are marked in bold. Table 4.5 shows that lower lRMSDs are obtained by Evo-Diverse in 6/10 cases compared to Rosetta's 4/10 cases. When the average of the 10 best-lRMSD conformations is considered, Evo-Diverse achieves

lower lRMSD in 9/10 cases compared to 2/10 cases of Rosetta.

Table 4.5: Comparison of lRMSD to the native conformation of the lowest lRMSD conformation and average lRMSD to the native of the 10 best conformations (measured in Angstroms – Å) obtained by each algorithm on each of the 10 CASP domains.

		Lowest lRMSD		**Avg. of the best 10**	
Domain	Length	Rosetta	Evo-Diverse	Rosetta	Evo-Diverse
T1008-D1	77	**3.2**	3.5	**3.4**	3.6
T0957s1-D1	108	**6.9**	7.1	8.1	**7.6**
T0892-D2	110	8	**7.4**	8.5	**7.6**
T0953s2-D3	93	8.7	**7.9**	9.3	**8.3**
T0960-D2	84	**7.2**	7.3	**7.6**	**7.6**
T0898-D2	55	6.5	**5.9**	6.7	**6.3**
T0859-D1	129	10.6	**9.4**	11.3	**9.9**
T0897-D1	138	**9**	9.3	10.8	**9.9**
T0886-D1	69	6.3	**6.2**	6.8	**6.6**
T0953s1-D1	67	7	**5.7**	7.4	**6.1**

The above comparisons are further strengthened via statistical analysis. Table 4.8(b) shows the p-values obtained in 1-sided statistical significance tests that pitch Evo-Diverse against Rosetta, evaluating the null hypothesis that Evo-Diverse performs similarly or worse than Rosetta. Again, both Fisher's and Barnard's test are conducted, and p-values less than 0.05 (which reject the null hypothesis) are marked in bold. Table 4.8(b) shows that the null hypothesis is rejected when the average of the best 10 conformations is considered and Evo-Diverse performs significantly better than Rosetta with 95% confidence. When the focus is on the lowest lRMSD reached, the performance improvement of Evo-Diverse over Rosetta is not statistically significant at 95% confidence level.

Table 4.6 shows the highest TM-score to the native conformation and the average TM-score of the 10 best (in terms of TM-scores) conformations obtained by Evo-Diverse and Rosetta on each of the 10 target domains denoted by their identifiers in column 1. The higher TM-score values between the two algorithms on each target domain are marked in bold. Table 4.6 shows that higher TM-scores are obtained by Evo-Diverse and Rosetta

on 5/10 cases. When the focus is on the average TM-score of the best (in terms of TM-scores) 10 conformations is considered, Evo-Diverse achieves higher TM-score in 6/10 cases compared to Rosetta's 5/10.

Table 4.6: Comparison of TM-score of the highest TM-score conformation and average TM-score of the 10 best conformations obtained by each algorithm on each of the 10 CASP domains.

		Highest TM-score		**Avg. of the best 10**	
Domain	Length	Rosetta	Evo-Diverse	Rosetta	Evo-Diverse
T1008-D1	77	**0.61**	0.59	**0.57**	0.55
T0957s1-D1	108	**0.49**	0.42	**0.42**	0.40
T0892-D2	110	0.45	**0.50**	0.42	**0.47**
T0953s2-D3	93	**0.28**	0.25	**0.25**	**0.25**
T0960-D2	84	0.37	**0.39**	0.35	**0.38**
T0898-D2	55	**0.39**	0.37	**0.37**	0.36
T0859-D1	129	0.30	**0.34**	0.29	**0.33**
T0897-D1	138	0.35	**0.36**	0.31	**0.32**
T0886-D1	69	0.42	**0.45**	0.40	**0.41**
T0953s1-D1	67	**0.47**	0.41	**0.43**	0.39

Table 4.8(c) shows the p-values obtained in 1-sided statistical significance tests that pitch Evo-Diverse against Rosetta, evaluating the null hypothesis that Evo-Diverse performs similarly or worse than Rosetta. Both Fisher's and Barnard's test are conducted, and p-values less than 0.05 (which reject the null hypothesis) are marked in bold. Table 4.8(c) shows that the null hypothesis is not rejected with 95% confidence and the performance improvement of Evo-Diverse over Rosetta is not statistically significant.

Table 4.7 shows the highest GDT_TS score to the native conformation and the average GDT_TS score of the 10 best (in terms of GDT_TS scores) conformations obtained by Evo-Diverse and Rosetta on each of the 10 target domains denoted by their identifiers in column 1. The higher GDT_TS scores between the two algorithms on each target domain are marked in bold. Table 4.7 shows that higher values (on both the highest GDT_TS score and the average GDT_TS score over the 10 best conformations) are achieved by Evo-Diverse

in 6/10 cases compared to Rosetta's 5/10.

Table 4.7: Comparison of GDT_TS score of the highest GDT_TS score conformation and average GDT_TS score of the 10 best conformations obtained by each algorithm on each of the 10 CASP domains.

		Highest GDT_TS score		**Avg. of the best 10**	
Domain	Length	Rosetta	Evo-Diverse	Rosetta	Evo-Diverse
T1008-D1	77	**0.62**	0.61	**0.61**	0.58
T0957s1-D1	108	**0.43**	0.39	**0.39**	0.37
T0892-D2	110	0.42	**0.45**	0.39	**0.44**
T0953s2-D3	93	**0.31**	**0.31**	**0.27**	**0.27**
T0960-D2	84	0.37	**0.42**	0.36	**0.39**
T0898-D2	55	**0.46**	0.44	**0.45**	0.43
T0859-D1	129	0.29	**0.32**	0.27	**0.31**
T0897-D1	138	0.30	**0.31**	0.26	**0.28**
T0886-D1	69	0.47	**0.49**	0.45	**0.46**
T0953s1-D1	67	**0.50**	0.46	**0.48**	0.45

Table 4.8(d) shows the p-values obtained in 1-sided statistical significance tests that pitch Evo-Diverse against Rosetta, evaluating the null hypothesis that Evo-Diverse performs similarly or worse than Rosetta. Both Fisher's and Barnard's test are conducted, and p-values less than 0.05 (which reject the null hypothesis) are marked in bold. Table 4.8(d) shows that the null hypothesis is not rejected with 95% confidence and the performance improvement of Evo-Diverse over Rosetta is not statistically significant.

4.1.5 Summary

This section presents a novel conformation ensemble generation algorithm, Evo-Diverse, that operationalizes the multi-objective, stochastic optimization framework. The algorithm does not use total energy as a basis of selection but instead makes use of non-domination rank and crowding distance in its selection operator to encourage conformation diversity. The results show that Evo-Diverse reaches regions of lower total energy in the energy landscape of the datasets used here for evaluation, showcasing its higher exploration capability over

Table 4.8: p-values obtained by 1-sided Fisher's and Barnard's tests on the CASP dataset for head-to-head comparison of the algorithms on lowest energy and average energy of the best 10 conformations (a), lowest lRMSD and average lRMSD of the best 10 conformations (b), highest TM-score and average TM-score of the best 10 conformations (c), and highest GDT_TS score and average GDT_TS score of the best 10 conformations (d). All tests evaluate the null hypothesis that Evo-Diverse does *not* perform better than Rosetta.

(a)

Test	Lowest energy	Avg. energy of the best 10
Fisher's	0.08945	**0.01151**
Barnard's	0.05789	**0.005909**

(b)

Test	Lowest lRMSD	Avg. lRMSD of the best 10
Fisher's	0.3281	**0.002739**
Barnard's	0.2617	**0.001288**

(c)

Test	Highest TM-score	Avg. TM-score of the best 10
Fisher's	0.6719	0.5
Barnard's	0.9991	0.4119

(d)

Test	Highest GDT_TS score	Avg. GDT_TS score of the best 10
Fisher's	0.5	0.5
Barnard's	0.4119	0.4119

the Rosetta conformation sampling protocol and other, state-of-the-art multi-objective EAs that use total energy as an additional optimization objective. In addition, Evo-Diverse is able to reach comparable or lower lRMSDs than Rosetta, thus constituting a clear advantage over the current state-of-the-art multi-objective EAs. It is worth noting that Evo-Diverse does not make use of an archive of all the conformations ever sampled, unlike other multi-objective EAs that do so to update the Pareto metrics for use in the selection operator. Evo-Diverse uses only the current population and their offspring to perform selection, thus saving computational overhead.

4.2 Using Sequence-Predicted Contacts to Guide Conformation Ensemble Generation

The work presented in this section has been published in [76]. As expressed in Chapter 1 and 2, the unreliability of existing energy functions has raised questions in the community on the proper role and utilization of energy functions in protein conformation sampling. The algorithm we presented in Section 4.1 generates conformations by balancing multiple energetic objectives. In this section, we look to improve over it. Recent work suggests employing complementary information in the form of amino-acid contacts and investigates replacing the energy function with a contact-based scoring function, or devising an aggregate scoring function that adds contact information as restraints to an energy function which can be problematic (as explained in Section 1.1). Here, we advance this line of work and present a thorough study on the separate and combined roles and guidance of interatomic energy with contact-based scoring.

We take a single-objective optimization algorithm as a baseline, where the optimization is driven by the energy function entirely. In addition, we develop a single-objective optimization algorithm that utilizes a novel contact-based scoring function as its objective. Building on our work described above on multi-objective optimization, we additionally provide a multi-objective optimization setting, where the energy function is decoupled into several optimization objectives. These are compared with two novel algorithms that treat energy and contact-based scoring as separate optimization objectives, thus providing a comprehensive picture of the contribution of each in isolation and combination in the search for native conformations.

Evaluation on diverse datasets yields many interesting observations and advocates the superiority of combining contact information in conjunction with energy functions for de novo conformation ensemble generation. That is, our findings suggest that neither energy functions nor contact-based scoring are sufficient by themselves as guides towards native conformations; instead, they each provide complementary information that together confers

better performance in a multi-objective optimization setting. The contribution of the work presented in this section goes beyond the specific algorithmic platforms employed here; the work indicates that better performance can be obtained from an optimization method if both energy and contact-based information are employed and considered as optimization objectives.

4.2.1 Algorithms

For the single-objective baseline EA that only utilizes energy, we choose HEA as described in Section 2.2.2. For the baseline multi-objective EA, we employ Evo-Diverse as described in Section 4.1. Note that, in Evo-Diverse, the energy terms in Rosetta *score*4 energy functions are decomposed into 3 groups (short-range hydrogen bonding, long-range hydrogen bonding, and the summation of the rest of the energy terms) that are considered as multiple objectives. We refer to Evo-Diverse as MOEANS (Multi-Objective Evolutionary Algorithm with Non-dominated Sorting) from now on to aid comparison with the novel algorithms we describe below.

HEA-C: Guiding a Hybrid Evolutionary Algorithm by Contact-based Scoring

The selection operator in the HEA is known as selection via truncation. This operator is the one that guides an EA globally in the conformation space. It can be easily modified to select the fittest individuals based on a contact scoring function. In this modified single-objective EA, which we deem HEA-C (C for 'C'ontacts), a conformation is evaluated as follows. The contacts in it are first calculated, using a threshold of 8Å to record whether distances between pairs of non-bonded CB atoms are below the threshold and thus recorded as contacts. RaptorX-Contact [57] is used to obtain contacts predicted from the amino-acid sequence alone of a given target protein. These contacts come with probabilities that provide the confidence of the prediction for each contact. Based on our experiments conducted over known native conformations, the top ten RaptorX-Contact predicted contacts are more accurate and lead to more accurate contact-based scoring functions.

In the CASP contact prediction category that evaluates sequence-based predicted contacts, the known native conformation is treated as the ground truth [58]. In our setting, the native conformation is not known. Instead, we treat the sequence-predicted contacts as the ground truth. Specifically, we focus on the top ten of them (predicted with highest confidence by RaptorX-Contact from a target sequence). These ten pairs of amino acids are evaluated in terms of whether they form a contact or not in a computed conformation that is to be evaluated in HEA-C. If the pairs are also found in contact in a computed conformation, then they contribute to incrementing the number of true postives (TPs). Otherwise, they contribute to incrementing the number of false negatives (FNs) (reported to be in contact by RaptorX-Contact but not found in contact in an HEA-C computed conformation). Each HEA-C conformation evaluated in the selection operator is then scored via TP/(TP+FN); this score is known as sensitivity (or true positive rate – TPR). Note that since we cannot employ contacts of the actual native conformation as the ground truth but instead treat sequence-predicted contacts as the ground truth, we employ sensitivity rather than precision. In summary, HEA-C uses only sensitivity (and not energy) to evaluate parent and offspring in the selection operator; the top N conformations (with the highest sensitivity scores) survive in the next generation, where N is the population size. The rest of the evolutionary ingredients are same as HEA.

MOEANS-EC and MOEANS-SLEC: Energetic and Contact-based Scoring as Optimization Objectives

In MOEANS, the selection operator can be modified to consider more than energetic objectives. We do so in two ways. First, we consider the contact-based scoring function described above and the (total) *score*4 energy as two separate objectives to develop a two-objective EA. We refer to this algorithm as MOEANS-EC. Second, we consider the contact-based scoring function to be the fourth objective in addition to the 3 energetic objectives in MOEANS. We refer this algorithm as MOEANS-SLEC. We note that in these multi-objective EAs, a conformation dominates another conformation if it has a lower energy score for each of the

energy objectives but a higher contact-based score than the other conformation.

4.2.2 Implementation Details

The population for each EA contains $N = 100$ conformation. In keeping with earlier work that evaluates how to stall premature convergence [40], not all parents compete with offspring in HEA and HEA-C. An elitism rate of 25% is employed, where only the top 25% of the parents compete with offspring. This prevents a few fittest parents from taking over the population, effectively providing enough time for suboptimal conformations to improve and contribute "genetic material" over generations. In multi-objective EAs, the diversity of the objectives help to stall premature convergence. Hence, in these EAs, all parents can compete with offspring (elitism rate of 100%). All our EAs are run 5 times on each target protein's amino-acid sequence to account for their stochasticity, and each run has a fixed budget of $10,000,000$ fitness evaluations. These algorithms are implemented in Python, and they interface with PyRosetta and Biopython libraries.

4.2.3 Results

We carry out comparative evaluations on both the benchmark and the CASP datasets. The algorithms presented in Section 4.2.1, HEA, HEA-C, MOEANS, MOEANS-EC, and MOEANS-SLEC, are compared to the Rosetta conformation sampling algorithm, which represents state-of-the-art, energy-guided algorithms, and SCDE, a recent differential evolution algorithm [53] that does not make use of interatomic energy but instead penalizes computed conformations by how much they deviate in their content of secondary structure elements and contacts based on predictions from sequence. We recall that on each target, the algorithms described in Section 4.2.1 are run 5 times, each run employing a budget of 10M fitness evaluations. To ensure a fair comparison, we have run Rosetta with a total budget of $54M$ energetic evaluations per target. In contrast, since SCDE is not available, all results reported are those published in [53]; these results are obtained with 30 runs of SCDE, where each run exhausts a budget of 9M fitness evaluations. The runs (per target)

for each algorithm under comparison are combined, and performance on a target protein sequence is summarized by the lowest energy reached, and the closest proximity to the known native conformation.

Evaluation on Benchmark Dataset

We relate the comparison of HEA, HEA-C, MOEANS, MOEANS-EC, MOEANS-SLEC, and Rosetta on the benchmark dataset in terms of the lowest Rosetta *score*4 energy value reached by each algorithm in Table 4.9 and the lowest lRSMD to the known native conformation per target in Table 4.10. We note that while no energy values are reported for SCDE in [53], lowest lRMSD values are reported, which we have added in Table 4.10. To identify the 20 targets, Column 1 in Tables 4.9 and 4.10 shows the PDB identifier of a representative, known native conformation for each target sequence. The lowest *score*4 and lRMSD achieved on each target are marked in bold.

Several observations can be drawn from the results related in Table 4.9. HEA-C achieves higher energy than all other algorithms. This is expected, as the selection operator in HEA-C uses only the contact-based score. MOEANS-SLEC, which implements multi-objective optimization with objectives being contact-based score, short-term hydrogen bonding, long-term hydrogen bonding, and the rest of *score*4 terms as its fourth objective, reaches lower energy than Rosetta in 15/20 targets, lower energy than HEA in 14/20 targets, lower energy than HEA-C in 20/20 targets, lower energy than MOEANS in only 4/20 targets, and lower-energy than MOEANS-EC in only 3/20 targets. Table 4.13(a) which evaluates the 1-sided statistical significance tests of the performance of MOEANS-SLEC over Rosetta, HEA, and HEA-C, shows that the improvements of MOEANS-SLEC over these three algorithms are statistically significant for both Fisher's and Barnard's tests (p-values < 0.05). This indicates that the multi-objective setting that considers contact score affords a higher exploration capability over single-objective settings, such as Rosetta, HEA, and HEA-C, which consider either energy or contact-based scoring guidance alone.

Table 4.9: Comparison of the lowest *score*4 energy (in Rosetta Energy Units – REUs) obtained by each algorithm under comparison on each of the 20 benchmark targets is shown in Columns 2-7. The PDB ID of the known native conformation of each target is shown in Column 1. The lowest energy value reached per target is marked in bold.

	Lowest Energy (REUs)					
PDB ID	Rosetta	MOEANS	HEA	HEA-C	MOEANS-EC	MOEANS-SLEC
1ail	−29.9	**−82.2**	−56.1	−51	−78	−74.5
1aly	**−112.5**	−67.2	−81.1	−21.1	−77.6	−61.7
1aoy	−73.3	−114.9	−98.1	−74.4	**−115.2**	−101.7
1bq9	−46.9	−68.7	−50.5	−39.9	**−70.1**	−69
1c8ca	**−101.4**	−97.8	−86.4	−43.5	−91.4	−90.1
1cc5	−82.5	−88.3	−68.6	−55.9	**−91.6**	−86.1
1dtdb	−66.5	−67.9	−55	−18.5	**−71.2**	−69.6
1dtja	−72.5	**−87.1**	−82.2	−51.9	−83.9	−81
1fwp	−71.3	**−99.7**	−84.4	−19.5	−88.5	−90.7
1hhp	**−106.3**	−89.8	−104.5	−25.3	−92.8	−95.4
1hz6a	−117.1	−123.3	**−130.9**	−85.9	−124.7	−120.9
1isua	−27	**−63.8**	−46.5	−23.5	−51.5	−48.2
1sap	−107.8	−109.9	**−121.4**	−68.4	−99.1	−97.6
1tig	−138.2	**−145.9**	−128	−64.1	−142	−141.2
1wapa	−109	−105.2	**−132.5**	−45.9	−107.7	−107.1
2ci2	−37.8	−108.4	−109.8	−43.2	−99	**−110.9**
2ezk	−51.1	**−132**	−100.7	−98.4	−128.4	−126.3
2h5nd	−82.5	−148.5	−129	−131.3	**−152.7**	−149.8
2hg6	−82.5	**−117.1**	−102.6	−73.3	−114.5	−104.8
3gwl	−68.2	−115.2	−100	−80.3	**−115.4**	−107.3

MOEANS-SLEC does not provide an improvement over MOEANS and MOEANS-EC, indicating that the breakdown into many objectives does not provide an advantage. MOEANS-EC, where Rosetta *score*4 and the contact-based score are two separate optimization objectives, reaches lower energy than Rosetta in 15/20 targets, lower energy than HEA in 14/20 targets, lower energy than HEA-C in 20/20 targets, and lower energy than MOEANS-SLEC in 17/20 targets. Table 4.14(a) confirms that these improvements are statistically significant for both Fisher's and Barnard's tests. A tie between MOEANS-EC and MOEANS (each achieve lower energy than the other in 10/20 targets) indicates that

Table 4.10: Comparison of the lowest lRMSD (measured in Å) obtained by each algorithm under comparison on each of the 20 benchmark targets is shown in Columns 2-8. The PDB ID of the known native conformation of each target is shown in Column 1. The lowest lRMSD value reached per target is marked in bold.

	Lowest lRMSD (Å)						
PDB ID	Rosetta	SCDE	MOEANS	HEA	HEA-C	MOEANS-EC	MOEANS-SLEC
1ail	4.5	2.6	**1**	1.4	1.6	1.4	1.3
1aly	12.4	11.5	10.8	11.2	11	**10.5**	10.7
1aoy	4	**3**	3.3	3.9	3.7	3.2	3.2
1bq9	**2.9**	*N/A*	3.8	3	3.9	**2.9**	3.4
1c8ca	**2.2**	*N/A*	3.8	4.8	4.6	3.1	4.4
1cc5	**3.7**	4.4	4.6	4.7	4.7	4.7	4.6
1dtdb	4.2	5.8	4.5	4.4	5.4	**4.1**	**4.1**
1dtja	4.1	2.9	2.6	4.2	3.3	**1.9**	2.9
1fwp	**2.8**	*N/A*	3.8	4.3	3.6	3.2	3.7
1hhp	10.1	**7.4**	7.9	8.8	8.3	8.2	8.1
1hz6a	**1.9**	2.4	2.3	**1.9**	2.6	2	2.2
1isua	6.6	5.9	5.9	6.6	6.2	**5.5**	5.7
1sap	**2.8**	5.5	3.2	3.7	4.7	3.9	**2.8**
1tig	**2.5**	3	3.3	3.2	4.2	3.2	4
1wapa	6.5	6	5.4	6.3	6	**5.2**	5.6
2ci2	5.8	*N/A*	3.2	3.7	3.9	3.2	**3**
2ezk	3.6	**2.2**	2.7	3.4	3.1	2.5	2.7
2h5nd	7.4	*N/A*	7	6.2	6.1	**5.7**	6.1
2hg6	9.4	8.7	8.6	9.3	8.4	**8**	8.3
3gwl	5.8	**2.4**	3.2	5.4	3.7	3	3.1

guiding additionally by contact-based scoring does not hamper the exploration of the *score*4 energy surface.

Several observations can be drawn from the results related in Table 4.10. HEA-C achieves higher lRMSD than all other algorithms except for HEA, where it wins on 13/20 targets. This is informative, considering that HEA-C performs a lot worse than HEA and the other algorithms in reaching lower energies. It confirms that energy guidance is not reliable, which was the motivation for the work presented in this section. It also shows that indeed selection by contact-based scoring improves proximity to the native conformations

over selection by energy alone.

In addition, the results in Table 4.10 show that MOEANS-SLEC reaches lower lRMSD than Rosetta in 13/20 targets, lower lRMSD than SCDE in 9/15 targets, lower lRMSD than MOEANS in 14/20 targets, lower lRMSD than HEA in 17/20 targets, and lower lRMSD than HEA-C in 19/20 targets. Table 4.13(b) confirms that the improvements of MOEANS-SLEC in terms of lower lRMSD over MOEANS, HEA, and HEA-C are statistically significant. This result is also interesting, considering that MOEANS-SLEC performs worse than MOEANS in reaching lower energy (related above). The result indicates that adding contact-based score as another optimization objective may hamper the ability to get to the deepest regions of the energy surface by instead guiding the exploration towards regions that, while not as low in energy, are closer to the native conformations in the conformation space.

On the other hand, MOEANS-EC beats all the other algorithms in reaching lower lRMSDs; it wins over Rosetta in 14/20 targets, over SCDE in 9/15 targets, over MOEANS in 16/20 targets, over HEA in 18/20 targets, over HEA-C in 20/20 targets, and over MOEANS-SLEC in 15/20 targets. Table 4.14(b) confirms these improvements by MOEANS-EC are statistically significant over all other algorithms except for SCDE. Taken altogether, these results suggest that while injecting too many optimization objectives may not be beneficial, considering both energy and contact-based score as optimization objectives provides both high exploration capability and better proximity to the native conformation.

4.2.4 Evaluation on CASP Dataset

Table 4.11 compares all algorithms (except for SCDE) in terms of lowest $score4$ energy reached on each of the 10 CASP targets. As in the benchmark dataset, HEA-C achieves higher energy than all other algorithm, as its selection operator does not make use of $score4$. MOEANS-SLEC reaches lower energy than HEA in 6/10 targets and lower energy than HEA-C in all 10/10 targets. Table 4.13(c) confirms that this performance of MOEANS-SLEC is statistically significant. However, as in the benchmark dataset, MOEANS-SLEC

loses to MOEANS in 7/10 targets, loses to MOEANS-EC in 8/10 targets, but ties with Rosetta on the CASP targets, again indicating that the high number of objectives may hamper the exploration capability. On the other hand, MOEANS-EC beats all other algorithms in reaching lowest energy. It wins over Rosetta in 8/10 targets, over MOEANS in 7/10 targets, over HEA in 8/10 targets, over HEA-C in 10/10 targets, and over MOEANS-SLEC in 8/10 targets. Table 4.14(c) confirms these improvements by MOEANS-EC are statistically significant over all algorithms except for MOEANS (which is very close to the 95% confidence cutoff). This agrees with the evaluation on the benchmark dataset and confirms again the high exploration capability of MOEANS-EC.

Table 4.11: Comparison of the lowest *score*4 energy (in Rosetta Energy Units – REUs) obtained by each algorithm under comparison on each of the 10 CASP targets is shown in Columns 2-7. The CASP identifier of each target is shown in Column 1. The lowest energy value reached per target is marked in bold.

Domain	**Lowest Energy (REUs)**					
	Rosetta	MOEANS	HEA	HEA-C	MOEANS-EC	MOEANS-SLEC
T0859-D1	**−99.5**	−85.6	−88	−56.4	−97.8	−77
T0886-D1	−89.2	−85.4	−69.9	−27.5	**−94.5**	−78.5
T0892-D2	−101.8	−112.3	**−116.3**	−59.6	−105.6	−107.4
T0897-D1	−141.4	−147.4	−135.2	−108.5	**−149.8**	−133.6
T0898-D2	−65.5	−66.7	−65.7	−20.1	**−67.6**	**−67.6**
T0953s1-D1	−51.8	**−59**	−55.8	−18.7	−52.3	−48.9
T0953s2-D3	−53.1	−67.6	−62.2	−2.4	**−67.7**	−66.5
T0957s1-D1	**−121.5**	−112.6	−102.6	−66.2	−113.3	−115
T0960-D2	−79.7	**−82.3**	−67.6	−19.3	−81.2	−79.9
T1008-D1	−164.2	−166.4	−148.4	−108.6	**−168.1**	−167.7

Table 4.12 compares all algorithms in terms of lowest lRMSD and highest GDT_TS to the native conformation. As in the benchmark dataset, HEA-C achieves higher lRMSD than all other algorithms except for HEA, where it wins on 8/10 targets. This result confirms the utility of contact-based scoring in guiding towards near-native regions of the conformation space. MOEANS-SLEC reaches lower lRMSD than (and so wins over) Rosetta in

8/10 targets, wins over MOEANS in 8/10 targets, over HEA in 10/10 targets, and over HEA-C in 9/10 targets. Table 4.13(d) confirms the improvements by MOEANS-SLEC are statistically significant. This result also agrees with the benchmark dataset, and similar observations can be made regarding the utility of considering energetic and contact-based scoring objectives. On the other hand, MOEANS-EC beats all other algorithms in reaching lowest lRMSD. It wins over Rosetta in 10/10 targets, over MOEANS in 9/10 targets, over HEA in 10/10 targets, over HEA-C in 8/10 targets, and over MOEANS-SLEC in 7/10 targets. Table 4.14(d) confirms these improvements by MOEANS-EC are statistically significant over all algorithms except for MOEANS-SLEC. These observations are very similar to those drawn from the evaluation on the benchmark dataset. They confirm that additional guidance by contact-based scoring in a multi-objective setting improves proximity to the native conformations, but that adding too many objectives results in diminishing returns.

Table 4.12: Columns 2-7 relate the lowest lRMSD (in Å) and the highest GDT_TS obtained by each algorithm under comparison on each of the 10 CASP targets in comparison to corresponding native conformations. The CASP identifier of each target is shown in Column 1. The lowest lRMSD (and the highest GDT_TS) value reached per target is marked in bold.

Domain	**Lowest lRMSD (Å), Highest GDT_TS**					
	Rosetta	MOEANS	HEA	HEA-C	MOEANS-EC	MOEANS-SLEC
T0859-D1	10.6, 0.29	9.4, **0.32**	9.6, 0.29	9.3, 0.29	9.4, **0.32**	**9.1, 0.32**
T0886-D1	6.3, 0.47	6.2, **0.49**	6.4, 0.40	6.3, 0.41	**5.6**, 0.44	5.9, 0.41
T0892-D2	8, 0.42	7.4, **0.45**	7.2, 0.42	7.5, 0.40	**6.9**, 0.43	7.2, 0.42
T0897-D1	9, 0.30	9.3, 0.31	9.3, 0.27	8.7, 0.29	**8.5, 0.33**	8.9, 0.32
T0898-D2	6.5, 0.46	5.9, 0.44	6.1, 0.45	**5.8**, 0.45	5.9, **0.52**	**5.8**, 0.48
T0953s1-D1	7, 0.50	**5.7**, 0.46	6.2, 0.43	6.2, 0.44	5.9, **0.51**	6, 0.47
T0953s2-D3	8.7, **0.31**	7.9, **0.31**	8, 0.28	7.8, 0.30	**7.7**, **0.31**	**7.7**, **0.31**
T0957s1-D1	6.9, **0.43**	7.1, 0.39	7.4, 0.37	7.4, 0.36	6.8, 0.40	**6.5, 0.43**
T0960-D2	**7.2**, 0.37	7.3, **0.42**	7.6, 0.33	7.4, 0.33	**7.2**, 0.38	7.4, 0.38
T1008-D1	3.2, 0.62	3.5, 0.61	3.6, 0.62	3.8, 0.62	**2.8, 0.64**	3.5, 0.61

The comparison on GDT_TS in Table 4.12 shows similar results. HEA-C achieves lower

GDT_TS than all other algorithms except for HEA, where it wins in 8/10 targets compared to HEA's win in 6/10 targets. MOEANS-SLEC wins over Rosetta in 7/10 targets, over MOEANS in 7/10 targets, over HEA in 9/10 targets, and over HEA-C in 9/10 targets. Table 4.13(e) confirms the improvements by MOEANS-SLEC are statistically significant over HEA and HEA-C. On the other hand, MOEANS-EC beats all other algorithms in reaching highest GDT_TS. It wins over Rosetta in 8/10 targets, over MOEANS in 7/10 targets, over HEA in 10/10 targets, over HEA-C in 10/10 targets, and over MOEANS-SLEC in 9/10 targets. Table 4.14(e) confirms these improvements by MOEANS-EC are statistically significant over all the algorithms except for MOEANS. These results harden our observation that guidance by both energy and contact-based scoring in a multi-objective optimization setting improves proximity to the native conformations; however, when adding too many objectives, the performance may experience diminishing returns.

Discussion

Synthesizing all observations drawn from the comparative evaluations above, several conclusions emerge. First, if the goal is to reach deeper in the energy surface (which is often used to evaluate the exploration capability of an algorithm operating within a fixed computational budget), a multi-objective optimization setting outperforms a single-objective setting, unless too many optimization objectives are considered. Both observations have been drawn before by our own work in conformation ensemble generation and work of others in hard optimization problems beyond protein modeling [39, 45, 46, 74]. Higher exploration capability does not necessarily translate to better proximity to the native conformation. Our evaluation clearly makes this case. Indeed, considering a contact-based scoring in the selection operator, whether in isolation in a single-objective multi-optimization setting (as in HEA-C) or jointly with energy in a multi-objective optimization setting (as in MOEANS-EC and MOEANS-SLEC), improves proximity to the native conformations.

To shed more light over the role of contact information on the quality of computed conformations, Fig. 4.6 plots the (sensitivity) contact-based score against the GDT_TS

Table 4.13: Comparison of MOEANS-SLEC to other algorithms via 1-sided Fisher's and Barnard's tests. The tests evaluate the null hypothesis that MOEANS-SLEC does not achieve (a) lower lowest energy on benchmark dataset, (b) lower lowest lRMSD on benchmark dataset, (c) lower lowest energy on CASP dataset, (d) lower lowest lRMSD on CASP dataset, (e) higher highest GDT_TS on CASP dataset in comparison to a particular algorithm, considering each of the other algorithms in turn. P-values less than 0.05 are marked in bold.

Test	Rosetta	MOEANS	HEA	HEA-C	MOEANS-EC	SCDE
Fisher's	**0.001924**	N/A	**0.01282**	**7.254e-12**	N/A	N/A
Barnard's	**0.001118**	N/A	**0.008299**	**9.095e-13**	N/A	N/A

(a)

Test	Rosetta	MOEANS	HEA	HEA-C	MOEANS-EC	SCDE
Fisher's	0.1025	0.05548	**7.254e-12**	**2.91e-08**	N/A	0.3576
Barnard's	0.07693	**0.03517**	**9.095e-13**	**9.733e-09**	N/A	0.2923

(b)

Test	Rosetta	MOEANS	HEA	HEA-C	MOEANS-EC	SCDE
Fisher's	0.6719	N/A	0.3281	**5.413e-06**	N/A	N/A
Barnard's	0.9991	N/A	0.2617	**9.537e-07**	N/A	N/A

(c)

Test	Rosetta	MOEANS	HEA	HEA-C	MOEANS-EC	SCDE
Fisher's	**0.01151**	**0.03489**	**5.954e-05**	**0.009883**	N/A	N/A
Barnard's	**0.005909**	**0.02069**	**2.003e-05**	**0.003978**	N/A	N/A

(d)

Test	Rosetta	MOEANS	HEA	HEA-C	MOEANS-EC	SCDE
Fisher's	0.5	0.5	**0.002739**	**0.002739**	N/A	N/A
Barnard's	0.3883	0.3883	**0.001288**	**0.001288**	N/A	N/A

(e)

score of the best conformations (with highest GDT_TS) for each of the CASP targets. We recall that the only EAs that use contact-based score alone or in conjunction with energy are HEA-C, MOEANS-EC, and MOEANS-SLEC. Fig. 4.6 shows that conformations that have a better GDT_TS score (more than 0.4) also overall have a higher contact-based score (more than 0.5), providing further evidence that guidance by a contact-based score such as the one employed in this work provides a soft bias towards better (more near-native) conformations.

Table 4.14: Comparison of MOEANS-EC to other algorithms via 1-sided Fisher's and Barnard's tests. The tests evaluate the null hypothesis that MOEANS-EC does not achieve (a) lower lowest energy on benchmark dataset, (b) lower lowest lRMSD on benchmark dataset, (c) lower lowest energy on CASP dataset, (d) lower lowest lRMSD on CASP dataset, (e) higher highest GDT_TS on CASP dataset in comparison to a particular algorithm, considering each of the other algorithms in turn. P-values less than 0.05 are marked in bold.

Test	Rosetta	MOEANS	HEA	HEA-C	MOEANS-SLEC	SCDE
Fisher's	**0.001924**	0.6238	**0.01282**	**7.254e-12**	**9.693e-06**	N/A
Barnard's	**0.001118**	0.9982	**0.008299**	**9.095e-13**	**4.182e-06**	N/A

(a)

Test	Rosetta	MOEANS	HEA	HEA-C	MOEANS-SLEC	SCDE
Fisher's	**0.02808**	**0.0006159**	**3.43e-05**	**1.523e-10**	**0.01242**	0.233
Barnard's	**0.01924**	**0.0003401**	**1.109e-05**	**3.729e-11**	**0.006934**	0.1808

(b)

Test	Rosetta	MOEANS	HEA	HEA-C	MOEANS-SLEC	SCDE
Fisher's	**0.01151**	0.08945	**0.01151**	**5.413e-06**	**0.03489**	N/A
Barnard's	**0.005909**	0.05789	**0.005909**	**9.537e-07**	**0.02069**	N/A

(c)

Test	Rosetta	MOEANS	HEA	HEA-C	MOEANS-SLEC	SCDE
Fisher's	**5.954e-05**	**0.009883**	**5.413e-06**	**0.01151**	0.1849	N/A
Barnard's	**2.003e-05**	**0.003978**	**9.537e-07**	**0.005909**	0.1317	N/A

(d)

Test	Rosetta	MOEANS	HEA	HEA-C	MOEANS-SLEC	SCDE
Fisher's	**0.03489**	0.325	**5.413e-06**	**5.413e-06**	**0.02864**	N/A
Barnard's	**0.02069**	0.2617	**9.537e-07**	**9.537e-07**	**0.01139**	N/A

(e)

Our comprehensive evaluation demonstrates that considering energy and contact-based scoring jointly, as separate optimization objectives (but not too many objectives) performs best in both reaching lower energies and better proximity to known native conformations;

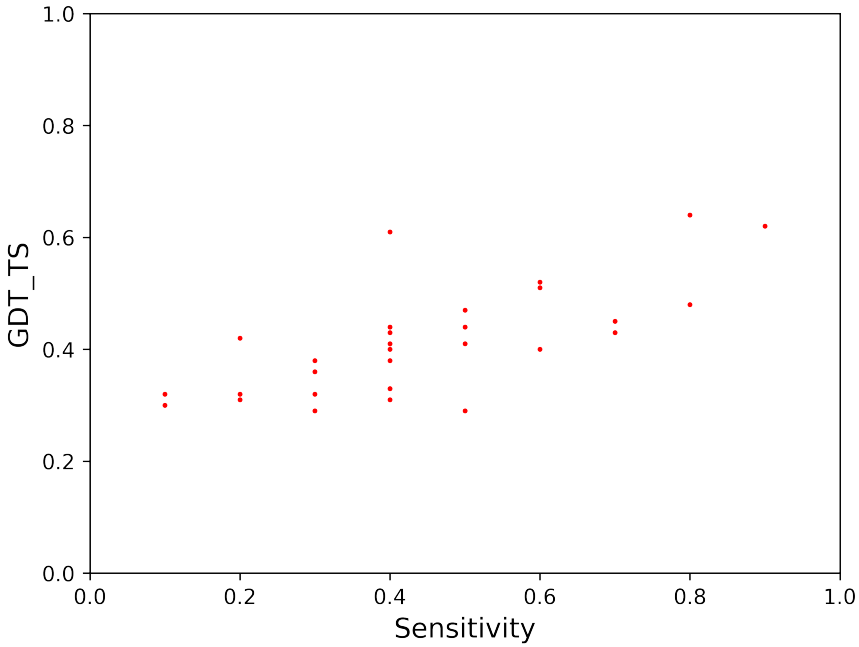
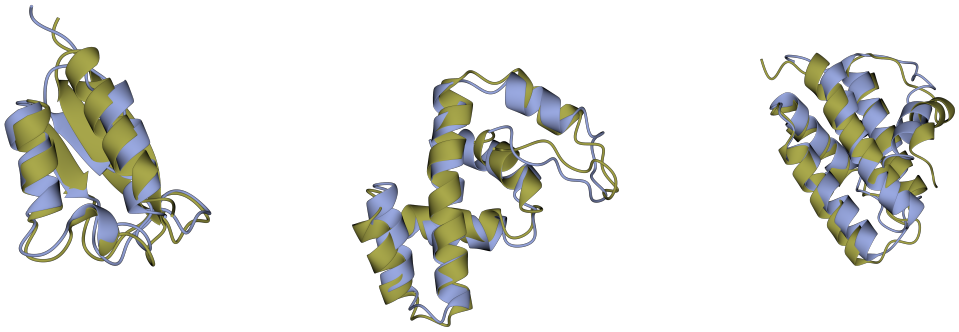


Figure 4.6: Best conformations sampled by HEA-C, MOEANS-EC, and MOEANS-SLEC for each of the CASP targets are shown by plotting their GDT_TS score vs. contact sensitivity score.

that is, when contact information is used as an additional objective to energy in a multi-objective EA (MOEANS-EC), it results in similar or better exploration and better proximity to the native conformation than if only energy or contact information are used alone. The quality of the conformations obtained by MOEANS-EC is shown qualitatively in Fig. 4.7, which draws the lowest-lRMSD conformation obtained by MOEANS-EC in three selected targets, superimposing it over the known native.

4.2.5 Summary

In this work, we investigate the role of energy and contact information in conformation ensemble generation. We use sequence-based predicted contacts from RaptorX-Contact and make use of sensitivity to evaluate the derived contact map of a computed conformation in relation to the sequence-predicted one from RaptorX. Unlike existing work, we do not resort



1dtja (1.9Å) 2ezk (2.5Å) 3gwl (3Å)

Figure 4.7: The MOEANS-EC conformation closest to the known native conformation for proteins with PDB ID 1dtja (left), 2ezk (middle), and 3gwl (right) is drawn in blue, superimposed over the native conformations, drawn in olive. Rendering is performed with the CCP4mg molecular graphics software [1].

to aggregating energy with contact-based scoring, but instead consider them as separate optimization objectives, making use of multi-objective and single-objective optimization frameworks to holistically evaluate the separate versus the combined role and impact of energy versus contacts in conformational sampling. The results suggest strong merit in using contact information jointly with interatomic energy in a multi-objective optimization setting.

Chapter 5: Promoting Practical Use of Conformation Ensemble Generation Algorithms

As stated in Chapter 1, Currently, when employing conformation ensemble generation algorithms, the common practice is to generate as many conformations that can be afforded. This practice acknowledges that more conformations means higher likelihood that some will reside near the sought native conformations, but it is impractical for various reasons. While generating conformations used to be significantly more expensive than analyzing conformations, now this relationship is less imbalanced. Great progress in software and hardware have made it less costly to generate conformations. Algorithms operating under the umbrella of evolutionary computation can generate hundreds of thousands of conformation. Selection algorithms that analyze these conformations to filter out the near-native conformations now have to additionally deal with a data size issue. Moreover, conformation ensemble generation algorithms typically operate on a coarse-grained representation to sample conformations in a simplified space. Therefore, the generated conformations need to go through the refinement stage before they are handed to the selection algorithms, which adds back the atomistic detail (the side-chains) and performs local improvements on the all-atom conformations. Refinement is computationally expensive as the energy function employed has to additionally deal with all the side-chain and hydrogen atoms of each amino acid. In this chapter, we aim to reduce the size of the ensemble generated by the conformation ensemble generation algorithms to promote the practical use of such algorithms. We first show that it is possible to effectively represent the originally generated ensemble with a reduced-size ensemble in Section 5.1. Then, we introduce a mechanism through which conformation ensemble generation algorithms can generate such a reduced ensemble on the fly in Section 5.2. Finally, we present a technique to utilize the generated reduced ensemble to guide the search simultaneously to enhance exploration of the conformation space.

5.1 Reducing Generated Ensemble

The work presented in this section has been published in [77, 78]. Here, we set out to evaluate the hypothesis that the generated conformation ensemble can be significantly reduced without sacrificing conformation quality. Our goal is to demonstrate that the ensemble of conformations produced by the conformation ensemble generation algorithms can be reduced, thus lowering the computational burden on the refinement and the selection phase, all the while without sacrificing the quality of the original ensemble. We do so via a clustering-based approach.

To the best of our knowledge, the problem of conformation ensemble reduction while preserving quality is unexplored. The problem is also not trivial. In such a setting, it may be tempting to tackle it by discarding higher-energy conformations over an energy threshold. Indeed, early work in [79] does so before proceeding to cluster the remaining conformations for the purpose of conformation selection. As we show in our evaluation later, an approach that simply utilizes an energy threshold, which we employ as a baseline for the purpose of comparison, does a poor job at retaining near-native conformations. This is not surprising as existing energy functions are not reliable indicators of nativeness. Some related attempts reduce the dimensionality of the conformation space populated by a conformation ensemble generation algorithm [80, 81]. However, while dimensionality reduction techniques may be useful for visualization of conformation ensembles, they do not directly apply to ensemble reduction.

We focus on representative clustering algorithms to cluster similar conformations in a generated ensemble based on their shape similarity and then choose from the clusters to populate a reduced ensemble. We refer to the ensemble of conformations generated by some conformation ensemble generation algorithm as Ω_{gen} and to the reduced ensemble (by our approach) as Ω_{red}. Evaluations carried out on diverse target protein datasets show that the proposed approach yields drastic reductions in ensemble size while retaining conformation quality. The results presented in this section suggest that research on conformation ensemble reduction is a promising direction to aid conformation sampling and can generally be useful

in reducing molecular structure data.

5.1.1 Generation of Conformations of a Target Protein

We first have to generate the ensemble Ω_{gen}. Many options are available. We could have utilized Rosetta, Quark, or other conformation sampling algorithms. We choose to utilize HEA (described in Section 2.2.2). While any conformation ensemble generation algorithm can be used to generate the Ω_{gen} ensemble for our purposes, we specifically employ HEA due to its high exploration capability [40,41]; the algorithm can generate hundreds of thousands of conformations for a target protein (given its amino-acid sequence) in a matter of hours.

5.1.2 Featurizing Generated Conformations

We utilize the Ultrafast Shape Recognition (USR) metrics that were originally introduced in [82] to summarize three-dimensional structures of ligands. USR metrics were used in [82] to expedite searches for similar structures in molecular structure databases. These metrics have also been used by others to expedite robotics-inspired algorithms exploring protein conformation spaces [83, 84] and protein motion computation [18, 85].

We use the USR metrics as features to describe a conformation. In summary, USR metrics are based on moments of distance distributions (of atoms). They summarize molecular shapes and so allow to compare molecular shapes efficiently. USR metrics summarize the distributions of distances of all atoms from four chosen reference points in a conformation: the molecular centroid (ctd), the closest atom to ctd (cst), the farthest atom to ctd (fct), and the farthest atom to fct (ftf). The moments of these discrete distributions are recorded to summarize the geometry of a molecule and its shape. Specifically, in our work (as originally in [82]), the resulting distributions are summarized with three momenta, the mean, variance, and skewness. Hence, each conformation in Ω_{gen} is represented by 12 features.

The motivation of encoding each conformation via features is three-fold. First, a lower number of coordinates required to represent each conformation reduces the computational time of any algorithm expected to process the generated conformations. Second, high

dimensionality has a negative impact on the performance of clustering algorithms [86–88]. Third, unlike representations based on Cartesian coordinates, the USR-based representation is invariant to rigid-body motions (translation and rotation in 3D space).

5.1.3 Clustering Featurized Conformations

The featurized conformations are subjected to a clustering algorithm. We evaluate four clustering algorithms, three popular, representative clustering algorithms, k-means, Gaussian Mixture Model (GMM), and hierarchical clustering, and a variant of the gmx-cluster algorithm in the GROningen MAchine for Chemical Simulations (GROMACS) package [89]. The latter has been shown to be effective in clustering protein conformations [90]. We briefly summarize each algorithm next, paying more attention to describing how we optimize their parameters and apply them to the featurized conformations.

In k-means, the number of clusters k is a hyper-parameter. The conformations that can serve as cluster centroids is another hyper-parameter. We optimize both as follows. For a given value of k, k conformations are initially selected uniformly at random over Ω_{gen} to act as the cluster centroids. This induces a particular grouping C of the conformations, with each conformation assigned to the cluster represented by the conformation to which it is closest. To evaluate this particular grouping C, we calculate the within-cluster scatter loss function: $L(C) = \frac{1}{2}\sum_{l=1}^{k}\sum_{i\in C_l}\sum_{j\in C_l, j\neq i} D(x_i, x_j)$, where $D(x_i, x_j)$ measures the Euclidean distance between two points/conformations $x_i \neq x_j$ in the same cluster C_l, where $l \in \{1, \ldots, k\}$. One can now vary the conformations selected to serve as cluster centroids over iterations and record the selection resulting in the smallest loss. We do so over 10 iterations for a given k, randomly selecting conformations as initial centroids in each iteration, recording the optimal selection (and associated grouping) for each iteration.

Note that the above is carried out for a given k as k varies in a permissive range. To find the optimal number of clusters, k, in some considered range, we utilize the popular knee-finding approach [91]. Specifically, after the centroids of clusters are determined (optimally) as above for a given k, the squared distance of each conformation in a cluster

from the centroid of the cluster can be recorded, and the sum of these squared distances can be obtained over the clusters k [92]. This sum of squared distances is known as the sum of squared errors (SSE) and is shown for a particular conformation dataset in Figure 5.1. In Figure 5.1, different values of k are plotted against the corresponding SSE values. The knee (also referred to as elbow) in the SSE curve indicates the optimal number of clusters. We are interested in a small value for SSE. Naturally, as one increases k, the SSE approaches 0. It is exactly 0 when $k = |\Omega_{\text{gen}}|$ (every conformation is the centroid of its own cluster). The goal is to choose a small value of k that results in a low SSE. The knee or elbow in the curve that tracks SSE as a function of k corresponds to the region where by increasing k, SSE does not change noticeably; this is annotated in Figure 5.1.

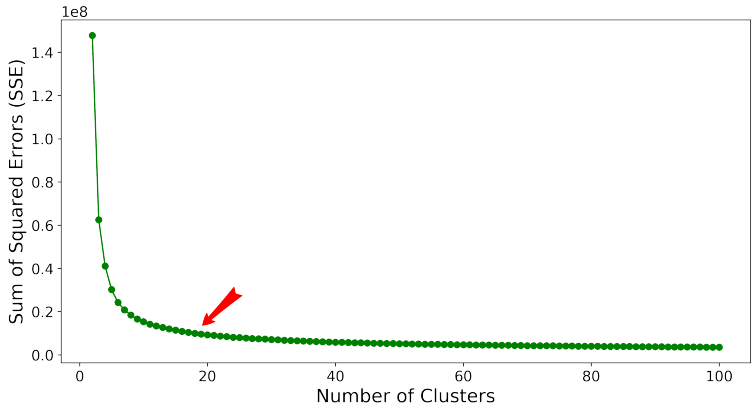


Figure 5.1: The sum-of-squared errors (SSE) is plotted as a function of the number of clusters k identified via k-means on conformations generated via HEA on a target protein. This target is part of our evaluation dataset related in Section 5.1.6. Specifically, it is the target protein with known native conformation in the PDB entry with identifier (id) 1ail. The red arrow points to the knee/elbow region where by increasing k SSE does not change noticeably; this is the region from where an optimal value of k can be selected.

GMM is a probabilistic model that assumes a mixture of finite number of Gaussian distributions with unknown parameters as the underlying process generation of the data. GMM can be thought of as generalizing k-means, as it includes both information from the covariance structure of the data along with the centers of the Gaussian distributions. The

main advantage of GMM is the estimation of uncertainty in data membership to clusters; a conditional probability is assigned to each data indicating the probability with which a specific point belongs to any cluster. As expected, sum of all these conditional probabilities for a given point is 1. This uncertainty assignment makes GMM more informative than k-means [93].

However, as in k-means, one needs to specify the number of clusters/components *a priori* in GMM. The optimal value can be determined by minimizing the Bayesian Information Criterion (BIC) [94] metric which considers both covariance type and the number of components. The BIC is a penalty term for the possible likelihood increment when adding more parameters into the model. Specifically, $BIC = \ln(n)k - 2\ln(\hat{L})$, where k is the number of components, $\hat{L}$ is the maximized value of the likelihood function, and n is the number of data points. In Figure 5.2, we plot the BIC value as the function of the number of components k to demonstrate how one can identify a reasonable value for k at the lowest BIC value.

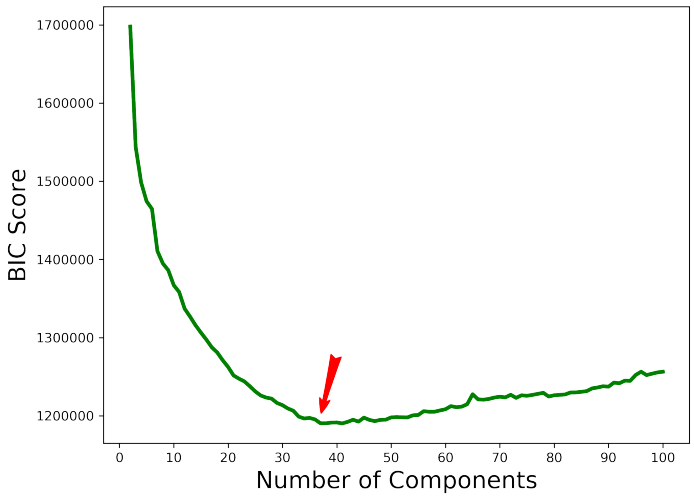


Figure 5.2: The BIC is plotted as a function of the number of components k. Clustering is carried out via GMM on conformations generated via HEA on a target protein (known native conformation in the PDB entry with identifier (id) 2h5nd). The red arrow points to the value for k identified at the lowest BIC value.

Unlike k-means and GMM, hierarchical clustering does not require *a priori* specifying the number of clusters. It refers to a family of clustering algorithms that build a sequence of nested clusters by merging or splitting them successively [95]. We make use of the bottom-up (agglomerative) approach for hierarchical clustering; each conformation is first in its own clusters, and then clusters are successively merged until the root of the resulting dendrogram is reached, with a unique cluster containing all the data. The linkage criterion specifies the merge strategy. We select single linkage, where the distance between two clusters is defined as the distance between the two closest points across the two clusters [96].

"Cutting" at different locations of the dendrogram results in different partitions of the dataset into clusters. To avoid recomputation of the clusters, we make use of a cached implementation of hierarchical clustering, where cutting the tree at different places does not impose any further computation. We employ the Davies-Bouldin (DB) index [97] to determine where to cut the dendrogram. DB is as popular clustering validation technique in the absence of ground truth labels. It is computed on features inherent to the dataset and gives a measure of the average similarity of each cluster with its most similar cluster. Specifically, the DB index evaluates the intra-cluster similarity and inter-cluster differences to provide a non-negative score. A lower DB index corresponds to a better separation between the clusters. In our application of hierarchical agglomerative clustering with single linkage, we consider the DB index at every height of the tree, and we select the height that results in the smallest DB as the optimal partition (and optimal corresponding number of clusters) of a conformation dataset.

Unlike the above clustering algorithms, the gmx-cluster algorithm determines clusters based on a pre-specified distance cutoff. The algorithm first calculates the pairwise distance between all pairs of conformations. For each conformation x_i, the algorithm then counts the number of other conformations (neighbors) that are within the distance cutoff. The conformation with the highest number of neighbors is then chosen as the central conformation and forms a cluster together with all its neighbors. The formed cluster is then removed from the ensemble of conformations and the process is repeated for the remaining conformations

in the ensemble until the ensemble contains no more conformations.

The computation of pairwise distances can potentially be very demanding on large datasets, if one were to use the gmx-cluster implementation that uses lRSMD as the distance metric. Our adaptation of this algorithm transfers all neighbor computations in the USR feature space, using Euclidean distance in the USR feature space as a proxy for lRMSD. These distances, to which we refer as USR scores (and analyze in some detail in Section 5.1.7), are normalized between 0 and 1, so that we can set a distance cutoff. We set this cutoff to 0.1; our analysis shows that this is a reasonable value. From now on, we will refer to the adaptation of gmx-cluster as gmx-cluster-usr or gmx-usr for short.

5.1.4 Selecting Conformations to Populate the Reduced Ensemble

After clustering the featurized Ω_{gen}, the conformations are grouped in clusters. The *selector* now selects a subset of conformations from each cluster to populate the reduced ensemble Ω_{red}. The selector makes this decision by considering both the identified clusters and the Rosetta *score*4 energy score of conformations. The selector we propose organizes the conformations in a cluster into levels/bins; the conformations placed in the same bin have identical *score*4 energies up to two digits after the decimal sign. One conformation is selected at random from each bin and placed in the reduced ensemble Ω_{red}. This process is repeated for each identified cluster. We note that the selector can control the size of the reduced ensemble by tuning the width of a bin/level. This approach indirectly biases the reduced ensemble by cluster size. Larger clusters with more conformations result in more energy levels; therefore, more representative conformations are selected from larger clusters to populate the reduced ensemble. Conformation diversity retention is another indirect property of this approach as demonstrated experimentally in Section 5.1.6.

5.1.5 Implementation Details

To account for stochasticity, the HEA algorithm is run 5 times for each protein target to generate $50,000$ conformations on each run; the conformations generated in each run are

aggregated to populate the Ω_{gen} ensemble of $250,000$ conformations per target. In all the clustering algorithms employed here to cluster the featurized Ω_{gen} ensemble, determining the number of clusters takes most of the time. Including the conformation ensemble generation phase, runtime varies between 7-16 hours for a single run on one Intel Xeon E5-2670 CPU with 2.6GHz base processing speed and 100GB of RAM. We note that all our implementations and analyses are carried out in Python. The scikit library is utilized to obtain access to the k-means, GMM, and hierarchical clustering algorithms employed here.

5.1.6 Results

The Ω_{gen} and Ω_{red} ensembles are compared by size, quality, and diversity. The Ω_{red} ensemble obtained by k-means, GMM, hierarchical clustering, or gmx-cluster-usr is also compared to the Ω_{red} ensemble obtained via truncation selection. To populate the reduced ensemble from the truncation-based approach, given a target size M, higher-energy conformations are discarded to keep the M lowest-energy conformations in an ensemble. We choose the maximum size over Ω_{red} identified by k-means, GMM, hierarchical clustering, and gmx-cluster-usr to set the target size M for truncation selection. As the results presented below demonstrate, k-means and GMM result in larger, reduced ensembles compared to those obtained with the hierarchical clustering or gmx-cluster-usr; therefore, the size of the reduced ensemble in the truncation based-approach matches the size of the reduced ensemble obtained via k-means or GMM.

Evaluation is carried out on 10 targets each from the benchmark and the CASP datasets. As described above, k-means, GMM, hierarchical, or gmx-cluster-usr clustering are employed. Regardless of which process is used (SSE-, BIC- or DB-based) to identify an optimal value for the number of clusters, this number varies for each target protein. For most of the target proteins, the number of clusters is in the $20 - 40$ range. This suggests that a large number of similar conformations are present in the generated conformation ensemble; therefore, finding the underlying structure to reduce the generated conformation ensemble while retaining the quality and diversity is a reasonable goal.

Comparing Ensemble Sizes Pre- and Post Reduction

In Table 5.1, Ω_{gen} and Ω_{red} are first compared in terms of size over the benchmark dataset. The reduction percentage $(1 - \frac{|\Omega_{\text{red}}|}{|\Omega_{\text{gen}}|}) \cdot 100\%$ is also reported for each target. The reductions obtained by k-means range from 54% to 71%. The GMM reductions vary from 53% to 71%. Gmx-cluster-usr and hierarchical clustering result in more dramatic reductions of more than 72% and 77% on all targets respectively, and over 80% on 5/10 and 9/10 of the targets, respectively. Similar results are obtained over the CASP dataset, as shown in Table 5.2. Reductions of 59% and higher are obtained via k-means. Reductions obtained via GMM are comparable to those obtained via k-means. Reductions of 72% and higher are achieved via gmx-cluster-usr. Reductions of around 80% and higher are obtained via hierarchical clustering.

Table 5.1: Ω_{gen} and Ω_{red} are compared in terms of size over the benchmark dataset. The PDB ids of each target is shown in Columns 1. Column 2 shows the size of Ω_{gen}. The size of Ω_{red} and the reduction of Ω_{red} over Ω_{gen} are shown in Columns $3 - 10$ for all clustering algorithms.

		K-means		**GMM**		**Hierarchical**		**Gmx-usr**	
PDB Id	$\lvert\Omega_{\text{gen}}\rvert$	$\lvert\Omega_{\text{red}}\rvert$	Red. (%)	$\lvert\Omega_{\text{red}}\rvert$	Red. (%)	$\lvert\Omega_{\text{red}}\rvert$	Red. (%)	$\lvert\Omega_{\text{red}}\rvert$	Red. (%)
1ail	250K	94,867	62.05	99,707	60.12	32,432	87.03	48,450	80.62
1bq9	250K	79,181	68.33	77,873	68.85	24,705	90.12	39,716	84.11
1c8ca	250K	87,209	65.12	88,437	64.63	29,795	88.08	46,817	81.27
1cc5	250K	97,878	60.85	101,589	59.36	36,630	85.35	55,673	77.73
1dtja	250K	75,421	69.83	79,134	68.35	29,506	88.2	41,456	83.42
1hhp	250K	71,926	71.23	71,390	71.44	27,208	89.12	42,226	83.11
1tig	250K	94,656	62.14	97,010	61.2	40,145	83.94	57,033	77.19
2ezk	250K	114,244	54.3	115,929	53.63	49,509	80.2	62,439	75.02
2h5nd	250K	110,196	55.92	111,353	55.46	55,153	77.94	67,671	72.93
3gwl	250K	101,827	59.27	105,214	57.91	46,480	81.41	63,116	74.75

Table 5.2: Ω_{gen} and Ω_{red} are compared in terms of size over the CASP dataset. The CASP ids are shown in Columns 1. Column 2 shows the size of Ω_{gen}. The size of Ω_{red} and the reduction of Ω_{red} over Ω_{gen} are shown in Columns $3-10$ for all clustering algorithms.

CASP Id	$\lvert\Omega_{\text{gen}}\rvert$	**K-means**		**GMM**		**Hierarchical**		**Gmx-usr**	
		$\lvert\Omega_{\text{red}}\rvert$	Red. (%)	$\lvert\Omega_{\text{red}}\rvert$	Red. (%)	$\lvert\Omega_{\text{red}}\rvert$	Red. (%)	$\lvert\Omega_{\text{red}}\rvert$	Red. (%)
T0859-D1	250K	91,236	63.51	94,014	62.39	32,060	87.18	50,903	79.64
T0886-D1	250K	72,351	71.06	88,986	64.41	27,328	89.07	42,397	83.04
T0892-D2	250K	89,943	64.02	92,200	63.12	39,669	84.13	55,482	77.81
T0897-D1	250K	98,262	60.7	101,119	59.55	50,352	79.86	68,703	72.52
T0898-D2	250K	67,046	73.18	67,283	73.09	21,332	91.47	35,053	85.98
T0953s1-D1	250K	51,078	79.57	50,509	79.8	16,417	93.43	29,690	88.12
T0953s2-D3	250K	73,191	70.72	74,974	70.01	22,143	91.14	38,372	84.65
T0957s1-D1	250K	92,028	63.19	93,872	62.45	38,665	84.53	54,951	78.02
T0960-D2	250K	53,388	78.64	52,136	79.15	22,171	91.13	32,548	86.98
T1008-D1	250K	101,433	59.43	105,360	57.86	51,809	79.28	68,428	72.63

Comparing Distributions of lRMSDs from the Native Conformation Pre- and Post Reduction

Table 5.3 compares the minimum, average, and standard deviation of lRMSDs of conformations in the Ω_{red} and Ω_{gen} ensembles to the known native conformation on each target in the benchmark dataset. The top panel of the table compares the minimum lRMSDs of the ensembles including the ensemble generated by truncation-based selection; the middle panel compares the average lRMSDs of the ensembles, and the bottom one compares the standard deviation of lRMSDs of conformations in each ensemble to the known native conformation per target protein. The minimum, average, and standard deviations over the generated ensembles are provided as reference in Column 2 (top, middle, and bottom panels, respectively). The difference of the (lRMSD) minimum, average, or standard deviation in Ω_{red} over the corresponding quantity in Ω_{gen} is reported in each setting.

Focusing on the *Diff.* columns listing differences in minimum lRMSDs, it is clear that truncation selection performs the worst in this regard; differences in minimum lRMSD range

from 0.73Å to 5.12Å (see Column 12 of the top panel in Table 5.3). This means in the worst case, the best conformation kept by truncation selection is 5.12Å further away from the native conformation than the best conformation in the original ensemble. Truncation-based selection cannot maintain the quality of original ensemble.

In contrast, in the case of GMM and k-means, differences in minimum lRMSD (see Columns 4 and 6) are all 0Å. The differences in minimum lRMSD for gmx-cluster-usr range from 0Å to 0.11Å (see Column 10); for hierarchical clustering, the range is from 0Å to 1.12Å (see Column 8). This means that the conformations closest to the native conformations are always retained in the ensembles reduced by k-means and GMM. Not surprisingly, the slight increase in the differences when using gmx-cluster-usr and hierarchical clustering is the consequence of more drastic reduction in size of the reduced ensemble when using these two clustering algorithms over k-means or GMM.

The comparison shown in the middle panel of Table 5.3 indicates very little difference between the generated and reduced ensembles in terms of average lRMSDs. Column 4 shows the differences in average lRMSDs for k-means, which range from 0.41Å to 0.60Å. Column 6 shows an overall similar range for GMM (0.36Å to 0.50Å). Average lRMSD differences for hierarchical clustering range from 0.02Å to 0.61Å, as shown in Column 8. Column 10 shows that the differences in average lRMSD for gmx-cluster-usr range from 0.59Å to 1.04Å.

The comparison of differences on lRMSDs standard deviation for k-means is shown in Column 4 on the bottom panel and vary from 0.02Å to 0.26Å. These values are slightly different for GMM, ranging from 0.03Å to 0.25Å(see Column 6). As in the minimum lRMSD comparison, the differences obtained by gmx-cluster-usr and hierarchical clustering are slightly larger. Differences in standard deviation range from 0.23Å to 0.53Å for gmx-cluster-usr (shown in Column 10) and from 0Å to 0.36Å (with less than 0.1Å on 7/10 targets; shown in Column 8) for hierarchical clustering.

Similar observations can be extracted from Table 5.4, which shows the performance over the CASP dataset. Table 5.4 confirms again that truncation selection loses the quality of

Table 5.3: Comparison of minimum, average, and standard deviation of lRMSDs (to the known native conformation) of conformations in the Ω_{gen} and Ω_{red} ensembles of each target in the benchmark dataset. Comparison of minimum lRMSDs includes the ensemble reduced via truncation selection. Differences between the minimum, average, and standard deviation obtained over Ω_{red} from those obtained over Ω_{gen} are also related.

	Minimum lRMSD (Å)									
		K-means		**GMM**		**Hierarchical**		**Gmx-usr**		**Truncation**
PDB Id	Ω_{gen}	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.
1ail	3.64	3.64	0	3.64	0	3.64	0	3.64	0	4.37	0.73
1bq9	5.42	5.42	0	5.42	0	5.47	0.05	5.47	0.05	7.31	1.89
1c8ca	4.43	4.43	0	4.43	0	4.43	0	4.43	0	7.86	3.43
1cc5	5.4	5.4	0	5.4	0	6.52	1.12	5.4	0	7.85	2.45
1dtja	4.19	4.19	0	4.19	0	4.19	0	4.19	0	9.31	5.12
1hhp	11	11	0	11	0	11.29	0.29	11.02	0.02	12.88	1.88
1tig	5.34	5.34	0	5.34	0	5.45	0.11	5.45	0.11	6.59	1.25
2ezk	3.41	3.41	0	3.41	0	3.41	0	3.41	0	5.09	1.68
2h5nd	10.32	10.32	0	10.32	0	10.32	0	10.32	0	11.9	1.58
3gwl	4.85	4.85	0	4.85	0	4.85	0	4.85	0	7.81	2.96

	Average lRMSD (Å)								
		K-means		**GMM**		**Hierarchical**		**Gmx-usr**	
PDB Id	Ω_{gen}	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.
1ail	10.42	10.89	0.47	10.85	0.43	10.19	0.23	11.17	0.75
1bq9	9.76	10.18	0.42	10.17	0.41	9.74	0.02	10.39	0.63
1c8ca	12.29	12.75	0.46	12.72	0.43	12.25	0.04	12.96	0.67
1cc5	12.53	12.94	0.41	12.89	0.36	12.5	0.03	13.12	0.59
1dtja	11.87	12.35	0.48	12.29	0.42	11.95	0.08	12.5	0.63
1hhp	15.56	16.06	0.5	16.03	0.47	15.85	0.29	16.31	0.75
1tig	12.85	13.36	0.51	13.31	0.46	12.81	0.04	13.81	0.96
2ezk	10.17	10.77	0.6	10.72	0.55	10.78	0.61	11.21	1.04
2h5nd	15.79	16.24	0.45	16.2	0.41	16.16	0.37	16.51	0.72
3gwl	12.44	13.02	0.58	12.94	0.5	12.68	0.24	13.34	0.9

	Standard Deviation lRMSD (Å)								
		K-means		**GMM**		**Hierarchical**		**Gmx-usr**	
PDB Id	Ω_{gen}	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.
1ail	3.11	3.2	0.09	3.17	0.06	3.11	0	3.49	0.38
1bq9	1.78	1.76	0.02	1.89	0.11	1.89	0.11	2.11	0.33
1c8ca	2.18	2.23	0.05	2.22	0.04	2.15	0.03	2.45	0.27
1cc5	2.31	2.34	0.03	2.34	0.03	2.35	0.04	2.54	0.23
1dtja	2.08	2.31	0.23	2.27	0.19	2.15	0.07	2.44	0.36
1hhp	1.82	1.94	0.12	1.93	0.11	1.9	0.08	2.08	0.26
1tig	3.22	3.34	0.12	3.34	0.12	3.21	0.01	3.72	0.5
2ezk	2.41	2.67	0.26	2.66	0.25	2.77	0.36	2.94	0.53
2h5nd	2.03	2.22	0.19	2.21	0.18	2.3	0.27	2.46	0.43
3gwl	2.9	3.02	0.12	3.01	0.11	2.99	0.09	3.22	0.32

the original ensemble in the reduced one. The quality of the reduced ensemble is preserved by all clustering algorithms, and the best results belong to k-means and GMM. All four clustering algorithms produce ensembles that have small differences in average lRMSDs and perform comparably in terms of standard deviation.

Greater detail can be inferred from Figure 5.3, which shows results over a selected target protein (with native conformation under PDB id 1ail). Figure 5.3 shows the actual distribution of conformation lRMSDs from the known native conformation for the Ω_{gen} ensemble along with the ensembles Ω_{red} reduced via k-means, GMM, gmx-cluster-usr, and hierarchical clustering. Figure 5.3 shows that conformations with similar relative frequencies of lRMSDs as in Ω_{gen} are included in the reduced ensembles identified by each clustering algorithm.

Visually Comparing Distributions of lRMSDs and Energies Pre- and Post Reduction

We now compare the Ω_{gen} and Ω_{red} ensembles visually on target proteins in terms of Rosetta *score*4 energies and lRMSDs to the native conformation. Here we show one representative landscape on each dataset (benchmark and CASP) that illustrates the behavior of each of the clsutering algorithms. Conformations in Ω_{gen} are drawn in purple, while those in the Ω_{red} ensemble are drawn in green. Figure 5.4 does so for the benchmark dataset, and Figure 5.5 does so for the CASP dataset.

Figures 5.4 and 5.5 show that the reduced ensemble Ω_{red} includes conformations from all the regions in the conformation space populated by the original ensemble Ω_{gen}. All the purple dots being occluded by the superimposition in the k-means and GMM case visually makes the case that these two clustering algorithms perform better than gmx-cluster-usr and hierarchical clustering. This is not surprising, as k-means and GMM preserve more of the original ensemble.

Table 5.4: Comparison of minimum, average, and standard deviation of distribution of lRMSDs (to the known native conformation) of conformations in the Ω_{gen} and Ω_{red} ensembles of each target in the CASP dataset. Comparison of minimum lRMSDs includes the ensemble reduced via truncation selection. Differences between the minimum, average, and standard deviation obtained over Ω_{red} from those obtained over Ω_{gen} are also related.

	Minimum lRMSD (Å)										
		K-means		**GMM**		**Hierarchical**		**Gmx-usr**		**Truncation**	
CASP Id	Ω_{gen}	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.
T0859-D1	11.37	11.37	0	11.37	0	11.96	0.59	11.96	0.59	13.12	1.75
T0886-D1	7.96	7.96	0	7.96	0	8.73	0.77	8.73	0.77	11.24	3.28
T0892-D2	7.71	7.71	0	7.71	0	8.28	0.57	7.71	0	9.11	1.4
T0897-D1	10.18	10.18	0	10.18	0	10.64	0.46	10.64	0.46	11.62	1.44
T0898-D2	7.51	7.51	0	7.51	0	7.51	0	7.51	0	8.68	1.17
T0953s1-D1	6.14	6.14	0	6.14	0	6.29	0.15	6.29	0.15	8.18	2.04
T0953s2-D3	7.13	7.13	0	7.13	0	7.24	0.11	7.24	0.11	8.17	1.04
T0957s1-D1	7.65	7.65	0	7.65	0	7.76	0.11	7.76	0.11	9.39	1.74
T0960-D2	7.26	7.26	0	7.26	0	7.26	0	7.26	0	8.12	0.86
T1008-D1	3.85	3.85	0	3.85	0	3.85	0	3.85	0	5.67	1.82

	Average lRMSD (Å)								
		K-means		**GMM**		**Hierarchical**		**Gmx-usr**	
CASP Id	Ω_{gen}	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.
T0859-D1	17.47	17.64	0.17	17.63	0.16	17.49	0.02	17.78	0.31
T0886-D1	13.16	13.66	0.5	13.67	0.51	13.31	0.15	13.82	0.66
T0892-D2	14.81	15.49	0.68	15.43	0.62	15.02	0.21	15.71	0.9
T0897-D1	17.3	17.84	0.54	17.81	0.51	17.54	0.24	18.04	0.74
T0898-D2	11.56	11.72	0.16	11.71	0.15	11.63	0.07	11.86	0.3
T0953s1-D1	11.98	11.74	0.24	11.73	0.25	11.81	0.17	11.66	0.32
T0953s2-D3	13.28	13.89	0.61	13.88	0.6	13.48	0.2	14.06	0.78
T0957s1-D1	14.96	15.49	0.53	15.44	0.48	15.13	0.17	15.74	0.78
T0960-D2	12.63	13.07	0.44	13.07	0.44	12.93	0.3	13.27	0.64
T1008-D1	11.77	12.36	0.59	12.46	0.69	11.9	0.13	12.67	0.9

	Standard Deviation lRMSD (Å)								
		K-means		**GMM**		**Hierarchical**		**Gmx-usr**	
CASP Id	Ω_{gen}	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.	Ω_{red}	Diff.
T0859-D1	1.83	1.84	0.01	1.9	0.07	1.91	0.08	1.99	0.16
T0886-D1	1.67	1.82	0.15	1.79	0.12	1.69	0.02	1.98	0.31
T0892-D2	3.02	2.98	0.04	2.97	0.05	3.04	0.02	3.25	0.23
T0897-D1	2.75	2.74	0.01	2.73	0.02	2.76	0.01	2.92	0.17
T0898-D2	1.03	1.17	0.14	1.16	0.13	1.14	0.11	1.26	0.23
T0953s1-D1	1.51	1.51	0	1.51	0	1.5	0.01	1.49	0.02
T0953s2-D3	1.9	1.84	0.06	1.83	0.07	1.86	0.04	2.06	0.16
T0957s1-D1	3.04	3.04	0	3.03	0.01	3.04	0	3.23	0.19
T0960-D2	1.85	1.94	0.09	1.95	0.1	1.92	0.07	2.05	0.2
T1008-D1	3.7	3.72	0.02	3.69	0.01	3.74	0.04	3.94	0.24

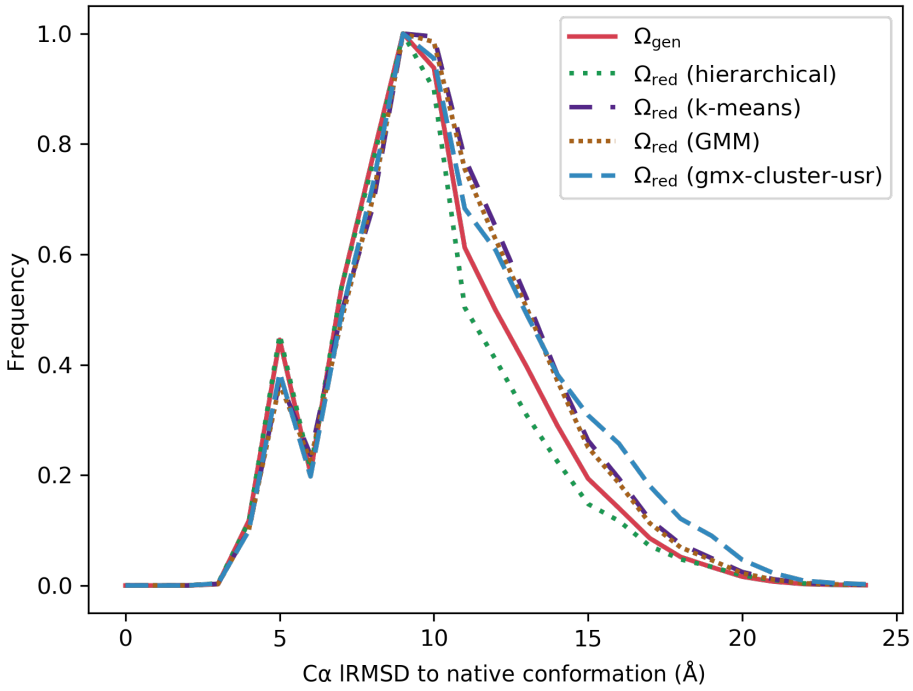


Figure 5.3: The distribution of conformation lRMSDs from the native conformation is shown for the Ω_{gen} ensemble (in red) and the reduced Ω_{red} ensembles obtained via k-means (purple), GMM (brown), hierarchical clustering (green), and gmx-cluster-usr (in blue). Results are shown for a representative target protein with native conformation under PDB id 1ail.

5.1.7 Discussion

The presented results make the case that all four clustering algorithms are able to drastically decrease the conformation ensemble size while preserving the quality and diversity of the original ensemble. GMM and k-means behave equally well in this regard, while gmx-cluster-usr and hierarchical clustering reduces the size of the ensembles more significantly and in response is also more prone to sacrificing quality.

Experience in molecular modeling informs that the choice of representation is often key to the success of a method. Here we provide further analysis into what the USR features are capturing. We do via a simple correlation analysis, where we compare the distribution of the lRMSDs versus USR scores of computed conformations to the native conformation.

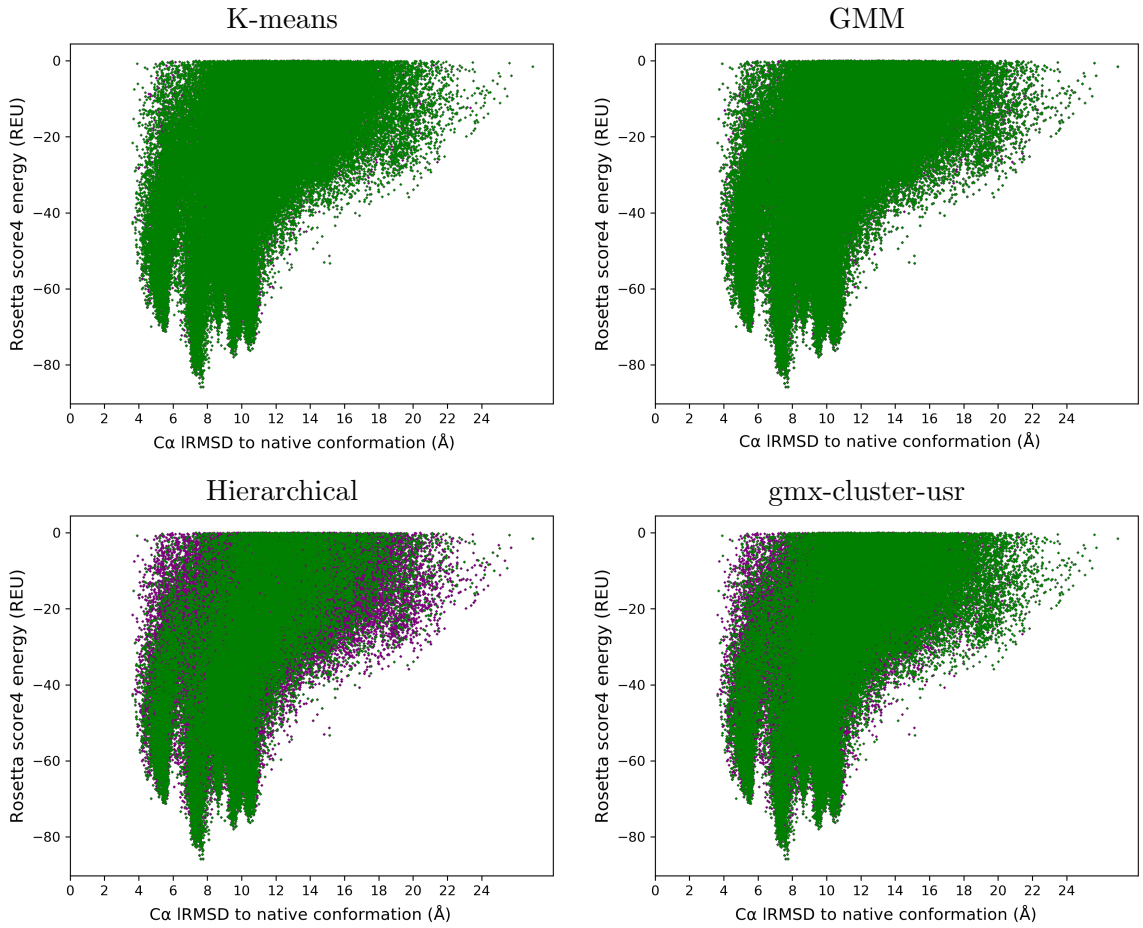


Figure 5.4: Benchmark Dataset: A representative target (with known native conformation under PDB id 1ail) is selected. Conformations in the Ω_{gen} ensemble are plotted in purple in terms of their lRMSD (Å) from the native conformation (x-axis) versus their Rosetta score4 energies (y-axis) measured in Rosetta Energy Units (REUs). Conformation in the Ω_{red} ensemble are superimposed in green.

USR score is calculated as the Euclidean distance in the 12-dimensional USR feature space for two conformations.

Figure 5.6 plots the distributions against each-other for two targets that are representatives of the Pearson's correlation coefficients obtained over targets in the benchmark and CASP datasets. Specifically, Figure 5.6(a) shows a correlation of 0.80 that is representative of what is observed over the benchmark dataset; Figure 5.6(b) shows a correlation of 0.74

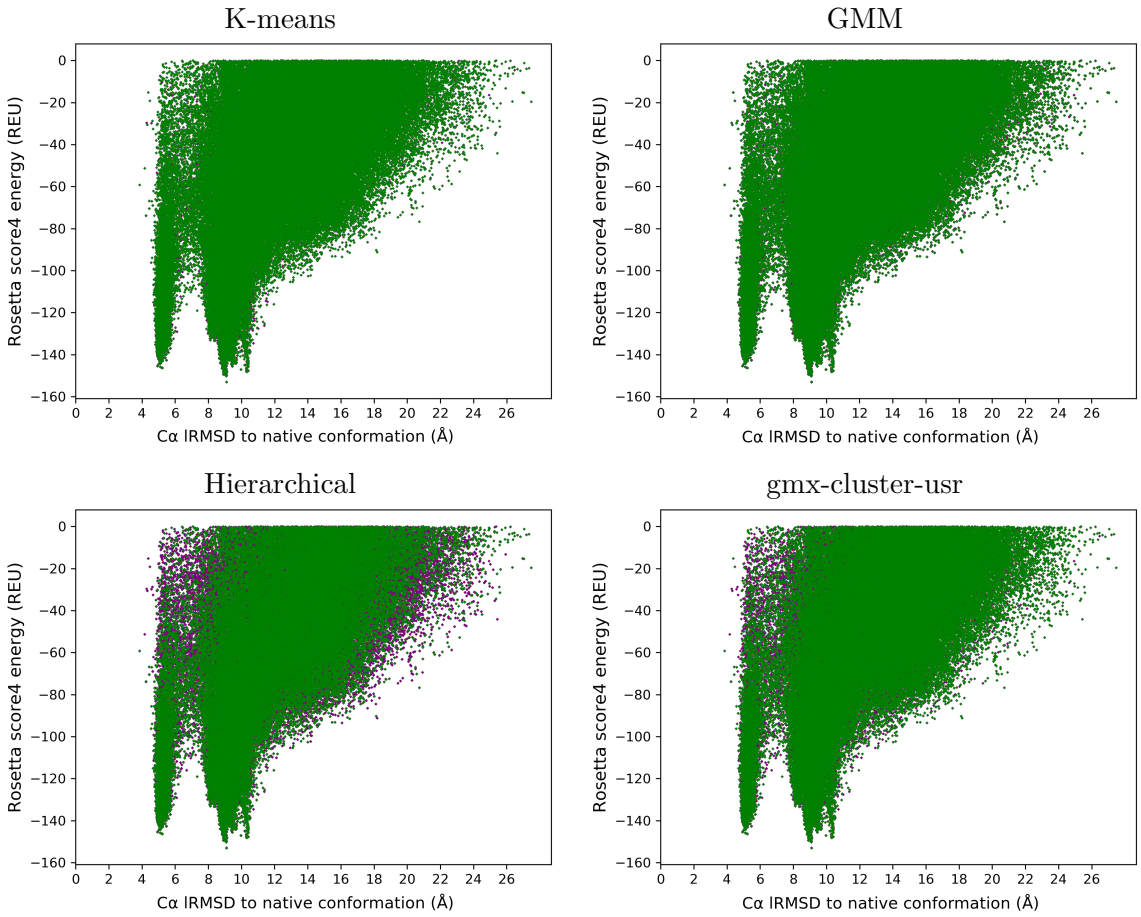
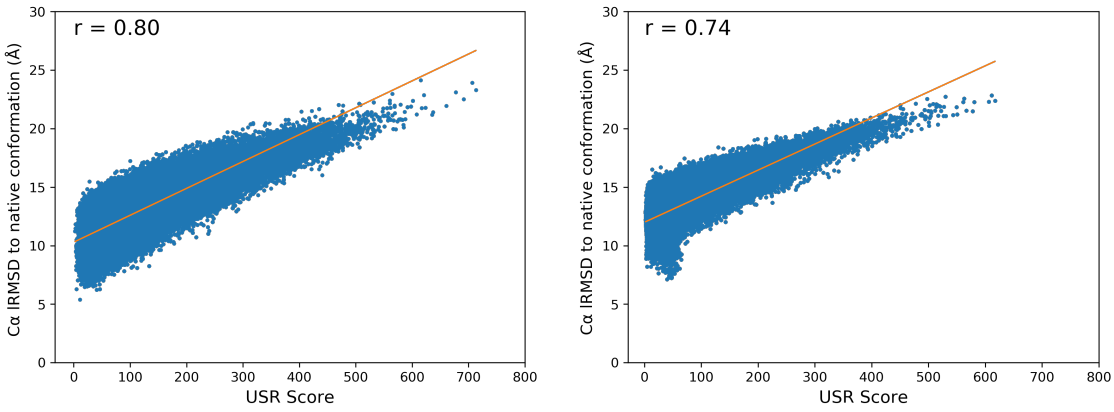


Figure 5.5: CASP Dataset: A representative protein (T1008-D1) is selected. Conformations in the Ω_{gen} ensemble are plotted in purple in terms of their lRMSD (Å) from the native conformation (x-axis) versus their Rosetta score4 energies (y-axis) measured in Rosetta Energy Units (REUs). Conformation in the Ω_{red} ensemble are superimposed in green.

that is representative of what is observed over the CASP dataset.

The median correlation over the benchmark dataset is 0.80, and the median correlation over the CASP dataset is 0.755. The correlations (representing what is observed over each of the datasets, with few outliers) show that the USR score is informative and a good proxy for lRMSD; we recall that the USR representation is also invariant to rigid-body motions, unlike Cartesian coordinate-based representations. Altogether, these results inform that the choice of the USR-based representation of conformations is advantageous,



(a) (b)

Figure 5.6: Correlation between USR scores and lRMSDs to the native conformation of all conformations computed on a target protein in the (a) benchmark dataset (with native conformation under PDB id 1cc5) and (b) CASP dataset (with native conformation under CASP id T0953s2-D3.)

allowing clustering algorithms to capture important conformational differences that are then retained in the reduced ensemble by the selector.

The findings presented in this chapter suggest that it is possible to significantly reduce the number of generated conformations without sacrificing quality and diversity of the ensemble. A three-step approach relying on featurization, clustering, and selection is shown effective at doing so independent of the particular clustering algorithm employed. Various clustering algorithms are evaluated in the proposed approach.

5.2 Building Concise Maps of Protein Conformation Space

The work presented in this section has been published in [98, 99]. We demonstrated in Section 5.1 that it is possible for the generated conformation ensemble by a conformation ensemble generation algorithm to be significantly reduced in size while retaining conformation quality. In this section, our goal is to get a conformation ensemble generation algorithm to generate such a reduced ensemble during its execution. To do so, we propose to equip

conformation ensemble generation algorithms with an evolving reduced-size memory of the protein conformation space that they explore. This is inspired by robot motion planning algorithms [100] and their adaptations in robotics-inspired algorithms for modeling molecular motions [21, 83], as detailed later in the section. We introduce an evolving, granularity-controllable map of the protein conformation space that makes use of low-dimensional representations of protein conformations. The map reduces the storage requirement drastically but provides similar quality of a map that would hold all the conformations ever generated by an algorithm. Our evaluations make the case that integrating a map of the protein conformation space is a promising mechanism to develop feasible conformation ensemble generation algorithms.

5.2.1 Choice of Conformation Ensemble Generation Algorithm

We utilize HEA as described in Section 2.2.2 as a vehicle to implement and demonstrate the power of the proposed mechanism of an evolving map of the protein conformation space. We note that the map can be integrated in any conformation ensemble generation algorithm. We choose HEA as it has been shown to have higher exploration capability than the SA-MMC conformational sampling employed in the Rosetta platform[41, 46] and the HEA generates hundreds of thousands of conformations that can be utilized to build a map of the protein conformation space.

5.2.2 Evolving Map of Protein Conformation Space

In this chapter, we equip the HEA with memory of the protein conformation space it explores during its execution. Without a map, one would resort to collecting all the individuals in the population over all the generations to constitute the conformation pool. Two key questions need to be addressed. First, how should the memory be implemented? We refer to this memory as a map from now on, viewing it as a map of the protein conformation space. Second, which individuals in the current population should be remembered for inclusion in the map?

The map needs to be a broad, sample-based representation of the protein conformation space explored by a conformation ensemble generation algorithm. To achieve this, we propose the use of two projection layers that allow selecting diverse yet good-quality individuals from the population in each generation. Quality is maintained via an energetic layer which introduces an energetic bias in the selection of individuals for inclusion in the map. The layer increases the likelihood of remembering low-energy individuals/conformations. Diversity is maintained via a geometric layer which introduces a geometric bias in the selection of individuals for inclusion in the map. The layer increases the likelihood of remembering conformationally-diverse individuals that represent different regions of the HEA-probed conformation space.

As mentioned earlier, the utilization of discretization layers is inspired by robot motion planning algorithms and their adaptations in robotics-inspired algorithms for modeling molecular motions. In these algorithms, the layers are employed to bias the exploration of a high-dimensional, continuous robot or molecular configuration space. Here, we propose a novel utilization of discretization layers in the concept of a map/memory for conformation ensemble generation algorithms. Next we detail the energetic and geometric layers and their utilization.

Energetic Layer

The energetic layer is one-dimensional grid defined over Rosetta score4 in the range $[E_{\min}, 0]$. The upper bound of 0 acknowledges that conformations with positive energy are infeasible (low-quality); indeed, conformation ensemble generation algorithms generate conformations with negative energies early in their exploration process. The lower bound $E_{\min}$ is set to -200; This is informed by previous work and experiments, where we have observed that the *score*4 energy of a good-quality conformation is well above -200 Rosetta Energy Units (REUs) on target proteins of different lengths and folds[41, 46, 74, 76, 101]. In the grid, each interval is set to a small value of 2 REUs (thus totaling 100 energy intervals over the employed range) to ensure good granularity. Conformations that fall in the same interval

are deemed to be energetically similar.

Geometric Layer

We associate a three-dimensional (geometric) grid with each energy interval in the energetic layer. The dimensions of the grid are 3 features that capture/summarize different aspects of molecular shape. We borrow here from the Ultrafast Shape Recognition (USR) metrics introduced originally in Ref.[82] to summarize molecular shapes for fast searching. Specifically, the features we employ to summarize a protein conformation and map it to a cell in the grid are first momenta of the atomic distance distributions from 3 different reference points in a conformation. The first reference point is the molecular centroid (ctd). The resulting first moment of the distance distribution of all atoms from the ctd gives the first axis/dimension in the geometric grid. The second reference point is the point farthest from the centroid (fct). Similarly, it gives the second axis in the grid. Finally, the third reference point is the one farthest from the fct (ffct). The resulting first moment of the distance distribution of all atoms from the ffct gives the third axis. Essentially, a conformation is summarized with three coordinates in this way. To determine in which cell/cube a conformation falls, we consider only integer levels. So, each cube on the grid is defined by 3 integer coordinates. All conformations that fall in the same cube are deemed to be conformationally/geometrically similar.

Layer-based Selection of Conformations for Inclusion in the Map

We note that the map is a list of conformations. No additional storage is maintained, however, beyond the grids described above. Specifically, only one conformation is retained per cube of the geometric grid, and the map itself consists of the non-empty cubes of the geometric grid. Fig. 5.7 summarizes how computed conformations are selected to be remembered/included in the map. The energetic and geometric layers described above are utilized to make this decision for each conformation. For an EA-based conformation ensemble generation algorithm (as in our case), each improved offspring is subjected to

this decision process; parents in the initial population are also considered. Specifically, an individual or, more broadly, a computed conformation is first evaluated based on its energy (score4 in our case) and mapped to an energy interval on the 1d energy grid described above. Once mapped to an energy interval, the conformation is evaluated based on the three USR-based features described above, and the conformation is mapped to a cube in the 3d geometric grid. If the cube is empty, this is an indication that the conformation populates an unexplored region of the conformation space and should be remembered. Therefore, the conformation is included in the map. Otherwise, if the cube is not empty, this means that the region captured by the cube has already been explored. Two courses of action are reasonable here. The conformation under consideration can be ignored, as it is both geometrically- and energetically-similar (within 2 REUs) to the conformation already stored in the cube. The other is to replace the conformation in the cube if the one being considered has a lower energy. In this work, we implement the second option.

5.2.3 Implementation Details

The population size in the employed algorithm is set to $p = 100$. The elitism rate in the selection operator is set to 25%, as in Ref.[41]. The algorithm is executed 3 times on each target protein's amino-acid sequence to account for stochasticity. Each run has a fixed budget of $10,000,000$ energy evaluations. This budget translates to $2-9$ hours on a 2.6GHz Intel Xeon E5-2670 CPU with 100GB of RAM. The variation in execution time is primarily governed by the length (number of amino acids) of a target protein sequence. The algorithm is implemented in Python and interfaces with the PyRosetta library.

5.2.4 Results

We carry out our evaluation on 18 targets in the benchmark dataset and 10 targets in the CASP dataset. For the purpose of evaluation, the ensemble of conformations consisting of individuals from every population in the HEA without the map is referred to as the *original pool*. The ensemble of conformations retained in the map resulting from running HEA with

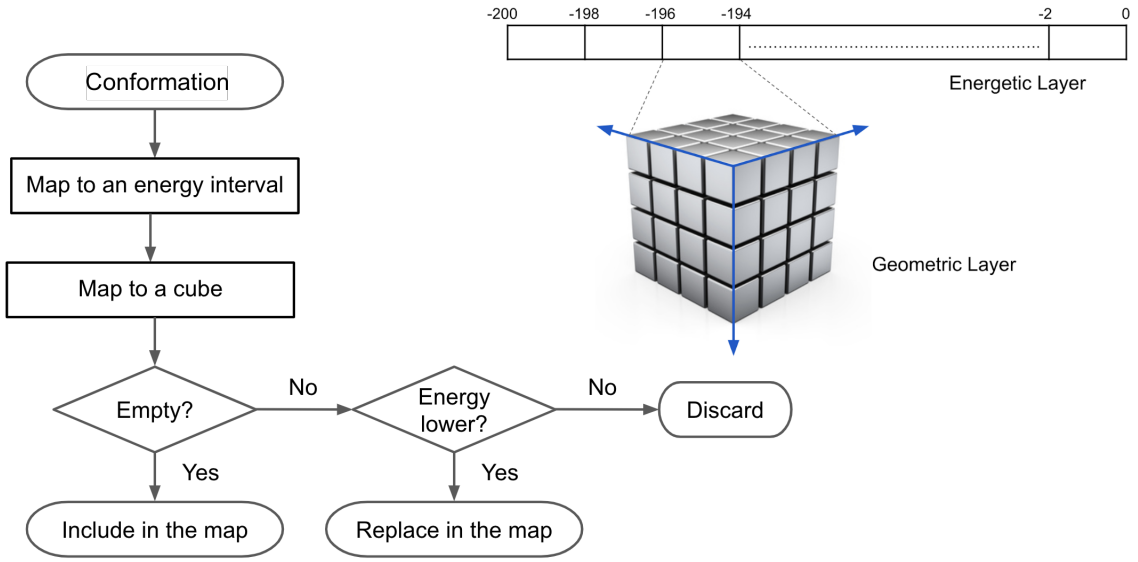


Figure 5.7: The schematic summarizes the process via which a generated conformation is considered for inclusion in the map. The decision considers both the energetic and geometric layer. In this manner, the map evolves during the course of a conformation ensemble generation algorithm and stores structurally-diverse, yet low-energy conformations.

the map is referred to as the *reduced pool*. The evaluation presented here is over combined results (over the 3 runs). The *original* and the *reduced pool* are compared in terms of size and quality. To compare quality, as the map never discards lower-energy conformations, we focus on the proximity of conformations to the known native conformation of a target sequence.

Conformation Ensemble Reduction versus Conformation Quality

Table 5.5 shows the comparison between the original and reduced pools in terms of size versus quality. The comparison first focuses on the benchmark dataset. The PDB identifiers (IDs) of the known native conformation for each of the target sequences in the benchmark dataset are listed in Column 1. The size (number of conformations) of the original pool and reduced pool/map are compared in Columns 2-3. The reduction afforded by the reduced

pool is shown in Column 4 as a percentage. The reduction is drastic. More than 90% reduction is achieved in the reduced pool in 10/18 test cases. The other 8 cases yield reductions that exceed 80% with a minimum decrease of 81.6%. The most dramatic reduction in size occurs in target sequences with PDB ID 1bq9 and 1dtdb; a reduction of 96.3% is reported in these two cases, which is almost a 27-fold reduction over the original pool.

Columns 5-6 in Table 5.5 compare the lowest lRMSD to the known native conformation over all the conformations in the original pool versus the reduced pool for each target. The difference between these two values (increase of lRMSD in the reduced pool from the original pool) for each target is reported in Column 7, indicating that dramatic reductions in size do not sacrifice conformation quality. The difference in lRMSD to the native between the reduced and original pools is 0Å for 9/18 targets. This means that on 50% of the targets, the reduction in size inflicts no penalty on the quality of the conformations. Indeed, the difference is less than 1Å for all targets, with a maximum difference of 0.7Å reached on the target with PDB ID 1aoy. This demonstrates the utility of the map in retaining conformations of good quality while drastically reducing the number of retained conformations.

Table 5.6 relates a similar evaluation for each of the CASP domains denoted by their corresponding identifiers in Column 1. The size (number of conformations) of the original pool and reduced pool (the map) are compared in Columns 2-3. The reduction in size is shown in Column 4 as a percentage. More than 90% reduction is achieved by the reduced pool in 5/10 test cases. Except for one case (target with identifier T0897-D1, where the reduction is 73.8%), the reduction exceeds 80% in all targets.

Columns 5-6 in Table 5.6 compare the lowest lRMSD to the native conformation over all the conformations in the original pool versus the reduced pool for each of the CASP targets. The difference between these two values (increase of lRMSD in the reduced pool from the original pool) is reported for each target in Column 7. The difference shown in Column 8 is 0Å for 6/10 test cases; this indicates that on 60% of the targets, the reduction in size comes at no cost to conformation quality. The difference is less than 1Å for all targets, with a

Table 5.5: Comparison of size reduction versus quality retainment in the original versus the reduced Pool on the benchmark dataset. Column 1 shows the PDB IDs of the known native conformation of each target sequence. Columns 2-4 juxtapose the sizes of the original and reduced pools. Columns 5-7 compare the quality of the pools in terms of their lowest lRMSD from the known native conformation for each target sequence.

	Size			**Lowest lRMSD**		
PDB ID	Original Pool	Reduced Pool	Reduction (%)	Original Pool (Å)	Reduced Pool (Å)	Difference (Å)
1ail	680,637	56,740	91.7	2.4	2.7	0.3
1aoy	628,010	75,413	88.0	4.2	4.9	0.7
1bq9	485,490	18,114	96.3	5.3	5.5	0.2
1c8ca	452,893	35,559	92.1	6.8	7.0	0.2
1cc5	559,247	39,805	92.9	5.8	5.8	0
1dtdb	180,667	6,722	96.3	8.0	8.3	0.3
1dtja	309,725	24,593	92.1	3.6	3.6	0
1hhp	168,334	17,290	89.7	10.8	10.8	0
1hz6a	556,254	56,578	89.8	2.7	2.7	0
1isua	336,770	17,361	94.8	6.9	6.9	0
1sap	678,521	65,238	90.4	5.7	6.1	0.4
1tig	496,741	56,180	88.7	6.2	6.2	0
1wapa	341,909	33,793	90.1	7.8	8.0	0.2
2ci2	331,022	23,497	92.9	4.3	4.3	0
2ezk	568,350	85,080	85.0	4.0	4.3	0.3
2h5nd	426,385	75,997	82.2	10.5	10.5	0
2hg6	418,324	76,989	81.6	11.4	11.6	0.2
3gwl	483,216	63,073	86.9	4.7	4.7	0

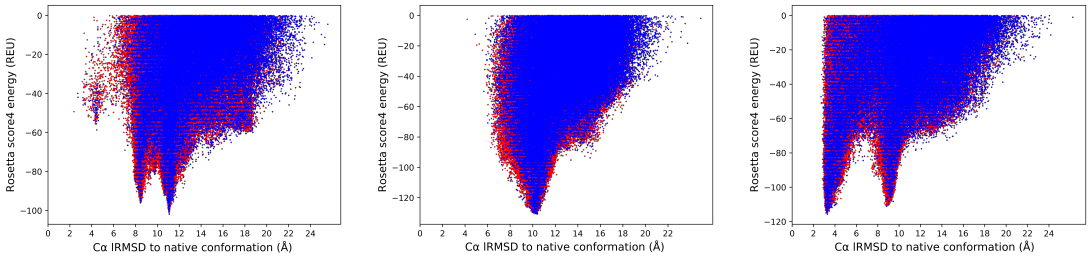
maximum difference of 0.7Å (for the target with identifier T0957s1-D1). These results agree with those obtained on the benchmark dataset and further confirm that the map provides drastic reduction in storage while retaining good-quality conformations.

Visual Comparison of Conformation Spaces

We can visualize the conformation space remembered in the map as follows. Each conformation in the map (to which we have been referring as the reduced pool) can be plotted with two coordinates, its lRMSD from the known native conformation and its Rosetta *score*4 energy. We can do the same for all conformations in the original pool (HEA without the

Table 5.6: Comparison of size reduction versus quality retainment in the original versus the reduced Pool on the CASP dataset. Column 1 shows the target CASP identifiers. Columns 2-4 juxtapose the sizes of the original and reduced pools. Columns 5-7 compare the quality of the pools in terms of their lowest lRMSD from the known native conformation for each target.

CASP ID	**Size**			**Lowest lRMSD**		
	Original Pool	Reduced Pool	Reduction (%)	Original Pool (Å)	Reduced Pool (Å)	Difference (Å)
T0859-D1	404,048	79,130	80.4	11.7	11.7	0
T0886-D1	250,607	13,836	94.5	7.5	7.5	0
T0892-D2	279,911	51,104	81.7	8.3	8.3	0
T0897-D1	319,782	83,720	73.8	10.8	10.8	0
T0898-D2	353,708	11,241	96.8	7.4	7.9	0.5
T0953s1-D1	167,707	7,361	95.6	7.3	7.3	0
T0953s2-D3	293,291	26,515	91.0	9.0	9.1	0.1
T0957s1-D1	344,604	60,775	82.4	7.2	7.9	0.7
T0960-D2	136,787	10,083	92.6	7.0	7.5	0.5
T1008-D1	656,333	79,989	87.8	2.5	2.5	0



1ail 1aoy 1hz6a

Figure 5.8: Each conformation is plotted with two coordinates, its lRMSD from the native conformation on the x-axis, and its Rosetta *score*4 energy on the y-axis. Conformations in the original pool are drawn in red, whereas those in the reduced pool are drawn in blue. The targets are indicated above each plot via the PDB IDs of their known native conformations. This figure shows the results for three selected targets in the benchmark dataset.

map).

Fig. 5.8 and Fig. 5.9 provide this visualization for three selected targets each in the benchmark dataset and the CASP dataset, respectively. In these figures, the conformations

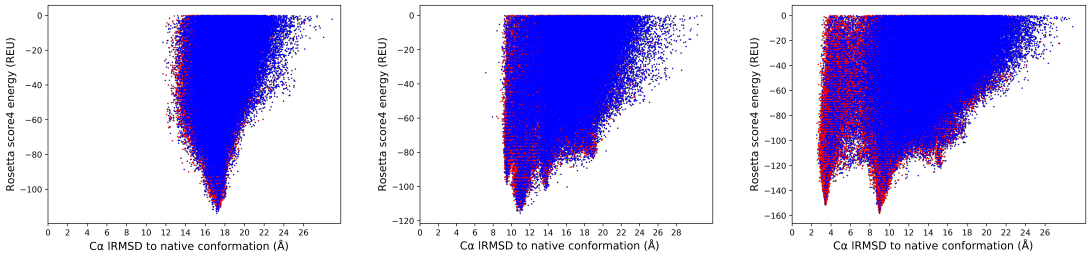


Figure 5.9: Here we visualize the original and the reduced pool for three selected targets in the CASP dataset. The targets are indicated above each plot via their CASP identifiers. Each conformation is plotted with two coordinates, its lRMSD from the native conformation on the x-axis, and its Rosetta *score*4 energy on the y-axis. Conformations in the original pool are drawn in red, whereas those in the reduced pool are drawn in blue.

in the original pool are drawn in red, whereas those in the reduced pool are drawn in blue. It is apparent from examining these plots that the map (reduced pool) remembers conformations from every region in the conformation space (original pool) probed by the HEA. It is also evident that the map captures all local minima well.

Finally, Fig. 5.10 juxtaposes the best conformation (with lowest lRMSD to a known native conformation) among all conformations in the original pool with the best conformation in the reduced pool by superimposing them over the known native conformation. This is done for the target protein with known native conformation under PDB ID 1aoy. The reason for choosing this target is due to the largest difference (0.7Å) in lRMSD between the reduced and original pools reported in Table 5.5. The CCP4mg molecular graphics software[1] is used to perform all conformation rendering. As can be seen, the conformations are very similar.

5.2.5 Summary

In this section, we present a mechanism by which one can equip conformation ensemble generation algorithms with memory of the protein conformation space that they explore. Specifically, generated conformations are considered for inclusion in a map. The map stores

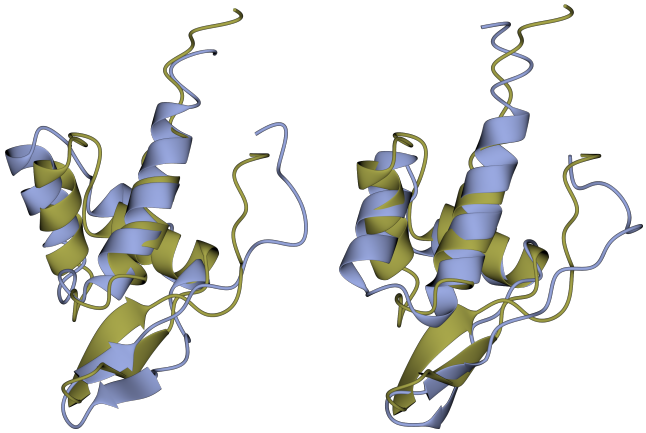


Figure 5.10: The best conformation (lowest lRMSD to the native conformation) among all HEA-generated conformations (in the original pool) is rendered in blue on the left. The best conformation in the map (reduced pool) is rendered in blue on the right. Each is superimposed over the known native conformation (PDB ID 1aoy), which is rendered in olive.

non-redundant conformations and utilizes low-dimensional (energetic and geometric) representations of a protein conformation that facilitate computationally-efficient comparison of conformations. The granularity of the map can be controlled by increasing/decreasing the number of energy intervals in the energetic layer and the number of cubes in the geometric layer. Increasing granularity would result in higher storage demands but also provide great detail. Controlling granularity allows balancing between the demand on storage and the amount of desired detail. Evaluation on diverse targets shows that drastic reductions in storage do not sacrifice conformation quality. While the results presented here have integrated the proposed map in an evolutionary algorithm, the map can be easily integrated in any conformation ensemble generation algorithm, as it allows making decisions on a per conformation basis.

5.3 Guiding Conformation Ensemble Generation Algorithms with Maps

In Section 5.2, we equipped conformation ensemble generation algorithms with an evolving map of the conformation space to promote their feasibility. In this section, we investigate if we can guide such algorithms with the map at the same time to enhance exploration of the conformation space.

While use of a memory/map of the search space has been investigated in literature for protein modeling, they have not been utilized to guide the search to achieve more exploration of the search space for conformation ensemble generation. Such works either attempt to use the memory to attain a mapping of the the conformation space [102] or to store the best solutions [103–106]; not to guide an optimization algorithm. Moreover, the memories that are used in such works generally do not reduce dimensionality of the stored individuals [103–106] and would be infeasible for conformation ensemble generation as a conformation has hundreds of dimensions and a huge number of conformations are generated. Work in [102], however, uses a memory mechanism that compares distances in a reduced dimension of principal components (PC) space. But, such a map would be infeasible for the generated conformations as our experiments show low variance retained for the generated conformations for even 20 PCs. Works beyond protein modeling generally do not reduce dimensions [107–109] for the memory and would be inefficient if used as a memory of the explored spaces for de novo conformation ensemble generation. In addition, there is no evidence that the memories used in these works provide a proper representation of the explored search space.

In this work, we intend to build over our work described in Section 5.2, where we introduced an evolving map of the protein conformation space that makes use of low-dimensional representations of protein conformations (uses only 3 dimensions to capture the shape of a conformation), stores non-redundant diverse conformations, and sums up the already explored spaces well. Here, we propose to use this map as a guide for the conformation

ensemble generation algorithms to discourage conformational sampling in already sampled spaces and encourage exploration of the unknown parts of the conformation space.

The process of guiding by the map to avoid explored regions requires careful consideration. While modifying the fitness function to decrease fitness of individuals close to the explored spaces, as in [107, 110], is appealing, in our case the fitness functions are carefully constructed with domain-specific insights and it is not clear how to modify them. Another notable idea is to perform adaptive variation consulting the memory [108] which is also challenging as described later in Section 6.2. Therefore, we propose to periodically exclude similar individuals to the already explored individuals during selection consulting the map of the explored conformation space.

First, we focus on HEA equipped with the map as described in Section 5.2. We then change its selection operator which selects the individuals to construct the next generation. We propose a new selection mechanism that consults the map to select individuals in a way that allows the conformational sampling process to sample diverse conformations from different parts of the conformation space.

5.3.1 Guiding with the Map

The selection operator in HEA is modified to allow consultation with the map to select individuals for the next generation. How often this consultation happens is governed by the consultation frequency f. In all the generations the map is not consulted, the selection operator works the same as the selection operator in HEA.

In any generation g, let the f generation earlier version of the ever-evolving map be denoted as MAP_{g-f}. The selection mechanism checks the MAP_{g-f} in every f generations during selection. The map consulted is always the f generation earlier version to provide the individuals in the MAP_{g-f} enough opportunities to reproduce and improve. This enables the algorithm to exploit the conformation space around these individuals. Starting with an empty selection pool, during each consultation, the parents and offspring that fall on empty cubes in the MAP_{g-f} are added to the selection pool. The parents and offspring that fall

on already occupied cubes are excluded from the selection pool as the conformation space around these individuals have already been explored.

After all the individuals are checked, two scenarios can occur. First, the selection pool contains more individuals than the population size. In this case, we apply truncation selection to bring the selection pool down to the population size. Second, the selection pool contains less individuals than the population size. In this case, we randomly select the rest of the individuals from the map and apply molecular fragment replacement of length 9 on them once to have bigger conformational change for exploration in the unknown parts of the landscape and get more diverse conformations.

When the above process is completed, the selection pool contains the same number of individuals as the population size. These individuals constitute the next generation.

5.3.2 Implementation Details

The population size p is set to 100 and the elitism rate for elitism-based truncation selection is set to 25%, as in [41]. As is commonly done for conformation ensemble generation algorithms, the termination criterion is set to a total budget of fitness/energy evaluations. The algorithm is executed for a fixed budget of 10M energy evaluations. The consultation frequency f is set to 15. The algorithm is implemented in python and interfaces with the PyRosetta library. The algorithm runs for $2-5$ hours on one Intel Xeon E5-2670 CPU with 2.6GHz base processing speed and 100GB of RAM. The runtime differs mainly because of different lengths of the target proteins. The algorithm is run 5 times on each target to account for the variance due to stochasticity.

5.3.3 Results

We carry out our evaluation on 10 targets each from both the benchmark and the CASP datasets. We refer to the algorithm described above as HEA-Map. HEA-Map is compared to HEA for a baseline comparison. We also compare HEA-Map to two other state-of-the-art conformation ensemble generation algorithms. One is Rosetta's SA-MMC based

conformation sampling algorithm. The other is a recent subpopulation EA, SP-EA$^+$ [101], described in Section 6.1, that aims to prevent premature convergence and retain diversity during optimization by evolving and maintaining multiple subpopulations.

The HEA-Map, HEA, and SP-EA$^+$ algorithms are run 5 times on each target sequence, and what we report here is their best performance over all 5 runs combined. Each run exhausts a fixed computational budget of $10M$ energy evaluations for a total of $50M$ energy evaluations for the 5 runs. Rosetta is run for $54M$ energy evaluations. As is practice in EAs for conformation ensemble generation, performance is measured by lowest reached energy and the lowest reached distance to the known native conformation of the target. We use lRMSD for the proximity measure.

To present a principled evaluation, we further strengthen our comparison with statistical significance tests. We utilize Fisher's and Barnard's exact tests for this purpose. To provide a complete picture, we also employ performance profiles for the results.

Evaluation on Benchmark Dataset

Table 5.7 shows the lowest *score*4 energy reached by each of the algorithms under comparison on the benchmark dataset. Table 5.7 shows that HEA-Map achieves lower energy than all other algorithms in 8/10 cases. In a head-to-head comparison, HEA-Map beats all other algorithms comfortably and achieves lower energy than Rosetta in 9/10 cases, than HEA in 8/10 cases, and than SP-EA$^+$ in 9/10 cases. Table 5.11(a) evaluates the 1-sided statistical significance tests of the performance of HEA-Map over the other algorithms. Table 5.11(a) shows that the performance improvements are statistically significant at 95% confidence level (p-values < 0.05) for both Fisher's and Barnard's tests.

Table 5.8 shows the lowest lRMSD to the native conformation reached by each of the algorithms under comparison on the benchmark dataset. Table 5.8 shows that HEA-Map achieves lower lRMSD than all other algorithms in 6/10 cases. In a head-to-head comparison, HEA-Map beats all other algorithms comfortably and achieves lower lRMSD than

Table 5.7: Comparison of the lowest energy obtained by each algorithm under comparison on each of the 10 benchmark targets is shown in Columns 4-7. The PDB ID of the known native, sequence length, and fold of each target are shown in Columns 1-3. The lowest energy value reached per target is marked in bold.

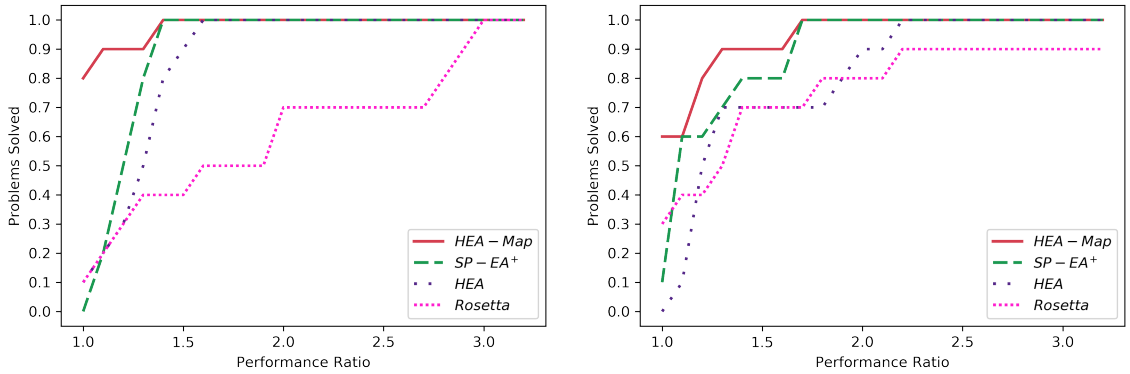
			Lowest Energy (REUs)			
PDB ID	Length	Fold	Rosetta	HEA	SP-EA$^+$	HEA-Map
1ail	73	α	-29.9	-56.1	-81.3	$\mathbf{-84.7}$
1bq9	53	β	-46.9	-50.5	-64.2	$\mathbf{-71.1}$
1c8ca	64	β	-101.4	-86.4	-78.3	$\mathbf{-105.7}$
1cc5	83	α	-82.5	-68.6	-76.4	$\mathbf{-93.7}$
1dtja	76	$\alpha+\beta$	-72.5	-82.2	-72.6	$\mathbf{-90.9}$
1hhp	99	β	$\mathbf{-106.3}$	-104.5	-83.5	-81.4
2ci2	83	$\alpha+\beta$	-37.8	$\mathbf{-109.8}$	-82.7	-108.8
2ezk	93	α	-51.1	-100.7	-135.2	$\mathbf{-138}$
2h5nd	123	α	-82.5	-129	-139.1	$\mathbf{-161.9}$
3gwl	106	β	-68.2	-100	-117.8	$\mathbf{-133.7}$

Rosetta in 7/10 cases, than HEA in 9/10 cases, and than SP-EA$^+$ in 7/10 cases. Table 5.11(b) evaluates the 1-sided statistical significance tests of the performance of HEA-Map over the other algorithms. Table 5.11(b) shows that the performance improvement over HEA is statistically significant at 95% confidence level (p-values < 0.05) for both Fisher's and Barnard's tests. Performance improvement over Rosetta and SP-EA$^+$ are not statistically significant at 95% confidence level but the p-values are close to 0.05.

Figure 5.11(a) shows the performance profiles of each algorithm over the benchmark dataset in terms of the lowest energy reached. Figure 5.11(a) shows that the probability of HEA-Map to be the optimal algorithm among these 4 algorithms is about 0.80, considerably more than any of the other algorithms. At $pr = 1.1$, HEA-Map succeeds for 90% targets. HEA-Map and SP-EA$^+$ reaches a success of 100% at a $pr = 1.4$, while HEA do so at $pr = 1.6$. Rosetta's performance profile rises very slowly and reaches 100% at $pr = 3.0$. Figure 5.11(b) relates a similar analysis focusing on the lowest lRMSD to the native conformation and shows that the probability of HEA-Map to be the optimal algorithm among these 4 algorithms is about 0.60, considerably more than any of the other algorithms. At

Table 5.8: Comparison of the lowest lRMSD to the native conformation obtained by each algorithm under comparison on each of the 10 benchmark targets is shown in Columns 4-7. The PDB ID of the known native, sequence length, and fold of each target are shown in Columns 1-3. The lowest lRMSD value reached per target is marked in bold.

			Lowest lRMSD (Å)			
PDB ID	Length	Fold	Rosetta	HEA	SP-EA$^+$	HEA-Map
1ail	73	α	4.5	1.4	**1.2**	1.4
1bq9	53	β	2.9	3	4.7	**2.8**
1c8ca	64	β	**2.2**	4.8	3.6	3.7
1cc5	83	α	**3.7**	4.7	4.7	4.4
1dtja	76	$\alpha+\beta$	**2.3**	4.2	2.5	2.8
1hhp	99	β	10.1	8.8	8.2	**7.8**
2ci2	83	$\alpha+\beta$	5.8	3.7	3.5	**3.3**
2ezk	93	α	3.6	3.4	2.9	**2.7**
2h5nd	123	α	7.4	6.2	7.4	**5.3**
3gwl	106	β	5.8	5.4	2.9	**2.7**



(a) (b)

Figure 5.11: Performance profiles for the algorithms on (a) lowest energy and (b) lowest lRMSD metrics on the benchmark dataset.

$pr = 1.2$ and $pr = 1.3$, HEA-Map succeeds for 80% and 90% targets respectively. HEA-Map and SP-EA$^+$ reaches a success of 100% at a $pr = 1.7$, while HEA do so at $pr = 2.2$. Rosetta saturates at $pr = 2.2$ with a success for 90% targets.

These results show the utility of guidance by the map for conformation ensemble generation. The superior performance of HEA-Map suggests the algorithm is able to sample from the parts of the conformation space missed by the algorithms that does not use the map to enhance exploration. The quality of the conformations obtained by HEA-Map is shown qualitatively in Fig. 5.13, which draws the lowest-lRMSD conformation obtained by HEA-Map (drawn in blue) in three selected targets, superimposing it over the known native (drawn in olive). Rendering is performed with the CCP4mg molecular graphics software [1].

Evaluation on CASP Dataset

Table 5.9 shows the lowest *score*4 energy reached by each of the algorithms under comparison on the CASP dataset. Table 5.9 shows that HEA-Map achieves lower energy than all other algorithms in 7/10 cases. In a head-to-head comparison, HEA-Map beats all other algorithms easily and achieves lower energy than Rosetta in 9/10 cases, than HEA in all cases, and than SP-EA$^+$ in 8/10 cases. Table 5.11(c) evaluates the 1-sided statistical significance tests of the performance of HEA-Map over the other algorithms. Table 5.11(c) shows that the performance improvements are statistically significant at 95% confidence level (p-values < 0.05) for both Fisher's and Barnard's tests.

Table 5.10 shows the lowest lRMSD to the native conformation reached by each of the algorithms under comparison on the benchmark dataset. Table 5.10 shows that HEA-Map achieves lowest lRMSD in 9/10 cases. In a head-to-head comparison, HEA-Map beats all other algorithms comfortably and achieves lower lRMSD than Rosetta in 9/10 cases, than HEA in all cases, and than SP-EA$^+$ in 8/10 cases. Table 5.11(d) evaluates the 1-sided statistical significance tests of the performance of HEA-Map over the other algorithms. Table 5.11(d) shows that the performance improvements are statistically significant at 95% confidence level (p-values < 0.05) for both Fisher's and Barnard's tests.

Figure 5.12(a) shows the performance profiles of each algorithm over the CASP dataset in terms of the lowest energy reached. Figure 5.12(a) shows that the probability of HEA-Map to be the optimal algorithm among these 4 algorithms is about 0.70, considerably

Table 5.9: Comparison of the lowest energy obtained by each algorithm under comparison on each of the 10 CASP targets is shown in Columns 3-6. The CASP ID of the native and the sequence length of each target are shown in Columns 1-2. The lowest energy value reached per target is marked in bold.

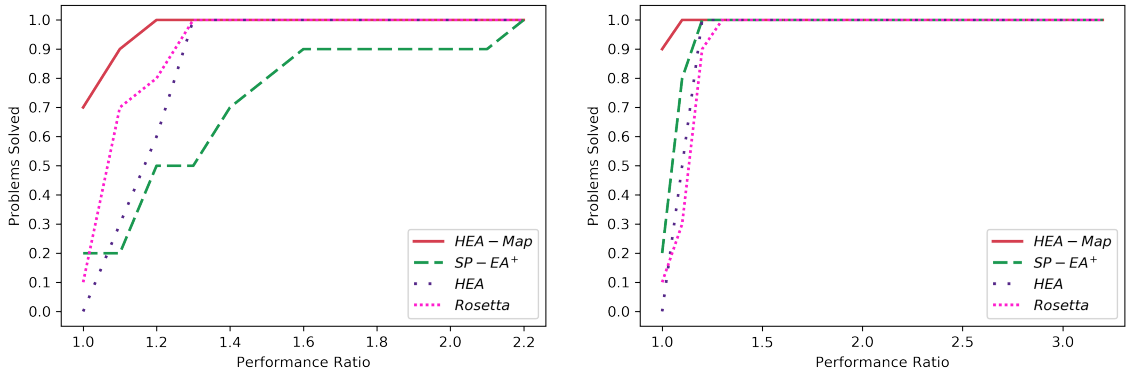
		Lowest Energy (REUs)			
Domain	Length	Rosetta	HEA	SP-EA$^+$	HEA-Map
T0859-D1	129	−99.5	−88	−92.4	**−103**
T0886-D1	69	**−89.2**	−69.9	−41.4	−83
T0892-D2	110	−101.8	−116.3	−76.7	**−120.8**
T0897-D1	138	−141.4	−135.2	−138.8	**−152.9**
T0898-D2	55	−65.5	−65.7	−51	**−70.1**
T0953s1-D1	67	−51.8	−55.8	**−67**	−60.7
T0953s2-D3	93	−53.1	−62.2	−44.5	**−66.3**
T0957s1-D1	108	−121.5	−102.6	−111.2	**−124.3**
T0960-D2	84	−79.7	−67.6	−63.2	**−87.5**
T1008-D1	77	−164.2	−148.4	**−170.9**	−167

Table 5.10: Comparison of the lowest lRMSD to the native conformation obtained by each algorithm under comparison on each of the 10 CASP targets is shown in Columns 3-6. The CASP ID of the native and the sequence length of each target are shown in Columns 1-2. The lowest lRMSD value reached per target is marked in bold.

		Lowest lRMSD (Å)			
Domain	Length	Rosetta	HEA	SP-EA$^+$	HEA-Map
T0859-D1	129	10.6	9.6	9.2	**9.1**
T0886-D1	69	6.3	6.4	6.2	**5.8**
T0892-D2	110	8	7.2	**6.7**	6.8
T0897-D1	138	9	9.3	8.4	**8.1**
T0898-D2	55	6.5	6.1	**5.8**	**5.8**
T0953s1-D1	67	7	6.2	5.7	**5.6**
T0953s2-D3	93	8.7	8	8	**7.6**
T0957s1-D1	108	6.9	7.4	7.2	**6.2**
T0960-D2	84	**7.2**	7.6	7.3	**7.2**
T1008-D1	77	3.2	3.6	3.6	**3**

more than any of the other algorithms. At $pr = 1.1$, HEA-Map succeeds for 90% targets. HEA-Map and SP-EA$^+$ reaches a success of 100% at a $pr = 1.2$, while HEA and Rosetta

do so at $pr = 1.3$. The performance profile of SP-EA$^+$ rises very slowly and reaches 100% at $pr = 2.2$. Figure 5.12(b) relates a similar analysis focusing on the lowest lRMSD to the native conformation and shows that the probability of HEA-Map to be the optimal algorithm among these 4 algorithms is about 0.90, considerably more than any of the other algorithms. HEA-Map reaches a success of 100% at a $pr = 1.1$, while SP-EA$^+$ and HEA do so at $pr = 1.2$. Rosetta reaches 100% at $pr = 1.3$.



(a) (b)

Figure 5.12: Performance profiles for the algorithms on (a) lowest energy and (b) lowest lRMSD metrics on the CASP dataset.

These results agree with the results in the benchmark dataset and emphasizes the effectiveness of guidance by the map to achieve more exploration of the energy landscape. The superior ability of HEA-Map to sample lower energy regions in the landscape also translates into better quality conformations closer to the native conformations.

5.3.4 Summary

In this section, we present an EA that is guided by a concise evolving map of the already explored regions of the conformation space. The EA is able to sample from unexplored regions of the conformation space through periodically excluding sampled individuals during selection and generating reasonably different new individuals. The results presented in the previous subsection demonstrates the effectiveness of the proposed EA for sampling better

Table 5.11: Comparison of HEA-Map to other algorithms via 1-sided Fisher's and Barnard's tests. The tests evaluate the null hypothesis that HEA-Map does not achieve (a) lower lowest energy on benchmark dataset, (b) lower lowest lRMSD on benchmark dataset, (c) lower lowest energy on CASP dataset, (d) lower lowest lRMSD on CASP dataset, considering each of the other algorithms in turn. P-values less than 0.05 are marked in bold.

(a)

Test	Rosetta	HEA	SP-EA$^+$
Fisher's	**0.0005467**	**0.01151**	**0.0005467**
Barnard's	**0.0002012**	**0.005909**	**0.0002012**

(b)

Test	Rosetta	HEA	SP-EA$^+$
Fisher's	0.08945	**5.95e-05**	0.08945
Barnard's	0.05789	**2.00e-05**	0.05789

(c)

Test	Rosetta	HEA	SP-EA$^+$
Fisher's	**0.0005467**	**5.41e-06**	**0.01151**
Barnard's	**0.0002012**	**9.54e-07**	**0.005909**

(d)

Test	Rosetta	HEA	SP-EA$^+$
Fisher's	**5.95e-05**	**5.41e-06**	**0.002739**
Barnard's	**2.00e-05**	**9.54e-07**	**0.001288**

1ail (lRMSD = 1.4Å) 1dtja (lRMSD = 2.8Å) 3gwl (lRMSD = 2.7Å)

Figure 5.13: The conformation obtained by HEA-Map that is closest to the native conformation is shown for three selected cases, the protein with known native conformation under PDB ID 1ail (left), 1dtja (middle), and 3gwl (right). The HEA-Map conformation is in blue, and the known native conformation is in olive.

quality conformations than the prominent conformation ensemble generation algorithms and shows the potential of such mechanisms to enhance exploration of the protein conformation space while remembering a reasonably small number of conformations in its generated ensemble.

Chapter 6: Balancing Exploration and Exploitation

As described in Chapter 1, for an optimization algorithm seeking multiple near-native conformations in a vast, high-dimensional, and multimodal landscape, a proper balance between exploration and exploitation is critical. In this chapter, we work on balancing exploration and exploitation for conformation ensemble generation algorithms to improve conformational sampling. We first focus on mapping the multimodal energy landscape by retaining diversity of the conformations through a subpopulation scheme in Section 6.1. We then employ an adaptive mechanism to control this balance in Section 6.2. The algorithms we present here take the amino-acid sequence of a protein as input and provide an ensemble of conformations generated through the evolutionary process as output.

6.1 Using Subpopulation EAs to Map Protein Energy Landscapes

The work presented in this section has been published in [101]. To sample diverse minima that correspond to different biologically-active conformations, the classic optimization that aims to find the global optimum is insufficient and mapping of the conformation space is necessary. The need to map landscapes as key to understanding a wide range of molecular phenomena has long been recognized across computational physics, organic and inorganic chemistry, and biology [20, 43, 111–116]. For instance, mapping the energy landscape of a cluster of 38 Lennard-Jones atomic particles reveals a double funnel that provides a microscopic basis for understanding how relaxation to the global minimum is diverted into a set of competing structures [112]. In [111], the mapped energy landscapes of small clusters of atoms are revealed to be highly heterogeneous and contain low-energy minima with large basins of attraction. In [113], the energy landscape is shown to facilitate the analysis and

interpretation of supercooling and glass-formation phenomena. In [43, 114], various studies in computational chemistry, physics, and biology are summarized to propose and support the holistic view of the energy landscape as central to explaining the behavior of atomic clusters, glasses, and even proteins.

While great progress has been made in mapping energy landscapes of atomic clusters [116], glasses [115], and short peptides [20], mapping protein energy landscapes remains challenging due to the complexity of such landscapes. In glasses, atomic particles, and short peptides, the number of interacting atoms/particles is small, and EAs that rely mainly on exploitation and limit exploration to naive strategies (e.g., random restart) can be useful. However, such approaches lose efficacy rapidly on landscapes of increasing modality, and sophisticated strategies are needed to balance between exploitation and exploration to avoid premature convergence.

Building on the pioneering efforts of Holland, De Jong, Goldberg, and Richardson [117, 118], various strategies have been proposed to address adequate exploration and diversity maintenance. Biased towards strategies that have been shown effective on computational structural biology problems, we highlight here three main techniques often used in combination: a hall of fame mechanism, multi-objective optimization, and hybridization. Work in [102] integrates a hall of fame mechanism in a hybridized/memetic EA to encode a detailed representation of the EA-explored landscape. Work in [41, 46] links the presence of multiple minima in protein energy landscapes to competing objectives in energy functions and demonstrates the utility of multi-objective optimization EAs. Work in [11, 12, 119] additionally debuts decentralized selection operators to retain diversity. Work in [14, 40] pursues various recombination strategies to promote generation of diverse candidates, hybridization for better exploitation, and non-local optimization operators to balance between exploration and exploitation.

Here, we develop subpopulation-oriented EAs as vehicles to do so. In a subpopulation EA, the population is divided into multiple subpopulations and each subpopulation seeks solutions in the subspace around the niche (basin of attraction) it occupies. The concept of

a subpopulation is appealing, as it can be directly linked to a conformational state. Thus, an EA that evolves and maintains multiple subpopulations at local minima while exploring new regions of the fitness landscape seems ideally suited for identifying multiple conformational states. To do so, subpopulation EAs must maintain diversity, a recurring theme in Evolutionary Computation (EC) research. While EC literature on subpopulation models is quite extensive, subpopulation EAs have not yet been considered for molecular modeling. Largely, existing research considers two scenarios, one where there is prior information on landscape modalities, and one where there is no such information. For the case of prior information, we highlight seminal work by Goldberg and Richardson [118], which assumes that the number of modalities/optima and their location are both known. This setting is not valid in molecular modeling, where the objective is to actually discover the diverse optima. An early survey by Spears [120] summarizes the use of restricted mating schemes to evolve subpopulations when no information about the optima is available. Work in [121] proposes a set of multi-population genetic algorithm (GA) operators for general landscape mapping. More recent work in [122] applies subpopulation EAs to the problem of feature selection but utilizes known information to organize the initial population into subpopulations (also referred to as tribes in [122]).

In this section, we presume no *a priori* information regarding the number and/or location of optima, or the distinct characteristics that may allow organizing individuals in the initial population into distinct subpopulations. We note that in a discovery setting, the location of the competitive states would not be known in molecular modeling, though occasionally in computational physics or chemistry applications information would be available regarding the number of such states and attributes distinguishing them. In de novo conformation ensemble generation, such information is not available, and one must proceed in more difficult blind settings.

First, we develop a subpopulation EA, to which we refer as SP-EA$^-$, by building on earlier work on effective representations of protein conformations and representation-aware

variation operators (described in Chapters 2 and 3). SP-EA$^-$ organizes the initial population into subpopulations and then applies subpopulation competition to provide more resources to the fitter subpopulations during the evolutionary process. The diversity introduced through different subpopulations helps the exploration component of the search. The exploitation comes from the evolutionary process within a subpopulation and the competition for resources between the subpopulations where more resources are allocated to fitter subpopulations. We then further extend this baseline EA so as not only to allocate more computational resources to fitter subpopulations, but additionally maintain stable and diverse subpopulations via a niche preservation technique. We refer to this algorithm as SP-EA$^+$.

While our primary motivation is identifying the modality of unknown protein energy landscapes, we first evaluate the two EAs on benchmark problems with fitness landscapes of known modalities. We then provide a comparative evaluation in the conformation ensemble generation setting and additionally compare the two EAs against Rosetta conformation sampling method. The results demonstrate that the subpopulation mechanism offers several advantages over the state-of-the-art, with the niche preservation technique yielding the best performance.

6.1.1 SP-EA$^-$: A Baseline Subpopulation EA

SP-EA$^-$, shown in pseudocode in Algorithm 1, first initializes a running counter that keeps track of fitness function evaluations (line 1) so as to evaluate and compare the two different algorithms (SP-EA$^-$ and SP-EA$^+$) using the same user-defined budget FMAX of fitness evaluations.

The initial population is obtained via an initialization mechanism (line 3). For the benchmark problems studied here, coordinates for individuals are drawn uniformly at random from the given parameter ranges. On applications to proteins, we employ the effective initialization mechanism used in HEA (described in Section 2.2.2).

Algo. 1 Baseline EA

Require: `FMAX` //total computational budget
`N` //population size
`CompFreq` //competition frequency
`ElitismRate` //elitism rate

1: $fcounter \leftarrow$ `FMAX` //counter of fitness evaluations
2: $i \leftarrow 0$ //generation counter
3: $\langle \mathcal{P}_i, budgetSpent \rangle \leftarrow$ InitOper(`N`) //generate initial population
4: $fcounter \leftarrow fcounter - budgetSpent$
5: $\{\mathcal{S}_1, \ldots, \mathcal{S}_K\} \leftarrow$ GenSubPops($\mathcal{P}_i$) //divide into subpopulations
6: **while** $fcounter > 0$ **do**
7: **for** $S \in \{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$ **do**
8: $C \leftarrow \emptyset$ //set of offspring
9: **for** $s \in S$ **do**
10: $c \leftarrow$ VarOper(s) //generate offspring
11: $\langle c', f', budgetSpent \rangle \leftarrow$ ImprovOper(c) //improve offspring
12: $fcounter \leftarrow fcounter - budgetSpent$
13: $C \leftarrow C \cup \{c', f'\}$ //add improved offspring
14: $S' \leftarrow$ SelOper(S, $\mathcal{C}$, `ElitismRate`) //select
15: $\mathcal{S} \leftarrow \mathcal{S}'$ //update subpopulation
16: **if** $i \bmod$ `CompFreq` $= 0$ **then**
17: $\{\mathcal{S}_1, \ldots, \mathcal{S}_K\} \leftarrow$ SubPopCompete($\{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$)
18: $i \leftarrow i + 1$

Defining Subpopulations

Unlike a classic EA, where, once initialized, the population evolves over generations, the subpopulation EAs we present here first organize the initial population into subpopulations. Unlike other work, where information may be available on the attributes that can be leveraged for such organization, here we assume no *a priori* information. That is why line 5 in Algorithm 1 simply refers to a mechanism to generate subpopulations from the initial population. In this work, we employ leader clustering, but other clustering algorithms can be utilized. The main idea behind leader clustering is that individuals are considered in order, and each individual either forms a new cluster (becoming its representative) or is assigned to the first cluster whose representative is within a distance threshold.

For the benchmark problems considered here, we utilize Euclidean distance to measure the distance between an individual yet to be assigned to a cluster and the representative individual of each cluster computed so far. In applications (and adaptation) of SP-EA$^-$ and SP-EA$^-$ to proteins, the distance function used is lRMSD as described in Section 3.4.

We note that the number of subpopulations in line 5 is not predetermined. Clustering algorithms that necessitate such determination can be used, but one of the reasons we prefer leader clustering is that the number of clusters follows based on the specified distance threshold. Once subpopulations are determined, they each undergo an evolutionary process. Lines 7-15 in Algorithm 1 evolve each subpopulation as follows. For each subpopulation, offspring are recorded in a set $\mathcal{C}$ that is initialized to the empty set (line 8). Each individual in the current subpopulation S under consideration (line 9) is selected to obtain an offspring c via a variation operator (line 10). The variation operator for the benchmark problems studied here is a Gaussian perturbation operator, which perturbs each coordinate of an individual by a value drawn from a zero-mean Gaussian distribution with a given variance. In applications on proteins, the variation operator is implemented as in Section 2.2.2.

Evolving Each Subpopulation

The obtained offspring is then subjected to a local search that seeks to improve the offspring (line 11). For the benchmark problems considered here, a naive local search chooses any of the coordinates of the offspring with equal probability and then applies a simple gradient descent on the chosen coordinate for a total of `budgetSpent` iterations/cycles. The local search utilized in the applications on proteins is implemented as described in Section 2.2.2. Note that all fitness evaluations that occur in the improvement operator are counted, and they are removed from the total budget (line 12).

Once the offspring of a subpopulation are generated and stored in D (line 13), they compete for survival with parents (line 14). An elitism-based truncation selection mechanism is employed for this purpose as described in Section 2.2.2.

Competition Among Subpopulations

Once this process completes for each subpopulation (line 7), subpopulations may now compete with one another. How frequently this occurs is determined via a user-defined competition frequency (line 16). In this work, the competition takes place once every `CompFreq` generations. Algorithm 1 does not provide details of the competition process (line 17) which may update subpopulations. Different implementations of this process give rise to different variants of subpopulation EAs. Let us delay momentarily the implementations we consider here in the interest of first explaining how the competition among subpopulations takes places.

Provided a mechanism exists to associate a fitness with an entire subpopulation, a competition mechanism aims to accomplish the following. The fittest subpopulation is rewarded with more resources in hopes of affording better exploration of the landscape. This is operationalized by replicating the fittest individual in the fittest subpopulation; the size of the fittest subpopulation increases by 1. In addition, the worst (lowest fitness) subpopulation is penalized by discarding its worst (lowest fitness) individual; the size of the worst subpopulation decreases by 1. Note that it is possible under this mechanism

for a subpopulation to gradually lose all its members, resulting in the elimination of a subpopulation.

As Algorithm 1 shows, the process of subpopulation evolution and subpopulation competition is repeated until the fitness evaluation budget is exhausted. At that point, the algorithm terminates. The competitive mechanism described above is greatly dependent on how the fitness of a subpopulation is defined. SP-EA$^-$ considers a straightforward definition of the fitness of a subpopulation as the average over the fitness values of individuals in the subpopulation:

$$F_S = \frac{\sum_{s \in S} f(s)}{|S|} \tag{6.1}$$

6.1.2 SP-EA$^+$: A Niche-Preserving Subpopulation EA

The population competition utilized in SP-EA$^-$ may result in a loss of population diversity in cases in which a subpopulation with the highest fit individuals may persist indefinitely, gradually acquiring more members, resulting in the loss of subpopulations containing less fit individuals. To provide some subpopulation stability, SP-EA$^+$ preserves niches in a population by redefining the fitness of a subpopulation to consider not only the fitness values of its members but also the size of the subpopulation. Specifically,

$$F_S = \frac{\sum_{s \in S} f(s)}{|S|} + T \cdot |S| \tag{6.2}$$

In Equation 6.2, the fitness of a subpopulation not only calculates the average over the fitness values of the members of the subpopulation, but also penalizes the subpopulation fitness by a factor (governed by the "temperature" parameter T) of the subpopulation size (number of members). Larger subpopulations have more penalty added to their score. This ensures that a large subpopulation can only win, if it really holds much fitter individuals than smaller subpopulations. Otherwise, smaller subpopulations get to increase their sizes.

This way, a small subpopulation can also win if it holds good individuals, even if they are not the fittest. This helps preserve the niches, lets the algorithm map more of the subspaces in the search space, and gives the algorithm a better chance of finding diverse optima.

Note that the temperature parameter shifts the balance in the fitness of a subpopulation towards fitness or size. For instance, if $T = 0$, SP-EA$^+$ reverts to SP-EA$^-$ and does not consider the size of a population.

6.1.3 Results

Here, we present a summary of our results. We first apply both algorithms on two generic landscapes with known global minima to analyze their performance in finding these minima as well as in the overall exploration of the subspaces. We also examine the stability of the subpopulations that they generate. Then, we execute both algorithms in the context of conformation ensemble generation on the benchmark dataset and compare them via different metrics against each other and against the popular Rosetta algorithm as described in 2.2.1.

Analysis on Known Fitness Landscapes

We choose two benchmark problems to comparatively evaluate the behavior of SP-EA$^-$ and SP-EA$^+$:

- A sphere: $f(x) = \sqrt{\sum_{i=1}^{n} x_i^2}$.
- The product of two spheres: $f(x) = \sqrt{\sum_{i=1}^{n} (x_i - 200)^2} \times \sqrt{\sum_{i=1}^{n} (x_i + 200)^2}$.

where x is a D-dimensional vector. The landscape of the sphere function contains 1 global minimum, and the landscape of the product of two spheres contains 2 global minima. Each algorithm is run $1,000$ times on each problem. On each run, we randomly pick the dimensionality D from $\{2, 5, 10, 20\}$. We set the temperature for SP-EA$^+$ to 6, 12, 25, and 50, respectively, for $D = 2$, 5, 10, and 20. We fix the range of values for each x_i to

$[-500, 500]$. The population size is set to 200, elitism rate for selection to 25%, the frequency for subpopulation competition to 2, and the evaluation budget for each run to $10,000,000$ fitness evaluations (this same budget is used in our evaluation on protein landscapes in the context of conformation ensemble generation).

We first evaluate the number of times each algorithm converges to the known global minima (or minimum). We consider an algorithm to have converged if for 1-sphere problem, the final population generated by the algorithm consists of a single subpopulation and that subpopulation contains the global minimum; and for the 2-sphere problem, the final population generated by the algorithm consists of only two subpopulations and each of the subpopulations contains one global minimum each. Table 6.1 shows the percentage of times the two EAs converge in 1000 runs on each problem. Both EAs converge in the 1-sphere problem to the only minimum in all the runs. On the 2-sphere problem, SP-EA$^-$ does not converge to both minima in the final subpopulations. In most cases, SP-EA$^-$ converges to a single subpopulation. This result indicates the genetic drift that occurs along the way, with the population losing diversity early. SP-EA$^+$ performs well and retains both minima the majority of the time, indicating that the niche-preserving technique is effective in preventing premature convergence.

Table 6.1: Percentage of times (out of $1,000$ runs) SP-EA$^-$ and SP-EA$^+$ converge to the 1 minimum and 2 minima in the known landscapes of the sphere problems considered here.

Algorithm	1-sphere	2-spheres
SP-EA$^-$	100%	0.13%
SP-EA$^+$	100%	71.2%

We now provide a visual analysis of the stability of the subpopulations by examining the size of the subpopulations in the final population of SP-EA$^+$. Fig. 6.1 shows the histogram of the smaller subpopulation sizes in the final populations for the 2-sphere problem in the cases where SP-EA$^+$ converges. In 75.9% cases, the smaller subpopulation has a size

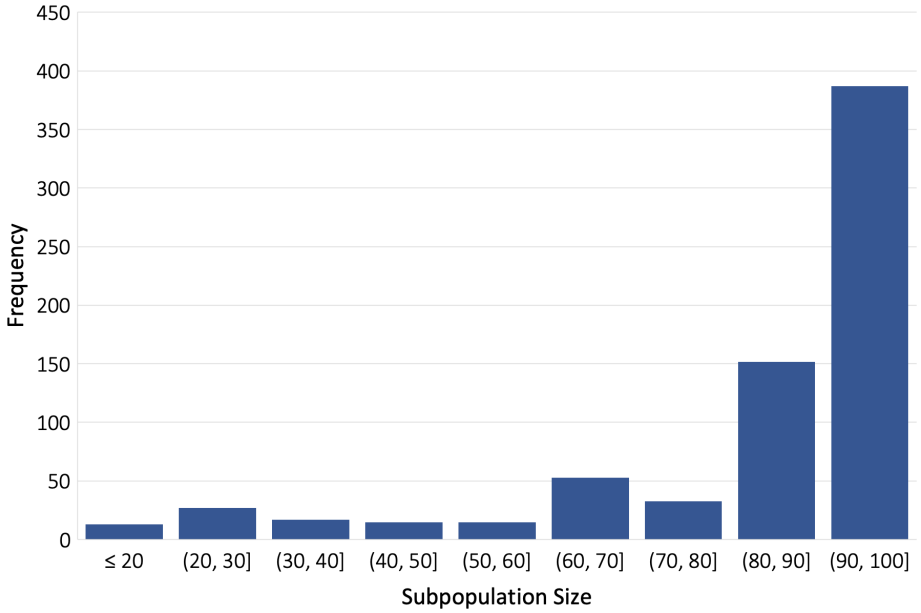


Figure 6.1: Histogram of smaller subpopulation sizes in the final population for the 2-sphere problem on the runs where SP-EA$^+$ produces 2 subpopulations that contain one minima each.

of 80 or more out of 200 individuals in the population. Only in 1.8% of the cases, the smaller subpopulation has a size of 20 or less. Considering the substantial budget, these results confirm that SP-EA$^+$ not only retains population diversity, but also produce stable subpopulations.

Analysis on Protein Landscapes

To analyze the performance of the two algorithms on the protein conformation space, we consider the 20 proteins in the benchmark dataset listed in Section 3.3. With regards to parameter values, differences from the above evaluation include the distance threshold, which is set to 5Å, and the temperature in SP-EA$^+$, which is set to 2, and the number of runs, which is set to 5 times on each protein sequence to account for stochasticity. We report the best performance over all 5 runs combined for each EA. Since the evaluation budget for

each run of each of our EAs is fixed to 10M evaluations, this adds up to 50M over 5 runs. We compare the two EAs to each other and the Rosetta conformation sampling algorithm. For a fair comparison, Rosetta is run for 54M energy evaluations on each target. Rosetta is evaluation expensive, and one run of it exhausts 36K score evaluations. The above total budget results in $1,500$ conformations over $1,500$ runs.

Table 6.2: Comparison of the lowest energy (in Rosetta Energy Units – REUs) obtained by each algorithm on each of the 20 test cases is shown in Columns 2, 3, and 4. Comparison of the lowest lRMSD (measured in Angstroms – Å) to the known native conformation for each test case is shown in Columns 5, 6, and 7.

	Lowest Energy			**Lowest lRMSD**		
PDB ID	Rosetta	SP-EA$^-$	SP-EA$^+$	Rosetta	SP-EA$^-$	SP-EA$^+$
1ail	−29.9	−74.1	**−81.3**	4.5	1.5	**1.2**
1aly	**−112.5**	−63.4	−74.6	12.4	11.1	**10.9**
1aoy	−73.3	−103.2	**−116.8**	4	**3.1**	**3.1**
1bq9	−46.9	−51.3	**−64.2**	**2.9**	4.3	4.7
1c8ca	**−101.4**	−69.5	−78.3	**2.2**	3.7	3.6
1cc5	**−82.5**	−67.6	−76.4	**3.7**	4.2	4.7
1dtdb	−66.5	−57.5	**−69.6**	**4.2**	5.2	5
1dtja	−72.5	**−85.5**	−72.6	**2.3**	**2.3**	2.5
1fwp	−71.3	−66.9	**−72.1**	**2.8**	4	3.7
1hhp	**−106.3**	−87.8	−83.5	10.1	8.4	**8.2**
1hz6a	−117.1	−122.5	**−122.8**	**1.9**	2.3	**1.9**
1isua	−27	−38.8	**−41.8**	6.6	6.1	**5.8**
1sap	−107.8	−91.2	**−109.9**	**2.8**	4.4	4
1tig	**−138.2**	−104.1	−112.2	**2.5**	3.5	3.7
1wapa	**−109**	−65.9	−71	6.5	5.9	**5.6**
2ci2	−37.8	−72.7	**−82.7**	5.8	3.6	**3.5**
2ezk	−51.1	−126.4	**−135.2**	3.6	3	**2.9**
2h5nd	−82.5	−134.9	**−139.1**	**7.4**	7.8	**7.4**
2hg6	−82.5	**−96.4**	−95.1	9.4	8.9	**8.7**
3gwl	−68.2	−112	**−117.8**	5.8	4.2	**2.9**

Table 6.2 summarizes the performance of each of the three algorithms in terms of the lowest reached Rosetta *score*4 energy and the lowest reached distance (lRMSD) to the known native conformation of the target under consideration; the lowest values on each

target are marked in bold. The first column lists the test cases by identifying the PDB IDs of the entry where an active conformation known for each test case is deposited.

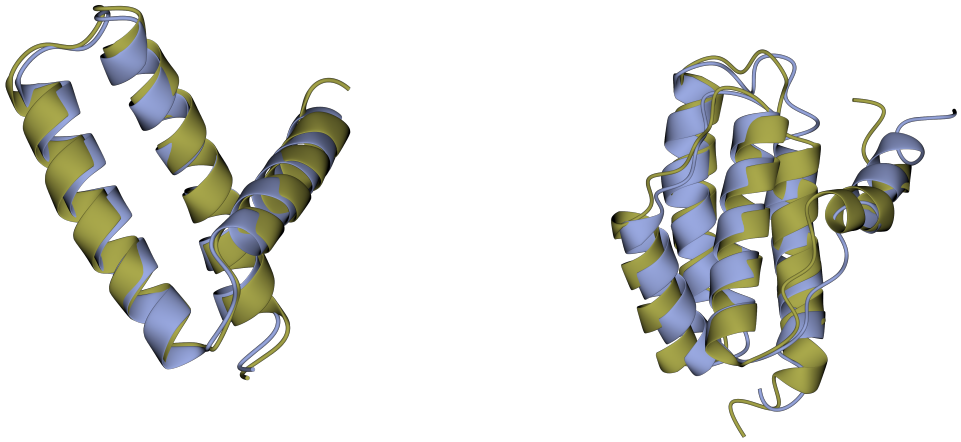
Table 6.2 shows that SP-EA$^+$ achieves the lowest energy in 12/20 targets, whereas SP-EA$^-$ and Rosetta do so on 2/20 and 6/20 targets, respectively. In a head-to-head comparison between SP-EA$^+$ and Rosetta, SP-EA$^+$ achieves lower energy in 14/20 targets over Rosetta. Between SP-EA$^+$ and SP-EA$^-$, the former wins in 17/20 cases. Finally, between SP-EA$^-$ and Rosetta, SP-EA$^-$ wins in 11/20 cases.

A similar comparison on lowest lRMSDs reveals that SP-EA$^+$ achieves the lowest lRMSD in 12/20 targets, whereas SP-EA$^-$ and Rosetta do so on 2/20 and 10/20 targets, respectively. In a head-to-head comparison between SP-EA$^+$ and Rosetta, Rosetta achieves lower lRMSD in 8/20 targets than SP-EA$^+$. Between SP-EA$^+$ and SP-EA$^-$, the former wins in 15/20 cases. Between SP-EA$^-$ and Rosetta, Rosetta wins in 9/20 cases.

To give some insight into these low lRMSD values, Fig. 6.2 selects two proteins (with respective active conformations under PDB IDs 1ail and 3gwl) and shows the lowest-lRMSD conformation obtained by SP-EA$^+$ in each case. These conformations (drawn in blue) are superimposed over the corresponding native conformations (drawn in olive). The superimposition highlights the quality of the solutions obtained by SP-EA$^+$.

The comparisons so far suggest that the subpopulation EAs outperform Rosetta on both metrics. We harden this result via statistical significance analysis tests. We use two statistical significance tests, Fisher's and Barnard's exact tests, to determine if the results are statistically significant. We employ the tests over 2x2 contingency matrices generated from the results obtained using the comparison metrics.

Table 6.3 shows the p-values for the 1-sided Fisher's and Barnard's tests for the lowest energy head-to-head comparison. All the values (< 0.05) that reject the null hypothesis with 95% confidence are marked in bold. Both null hypotheses (SP-EA$^+$ does *not* perform better than Rosetta and SP-EA$^+$ does *not* perform better than SP-EA$^-$) are rejected, confirming the superior performance of SP-EA$^+$. The null hypothesis that SP-EA$^-$ does *not* perform



1ail (1.2Å) 3gwl (2.9Å)

Figure 6.2: The lowest-lRMSD conformation obtained by SP-EA$^+$ on each protein is drawn in blue, superimposed over the corresponding known native conformation (with PDB id and lRMSD shown), which is drawn in olive. Rendering is performed with the CCP4mg molecular graphics software [1].

better than Rosetta is not rejected, indicating that the performance improvement of SP-EA$^-$ over Rosetta is not statistically significant with 95% confidence.

Similarly, Table 6.3 also shows the p-values for the 1-sided Fisher's and Barnard's tests for the lowest lRMSD head-to-head comparison. All the values (< 0.05) that reject the null hypothesis with 95% confidence are marked in bold. The null hypothesis that SP-EA$^+$ does *not* perform better than SP-EA$^-$ is rejected, confirming the superior performance of SP-EA$^+$ over SP-EA$^-$. The null hypotheses that SP-EA$^+$ does *not* perform better than Rosetta and that SP-EA$^-$ does not perform better than Rosetta are not rejected, indicating that the performance improvements of the two subpopulation EAs over Rosetta are not statistically significant with 95% confidence.

Taken altogether, the results presented above suggest a stronger conformation sampling capability of the subpopulation EAs over Rosetta and a superiority of the niche-preservation

Table 6.3: p-values obtained by 1-sided Fisher's and Barnard's tests for head-to-head comparison of the algorithms on lowest energy (left) and lowest lRMSD (right). Top panel evaluates the null hypothesis that SP-EA$^+$ does *not* perform better than Rosetta. Middle panel evaluates the null hypothesis that SP-EA$^+$ does *not* perform better than SP-EA$^-$. Bottom panel evaluates the null hypothesis that SP-EA$^-$ does *not* perform better than Rosetta.

SP-EA$^+$ vs. Rosetta

Test	Lowest energy	Lowest lRMSD
Fisher's	**0.01282**	0.3756
Barnard's	**0.008299**	0.3057

SP-EA$^+$ vs. SP-EA$^-$

Test	Lowest energy	Lowest lRMSD
Fisher's	**0.000009693**	**0.0006159**
Barnard's	**0.000004182**	**0.0003401**

SP-EA$^-$ vs. Rosetta

Test	Lowest energy	Lowest lRMSD
Fisher's	0.3762	0.5
Barnard's	0.3179	0.4373

technique in balancing exploration and exploitation. On the lRMSD-based comparison, none of the algorithms is a clear winner, but the subpopulation EAs perform comparably to Rosetta.

6.1.4 Summary

In this work, we employ subpopulation-oriented EAs to map the energy landscapes and attain a balance between exploration and exploitation of the landscape for conformation ensemble generation. Mapping is only relevant when the problem of interest is characterized by a multimodal landscape where the various modes contain information about the system being investigated. This is the case for most biological systems, and, in particular, protein molecules. Since neither the number of subpopulations nor their distribution are known ahead of time for unknown molecular landscapes, we present here a baseline subpopulation EA that makes use of phenotypic clustering to define initial subpopulations and makes use of

subpopulation competition to evolve subpopulations. We investigate two different strategies for such competition and show that taking into account not only the height/depth, but also the size/breadth of a local optimum allows better retaining diverse subpopulations that converge to the different modes of known landscapes. Evaluation on unknown landscapes in the context of conformation ensemble generation shows that niche preservation also confers better exploration-exploitation balance.

6.2 Adaptive Stochastic Optimization to Improve Protein Conformation Ensemble Generation

A proof-of-concept version of the work presented in this section has been published in [123] and the extended version is under review in [124]. The work presented in Section 6.1 attempts to balance the exploration and exploitation utilizing subpopulation schemes which yield better outcome in terms of sampling lower energy conformations but not so in terms of reaching closer towards the native conformations. This suggest a better balance between exploration and exploitation is necessary to sample from the near-native regions in the landscape. Therefore, in this section, we explore an adaptive setting.

The evolutionary computation setting exposes algorithmic knobs/parameters such as representation, population size, reproduction process, and selection process that can be varied to control the inherent trade-off between exploration and exploitation. Experimentally tuning these parameters to static/fixed values before the run of the algorithm and keeping the same values during the run to achieve a good exploration-exploitation balance is an extremely difficult task as different configurations of the parameters could be better suited at different points of the optimization process [125–129]. Therefore, lots of researches have been focused on adaptively changing the values of the parameters during the run of an EA.

In the EC literature, changing parameters on the fly has been approached in three ways; uninformed, self-adaptive, and adaptive parameter control. In an uninformed parameter

control setting [130], the value of a parameter is changed according to a schedule (a function of time elapsed or number of generation passed) set before the run of the algorithm, without considering feedback about the current state of the search [131, 132]. This approach to setting a schedule is hard as it requires predicting beforehand the number of generations for which the algorithm is likely to run before it converges.

In self-adaptive parameter control setting, the parameters to be adapted are subjected to the same evolutionary process as the search for optimal solutions. The parameters are usually encoded into the chromosome of the individuals and are evolved together [133–137]. The idea is that better values of the parameters will result in fitter individuals and these individuals will be more likely to survive for the next generations which will also pass on the better values of the parameters presumably responsible for the better fitness. However, evolving parameters in this way increases the dimensionality and complexity of the problem as the search space is extended to also include the algorithmic parameters. Moreover, the parameter values are also susceptible to premature convergence [129, 138].

On the other hand, adaptive parameter control setting explicitly adapts the parameters in an informed way by taking feedback from the optimization process and using them to determine the direction and/or the amount of change to the parameters. This way of parameter control tracks specific properties (for example, the fitness of the individuals) of an EA run and the update mechanism for the parameters are guided by the changes in the tracked properties [139–143]. Adaptive parameter control is deemed to be the more effective way of adapting parameters and most of the research activities in relevant literature are focused on it [144, 145]. The parameters that are generally adapted are variation operator, population size, representation, and selection operator. Early example of an adaptive variation operator is Rechenberg's 1/5 rule for mutation [139] which adapts the mutation step size. The rule suggests an increase in the mutation step size if the ratio of successful mutations to all mutations is greater than 1/5 and vice versa. Work in [146] adapts the mutation probability, work in [147] adapts the crossover probability, and works in [148, 149] adapt the probability of both mutation and crossover based on their success. Work in [150] monitors

the performance and recent contribution of crossover and mutation operators, and adapts the ratio of crossover to mutation accordingly. Attempts that adapts the parent population size include adjusting the population size based on the selection error probability [151], persistence of individuals [152], and fitness improvements [153]. Other notable attempts adapt the sizes of the subpopulations [141, 154] within the population or the size of the offspring population [155–157]. Researches that focus on adapting the representation typically do so for genotypic representation. Work in [158] adapts the number of bits used to represent a gene, the range of the values of the function variables that the genes are mapped to, and the center of the range. Work in [159] is based on multiple restarts of the algorithm where the encoded genes represent the distances between the current solutions and the solutions from the previous run (delta values), and the resolution of the delta values are adapted. Another notable work that adapts the representation [160] does so by adapting the position of the genes. Most of the adaptive techniques that adapt the selection operator do so via the tournament size parameter in tournament selection, mostly because some of the selection operators like the fitness proportional selection are inherently adaptive (fitness proportional selection exerts more selection pressure in the early generations than in the latter generations) and it is easier to increase/decrease the selection pressure by increasing/decreasing the tournament size. Notable attempts to adapt this parameter include works in [161–163]. An interested reader is referred to [144, 164–167] for comprehensive reviews on adaptive parameter control.

Here, we choose to focus on adaptive parameter control approach. The choice reflects the shortcomings of uninformed and self-adaptive settings stated above. Choosing what to adapt requires some careful considerations. There are challenges to adapting the representation and the variation operator for de novo conformation ensemble generation as it requires giving up the clever and domain-specific fragment assembly process, described in Section 2.1, which has been credited for the main leap in the performance of conformation sampling algorithms. The fragment libraries are based on phenotypic representations and any changes to the representation would be hard to accommodate for an EA that uses

fragment assembly. This is another reason for not choosing the self-adaptive setting as it requires encoding the EA parameter to adapt into the conformation parameter representation. The usefulness of the fragment replacement technique as a variation operator also makes adapting the parameters in the variation operator very difficult. Adapting mutation step size is not feasible for there is no fixed step size for fragment replacement as we choose fragments to replace from a finite set of fragments from the fragment library. Moreover, if we need to increase/decrease the step size (i.e. amount of change in the dihedral angles), there may not be any fragment stored in the fragment library which differs by that much. Adapting mutation rate is also not feasible as the fragment libraries offer a very few fragment length options. The fragment library we use contains fragments of length 3 and 9. Also, fragments are only considered "good" if the individual configurations are inserted together. The utility of the fragment replacement is also the reason crossover operators are generally not used for conformational sampling. On the other hand, adapting the population size and the selection operator requires no change in the fragment assembly process. However, finding an effective strategy to adapt the population size has proved to be rather difficult and as a result, most EAs in practice keep the population size fixed [165]. Therefore, here we choose to adapt the selection procedure based on the feedback received during the EA run.

In EAs, most of the exploitation comes from the applied selection mechanism. The greediness (the inclination to select fitter individuals) of the selection mechanism is directly related to the exploration/exploitation pressure exerted by the EA. A more greedy selection method applies stronger selection pressure than a less greedy selection method and results in more exploitation and less exploration of the search space, and vice versa. How to adjust this selection pressure during an EA run is nontrivial and requires some careful thinking. Adapting the tournament size parameter in tournament selection is appealing but literature has shown that tuning this parameter in standard tournament selection to control exploration might lead to premature convergence [168]. Moreover, adjusting the tournament size to control selection pressure has its limits. At a lower bound (binary tournament),

tournament selection applies much stronger selection pressure than a weak selection scheme such as a uniform selection scheme [39]. But, weak selection pressure exerted by schemes such as uniform selection could be useful to prevent premature convergence and stagnation of the population. Therefore, to apply a wider range of selection pressure, we propose to switch between different selection schemes with different selection pressure to control the exploration-exploitation balance during the the run of the EA which is a novel approach. We believe this approach has the potential to improve the performance of the conformation ensemble generation algorithms while this approach may also generate some interest in the EC community.

What to measure or what feedback from the EA run to use to adapt the EA parameters primarily differs in two ways in the literature; fitness and diversity of the individuals. Approaches that account for fitness include adapting parameter based on periodic checks for change in best fitness of the individuals [153], based on the difference between the best fitness and the average fitness of the population [169], based on the fitness ranking of each individual [170], based on best-fitness frequency [171], and based on the success/fitness gain of the parameter values [139, 148]. Approaches that account for diversity include adapting parameter based on the Euclidean distances between the individuals and the best solution found so far [171], based on Euclidean distance between individual solutions [161], based on the Hamming distance of the individuals [172], and based on diversity in the fitness of the individuals measured using the best, worst, and average fitness of the individuals in the population [173]. An interested reader is referred to [144,167] for more information about the feedback mechanisms used in literature. Here, we choose to take periodic change in best fitness as the evidence upon which the adjustments are performed while other measures could have also been explored.

To investigate the effects of changing selection pressure to obtain different exploration and exploitation capability in EAs, we build upon HEA. For the selection mechanism, HEA uses truncation selection which is well-known to provide strong selection pressure that results in more exploitation and less exploration. From now, we refer to this baseline HEA

algorithm as HEA-TR (TR for truncation). We design three variants of the HEA-TR algorithm. These variants only change the selection mechanism, utilizing the other operators (initial population, variation, and improvement) as in HEA-TR. The selection schemes that we propose to use in these EAs are *uniform stochastic*, *fitness proportional*, and *quaternary tournament*. The reason for choosing these schemes is as follows. Uniform stochastic selection applies the weakest selection pressure. Truncation selection falls in the other end of the spectrum, exerting the strongest selection pressure. Fitness proportional selection applies stronger selection pressure than uniform selection. Although it provides less selection pressure than binary tournament selection, its exerted pressure is higher when there is more diversity in the population [39]. Finally, the selection pressure exerted by quaternary tournament selection falls in between fitness proportional and truncation selection.

We first describe the three variants, HEA-QT, HEA-FP, and HEA-US, depending on the selection mechanism employed, as we detail below. Finally, we describe HEA-AD, which implements the adaptive selection mechanism.

6.2.1 HEA-QT

In HEA-QT, the initial population, variation, and improvement operators are kept unchanged but the selection operator uses the quaternary tournament selection scheme instead of the truncation selection of HEA-TR. The idea is to reduce the selection pressure to decrease exploitation and promote exploration. In HEA-QT, all the parents and the improved offspring are first combined to form a selection pool S and each individual in S is evaluated using $score4$. Next, a 4-way tournament is held for each of the n spots in the population for the next generation, where n is size of the population. A uniform probability distribution is used to randomly pick 4 individuals from S with replacement and these 4 individuals then compete with each other to survive for the next generation. The fittest individual according to $score4$ wins the competition and is selected to fill the next open spot in the population for the next generation.

6.2.2 HEA-FP

HEA-FP employs the fitness proportional selection scheme instead of the truncation scheme of HEA-TR, while all other operators remain the same. Fitness proportional selection employs lesser selection pressure than quaternary tournament and truncation. In HEA-FP, all the parents and the improved offspring are combined to form a selection pool S. Then, each individual in S is assigned a selection probability proportional to their fitness. Specifically, an individual $x \in S$ is assigned a selection probability of $f(x)/\sum_{i \in S} f(i)$, where $f()$ measures the fitness of the individual according to *score*4. This distribution is then sampled n times to pick n individuals for the next generation (n is population size).

6.2.3 HEA-US

HEA-US applies the weakest selection pressure through uniform stochastic selection. As in HEA-QT and HEA-FP, all the other operators remain unchanged. A selection pool S of size $2n$ (n is the population size) is first formed which contains all the parents and the improved offspring. HEA-US assumes identical fitness for all the individuals; n individuals are picked from S uniformly at random to form the population for the next generation.

6.2.4 HEA-AD: An Adaptive Algorithm

Here, we propose an adaptive algorithm, HEA-AD, that employs an adaptive selection operator to find better balance between exploration and exploitation. Instead of keeping the same selection pressure over generations, HEA-AD adapts the selection pressure based on the characteristics of the population. The algorithm evaluates the population every few generations for possible adjustments in the selection pressure and decrease or increase the selection pressure as needed.

Specifically, the adaptive mechanism periodically checks for a possible change of the selection pressure. The algorithm tracks the *best-so-far fitness*, which measures the best fitness (lowest Rosetta *score*4) over the g populations over the past g generations. Let us refer to this statistic as BSFF. The reasons for choosing BSFF metric are following.

BSFF is simple and computationally efficient to define and keep track of. In addition, a slowly improving BSFF suggests the selection pressure is too weak and a strong selection pressure typically results in rapid improvements in BSFF with a high risk of premature convergence [39]. Moreover, a lack of change in BSFF for several generations could be an indicator that the population is stuck exploiting some parts of the landscape and in this case a weaker selection pressure can be useful to achieve more exploration of the search space.

When a change needs to be made, as detailed below, the algorithm chooses a new selection mechanism from a scheme pool $SP = \{$uniform stochastic, fitness proportional, quaternary tournament, truncation$\}$. The pool is sorted in ascending order of selection pressure. HEA-AD first starts with a weaker selection scheme, the fitness proportional one, so as to encourage more exploration in the early generations. Over every g generations, the choice of the selection scheme is revisited as follows.

If the current BSFF (over the last g generations) increases by a small amount of $< s\%$ over the BSFF observed over g generations earlier, which suggests the selection pressure is too weak, the selection pressure is increased by replacing the current selection scheme with the next one in the pool SP that applies more selection pressure. Recall that the selection schemes are ordered from weakest to strongest. For example, if the current selection mechanism in HEA-AD is the fitness proportional one, HEA-AD will then set the current selection mechanism to be quaternary tournament.

If the current BSFF (over the last g generations) increases by a considerable amount of $> t\%$ over the BSFF observed over g generations earlier, we take this as an indication that too much exploitation is happening. The algorithm can converge prematurely to a suboptimal minimum. Therefore, the selection pressure is decreased by switching the current selection scheme with the previous scheme in SP. For example, if the algorithm is using truncation selection at this point, it will go on to use quaternary tournament from now on.

If the current BSFF (over the last g generations) is unchanged from the BSFF observed

over g generations earlier, and the algorithm is currently using truncation selection, the population could be stagnated and more exploration can help. Therefore, the selection pressure is decreased gradually by choosing the previous scheme in SP until the BSFF improves.

If the current BSFF (over the last g generations) is unchanged from the BSFF observed over g generations earlier, and the algorithm is currently using uniform stochastic selection, this indicates that the selection pressure has kept decreasing from truncation to end up in uniform stochastic. So, some exploration has already been performed by selecting weaker individuals and allowing them to reproduce. Therefore, we can now aim to improve the BSFF. The selection pressure is increased gradually for more exploitation by choosing the next scheme in SP until the BSFF improves.

This adaptive selection operator is utilized in HEA-AD algorithm to select individuals for the next generation. As with the other variants, the initial population, variation, and improvement operators remain unchanged.

6.2.5 Implementation Details

In all the EAs described above, the population size is $n = 100$ and the elitism rate for elitism-based truncation selection is 25%. As is commonly done for conformation sampling (and EAs more generally), the termination criterion is set to the exhaustion of a fixed budget of fitness/energy evaluations. Specifically, the algorithms presented above are executed for a fixed budget of $10,000,000$ energy evaluations. This results in typically $120 - 300$K conformations sampled over $700 - 1600$ generations. For HEA-AD, the checking parameter g is set to 15; the change parameter s is set to 5, and t is set to 15. We note that no specific effort has been made to fine-tune these parameters to the problem at hand. All algorithms are implemented in Python and interface with the PyRosetta library. Each algorithm takes $1-3$ hours on one Intel Xeon E5-2670 CPU with 2.6GHz base processing speed and 20GB of RAM. The runtime range is mainly due to the different lengths of the amino-acid sequences of the target proteins. As we describe further in Section 6.2.6, the algorithms are run 5

times on each target protein's amino-acid sequence to account for possible variance.

6.2.6 Results

Our evaluation is organized along two major sets of experiments. In the first, the focus is on the benchmark and CASP datasets in order to carry out an ablation study and pitch against one another HEA-US, HEA-FP, HEA-QT, HEA-TR, and HEA-AD. We include Rosetta here, as it provides a baseline. We will refer to this as the *monomorphic* experimental setting. This evaluation shows HEA-AD to be superior according to several metrics. In the second set of experiments, utilizing the metamorphic dataset we introduced in Section 3.3, we focus on the multiplicity of conformations, to which we will refer as the *metamorphic* experimental setting from now on. In this setting, we evaluate HEA-AD over several metrics, comparing it to Rosetta as a baseline method and SP-EA$^+$ to find out if the proposed adaptive mechanism improves the exploration-exploitation balance over it.

Each algorithm is run 5 times on each target to account for the stochasticity of the algorithms. We report the combined best performance over the 5 runs. Each run exhausts a fixed computational budget of $10,000,000$ energy evaluations for a total of $50,000,000$ energy evaluations for the 5 runs. Rosetta is run for $54,000,000$ energy evaluations on each target to conduct a fair comparison; each run of Rosetta exhausts $36,000$ energy evaluations and the total budget results in $1,500$ conformations over $1,500$ runs.

As is practice in EAs for conformational sampling [54], performance is measured on lowest reached energy and the lowest reached distance to the known native conformation of the target. We employ lRMSD, TM-Score, and GDT_TS to calculate the distance between the sampled conformations and the known native conformations.

To present a principled evaluation, we further strengthen our comparison with statistical significance tests. We utilize Fisher's and Barnard's exact tests for this purpose. To provide a complete picture, we also employ performance profiles for the results.

Evaluation in the Monomorphic Setting

Columns 2-7 in Table 6.4 show the lowest Rosetta *score4* energy reached by conformations generated via Rosetta, HEA-US, HEA-FP, HEA-QT, HEA-TR, and HEA-AD respectively for each target in the benchmark dataset. The entry id of the known native conformation in the PDB of each target is shown in Columns 1. Table 6.5 summarizes comparative observations that can be drawn from Table 6.4. Table 6.5(a) shows that HEA-AD achieves the lowest energy on 11/20 targets, Rosetta in 4/20, HEA-TR in 3/20, and HEA-QT in 2/20 target proteins. As Table 6.5(a) shows, HEA-AD comfortably outperforms each of the other algorithms in a head-to-head comparison. Table 6.6(a) presents the p-values for the statistical significance tests that suggest the performance improvements of HEA-AD are statistically significant at the 95% confidence level (p-values < 0.05) over other algorithms.

Table 6.5(b) shows that except HEA-AD, HEA-QT achieves better performance than all other algorithms. Table 6.6(b) shows that HEA-QT's performance improvements are statistically significant at the 95% confidence level. The better performance of HEA-QT over HEA-TR suggests that the selection pressure exerted by truncation selection is too strong which results in premature convergence. As less selection pressure is applied by quaternary tournament, HEA-QT is able to explore more of the space. On the contrary, Table 6.5(c) and Table 6.6(c) show that HEA-TR achieves significantly better performance than HEA-FP and HEA-US. These results suggest that little exploitation is performed by fitness proportional and uniform selection schemes as they apply too weak selection pressure.

We perform similar analysis for lowest lRMSD to the native conformation reached over generated conformations for a target in the benchmark dataset. Columns 2-7 in Table 6.7 show the lowest lRMSD reached by conformations generated via Rosetta, HEA-US, HEA-FP, HEA-QT, HEA-TR, and HEA-AD respectively for each target. Table 6.8 summarizes comparative observations. Table 6.8(b) shows that HEA-AD achieves the lowest lRMSD on 12/20 targets, Rosetta in 7/20, HEA-QT in 4/20, and HEA-TR in 1/20 target proteins. As Table 6.8(b) shows, HEA-AD comfortably outperforms each of the other algorithms in a

Table 6.4: Comparison of the lowest energy (measured in Rosetta Energy Unit - REU) obtained by each algorithm under comparison on each of the 20 benchmark targets is shown in Columns 2-7. The PDB ID of the known native is shown in Columns 1. The lowest energy value reached per target is marked in bold.

	Lowest Energy (REU)					
PDB ID	Rosetta	HEA-US	HEA-FP	HEA-QT	HEA-TR	HEA-AD
1ail	−29.9	−48.3	−65.5	−76.8	−56.1	**−83.6**
1aly	**−112.5**	−4.7	−8.7	−88.9	−81.1	−67.5
1aoy	−73.3	−76.5	−88.5	−110	−98.1	**−116.5**
1bq9	−46.9	−32.8	−54.6	−66.7	−50.5	**−70.8**
1c8ca	−101.4	−56.5	−64.7	−98.8	−86.4	**−111.6**
1cc5	−82.5	−58.8	−68.8	**−90.5**	−68.6	−90.2
1dtdb	**−66.5**	−18.7	−29.6	−59.9	−55	−64.1
1dtja	−72.5	−43.7	−79	−80.1	−82.2	**−88**
1fwp	−71.3	−13.3	−46.5	−82.4	**−84.4**	−71.5
1hhp	**−106.3**	−19.5	−23.1	−86.8	−104.5	−74.6
1hz6a	−117.1	−61.6	−89.8	−116.6	−130.9	**−131**
1isua	−27	−14.2	−50.1	−47.5	−46.5	**−63.1**
1sap	−107.8	−58.4	−83.2	−112.2	**−121.4**	−103.9
1tig	**−138.2**	−65.1	−82.3	−131	−128	−133.7
1wapa	−109	−32.7	−61.1	−108.8	**−132.5**	−90.4
2ci2	−37.8	−41.6	−72.3	**−110.9**	−109.8	−93.3
2ezk	−51.1	−99.3	−111.1	−129.5	−100.7	**−136.6**
2h5nd	−82.5	−131.7	−135.3	−167.3	−129	**−172.5**
2hg6	−82.5	−75.1	−78.5	−112.5	−102.6	**−118.6**
3gwl	−68.2	−89	−98.8	−117.2	−100	**−125.3**

head-to-head comparison. Table 6.9(a) presents the p-values for the statistical significance tests that suggest the performance improvements of HEA-AD are statistically significant at the 95% confidence level (p-values < 0.05) over other algorithms.

Table 6.8(b) shows that except HEA-AD, HEA-QT achieves better performance than all other algorithms. Table 6.9(b) shows that HEA-QT's performance improvements are statistically significant at the 95% confidence level except over Rosetta. Furthermore, Table 6.8(c) show that HEA-TR achieves better performance than HEA-FP and HEA-US. P-values in Table 6.9(c) suggest the performance improvement over HEA-US is statistically significant. These results mostly agree with the lowest energy evaluation results and confirm

Table 6.5: Comparison on the number of targets in the benchmark dataset in which an algorithm achieves a lower energy score than the others.

(a)
HEA-AD vs. others: 11 vs. 3 (HEA-TR), 2 (HEA-QT), 4 (Rosetta), 0 (HEA-FP), and 0 (HEA- US)
HEA-AD vs. Rosetta: 14 vs. 6 **HEA-AD vs. HEA-TR**: 14 vs. 6
HEA-AD vs. HEA-QT: 13 vs. 7 **HEA-AD vs. HEA-FP**: 20 vs. 0
HEA-AD vs. HEA-US: 20 vs. 0
(b)
HEA-QT vs. Rosetta: 13 vs. 7 **HEA-QT vs. HEA-TR**: 14 vs. 6
HEA-QT vs. HEA-FP: 19 vs. 1 **HEA-QT vs. HEA-US**: 20 vs. 0
(c)
HEA-TR vs. HEA-FP: 14 vs. 6 **HEA-TR vs. HEA-US**: 19 vs. 1

Table 6.6: Results for the 1-sided Fisher's and Barnard's tests on the comparisons presented in Table 6.5. The tests evaluate the null hypothesis that (a) HEA-AD does not achieve, (b) HEA-QT does not achieve, (c) HEA-TR does not achieve lower lowest energy on the benchmark dataset in comparison to a particular algorithm; p-values less than 0.05 are marked in bold.

Test	Rosetta	HEA-TR	HEA-QT	HEA-FP	HEA-US
(a) Fisher's	**0.01282**	**0.01282**	0.05642	**7.25E-12**	**7.25E-12**
Barnard's	**0.008299**	**0.008299**	**0.04035**	**9.10E-13**	**9.10E-13**
(b) Fisher's	0.05642	**0.01282**	N/A	**2.91E-09**	**7.25E-12**
Barnard's	**0.04035**	**0.008299**	N/A	**7.47E-10**	**9.10E-13**
(c) Fisher's	N/A	N/A	N/A	**0.01282**	**2.91E-09**
Barnard's	N/A	N/A	N/A	**0.008299**	**7.47E-10**

that the exploration-exploitation balance obtained by the adaptive selection mechanism in HEA-AD works well. However, HEA-QT's performance improvement over Rosetta and HEA-TR's performance improvement over HEA-FP in terms of lRMSD are not statistically significant which underscore the inherent inaccuracies in the energy functions.

Figure 6.3(a) shows the performance profiles of each algorithm over the benchmark dataset of 20 targets in terms of the lowest energy reached. Figure 6.3(a) shows that the

Table 6.7: Comparison of the lowest lRMSD (measured in Å) obtained by each algorithm under comparison on each of the 20 benchmark targets is shown in Columns 2-9. The PDB ID of the known native of each target is shown in Columns 1. The lowest lRMSD value reached per target is marked in bold.

	Lowest lRMSD (Å)					
PDB ID	Rosetta	HEA-US	HEA-FP	HEA-QT	HEA-TR	HEA-AD
1ail	4.5	2.1	2.1	**1.4**	**1.4**	**1.4**
1aly	12.4	11.4	11.5	**10.9**	11.2	11.4
1aoy	4	4.1	3.9	3.9	3.9	3.8
1bq9	2.9	4	3.6	3.1	3	**2.8**
1c8ca	**2.2**	4.2	4.2	3.8	4.8	4.2
1cc5	**3.7**	5.1	4.8	4.5	4.7	4.4
1dtdb	**4.2**	6.4	6.2	4.8	4.4	5.3
1dtja	**2.3**	3.7	3.3	3.2	4.2	2.5
1fwp	**2.8**	4.3	3.7	3.5	4.3	3.4
1hhp	10.1	9.1	8.6	8.3	8.8	**7.8**
1hz6a	1.9	3.1	2.8	2.4	1.9	**1.8**
1isua	6.6	6.2	5.9	**5.6**	6.6	**5.6**
1sap	**2.8**	5	4.5	4.2	3.7	4.2
1tig	**2.5**	5.2	4.4	4.3	3.2	4
1wapa	6.5	6.6	6.	5.8	6.3	**5.5**
2ci2	5.8	4.3	3.8	3.8	3.7	**3.3**
2ezk	3.6	3.3	3.1	2.8	3.4	**2.7**
2h5nd	7.4	6.3	5.6	5.8	6.2	**5.1**
2hg6	9.4	8	8	**7.9**	9.3	**7.9**
3gwl	5.8	4.8	3.8	3.8	5.4	**2.9**

probability of HEA-AD to be the optimal algorithm among these 6 algorithms is about 0.55, considerably more than any of the other algorithms. At $pr = 1.2$, HEA-AD succeeds for 85% targets. HEA-QT reaches a success of 100% at a $pr = 1.38$, while HEA-AD and HEA-TR do so at $pr = 1.45$. Rosetta's performance profile rises very slowly and reaches 100% at $pr = 3.0$. Figure 6.3(b) relates a similar analysis focusing on the lowest lRMSD to the native conformation and shows that the probability of HEA-AD to be the optimal algorithm among these 6 algorithms is about 0.6, considerably more than any of the other algorithms. At $pr = 1.3$, HEA-AD succeeds for 85% targets. HEA-QT reaches a success of 100% at a $pr = 1.8$, while HEA-AD and HEA-FP do so at $pr = 2.0$. Rosetta saturates

Table 6.8: Comparison on the number of targets in the benchmark dataset in which an algorithm achieves a lower lRMSD score than the others.

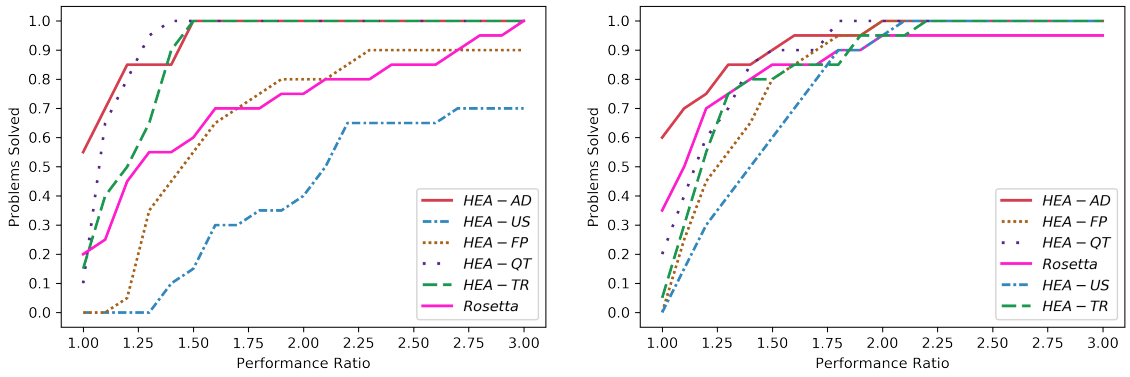
(a)
HEA-AD vs. others: 12 vs. 1 (HEA-TR), 4 (HEA-QT), 7 (Rosetta), 0 (HEA-FP), and 0 (HEA-US)
HEA-AD vs. **Rosetta**: 13 vs. 7 **HEA-AD** vs. **HEA-TR**: 16 vs. 5
HEA-AD vs. **HEA-QT**: 17 vs. 7 **HEA-AD** vs. **HEA-FP**: 20 vs. 1
HEA-AD vs. **HEA-US**: 20 vs. 2
(b)
HEA-QT vs. **Rosetta**: 11 vs. 9 **HEA-QT** vs. **HEA-TR**: 14 vs. 8
HEA-QT vs. **HEA-FP**: 19 vs. 4 **HEA-QT** vs. **HEA-US**: 20 vs. 0
(c)
HEA-TR vs. **HEA-FP**: 11 vs. 10 **HEA-TR** vs. **HEA-US**: 14 vs. 7

Table 6.9: Results for the 1-sided Fisher's and Barnard's tests on the comparisons presented in Table 6.8. The tests evaluate the null hypothesis that (a) HEA-AD does not achieve, (b) HEA-QT does not achieve, (c) HEA-TR does not achieve lower lowest lRMSD on the benchmark dataset in comparison to a particular algorithm. p-values less than 0.05 are marked in bold.

Test	Rosetta	HEA-TR	HEA-QT	HEA-FP	HEA-US
(a) Fisher's	0.05642	**0.0006159**	**0.001528**	**1.52E-10**	**1.68E-09**
Barnard's	**0.04035**	**0.0003401**	**0.0006061**	**3.73E-11**	**3.83E-10**
(b) Fisher's	0.3762	0.05548	N/A	**1.11E-06**	**7.25E-12**
Barnard's	0.3179	**0.03517**	N/A	**2.97E-07**	**9.10E-13**
(c) Fisher's	N/A	N/A	N/A	0.5	**0.02808**
Barnard's	N/A	N/A	N/A	0.4373	**0.01924**

at $pr = 2.0$ with a success for 95% targets. These results clearly establish HEA-AD as the superior algorithm.

For the CASP dataset, we can only evaluate the conformations submitted by the groups in the recent CASP competitions as we do not have access to all the conformations generated by the top 10 performing groups. We utilize lRMSD, TM-score, and GDT_TS score for the comparative analysis. Here, we focus on HEA-AD as the analysis on the benchmark



(a) (b)

Figure 6.3: Performance profiles for the algorithms on (a) lowest energy and (b) lowest lRMSD metrics.

dataset reveals superiority of HEA-AD over other algorithms. Table 6.10, 6.11, and 6.12 compares HEA-AD algorithm to the top 10 performing groups for each of the targets in the CASP dataset in terms of lowest lRMSD, highest GDT_TS, and highest TM-score reached respectively. In these tables, Columns 2-11 show the score for the top 10 groups while Column 12 shows the score for HEA-AD. Table 6.10 shows that for 6/10 targets, HEA-AD ranks in top 10 for lowest lRMSD reached. For two targets, T0859-D1 and T0957s1-D1, HEA-AD outperforms all other algorithms to rank 1st. HEA-AD also ranks 2nd for the target T0953s1-D1 and 3rd for T0897-D1. Table 6.11 shows that in 3/10 targets, HEA-AD ranks in top 10 for the highest GDT_TS score reached. For two of the targets, T0859-D1 and T0897-D1, HEA-AD outperforms all other algorithms to rank 1st. Similar results are achieved for TM-score comparison in Table 6.12; HEA-AD ranks 1st for T0859-D1, and 2nd for T0897-D1.

Evaluation on Metamorphic Dataset

In this setting, we first present a comparison of the three algorithms, Rosetta, SP-EA$^+$, and HEA-AD on the lowest-energy reached on the amino-acid sequence of each of the 13 proteins in the metamorphic dataset. As Table 6.13 shows, HEA-AD achieves the lowest

Table 6.10: Comparison of the lowest lRMSD (measured in Å) obtained by HEA-AD with top 10 performing groups in CASP competition on each of the 10 CASP targets is shown in Columns 2-12. The CASP ID of each target are shown in Columns 1. The lowest lRMSD values of HEA-AD that ranks in top 10 are marked in bold.

PDB ID	**Lowest lRMSD (Å)**										
	Gr1	Gr2	Gr3	Gr4	Gr5	Gr6	Gr7	Gr8	Gr9	Gr10	HEA-AD
T0859-D1	11.7	12.6	13.1	13.3	13.7	13.7	13.7	13.7	14.0	14.1	**9.1**
T0886-D1	2.9	3.9	3.4	4.4	4.4	5.4	5.5	5.5	5.5	5.2	6.2
T0892-D2	1.9	2.0	2.0	2.5	3.1	3.1	5.5	6.4	7.2	7.2	**6.8**
T0897-D1	7.9	8.2	10.4	10.4	11.9	12.4	12.8	12.8	13.0	13.1	**8.4**
T0898-D2	4.4	4.5	4.8	4.8	4.8	4.8	4.8	5.2	5.5	5.9	**5.4**
T0953s1-D1	4.6	8.7	8.7	8.7	8.7	8.9	9.0	9.2	9.3	9.4	**5.6**
T0953s2-D3	4.8	5.1	5.2	5.6	6.5	6.6	6.7	6.7	6.7	6.8	7.6
T0957s1-D1	6.8	8.1	8.3	8.4	8.4	8.4	8.9	8.9	8.9	9.0	**6.3**
T0960-D2	4.6	5.7	5.8	6.0	6.0	6.0	6.0	6.0	6.1	6.1	7.1
T1008-D1	1.1	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.4	2.5

Table 6.11: Comparison of the highest GDT_TS score (measured in %) obtained by HEA-AD with top 10 performing groups in CASP competition on each of the 10 CASP targets is shown in Columns 2-12. The CASP ID of each target is shown in Columns 1. The highest GDT_TS score values of HEA-AD that ranks in top 10 are marked in bold.

PDB ID	**Highest GDT_TS (%)**										
	Gr1	Gr2	Gr3	Gr4	Gr5	Gr6	Gr7	Gr8	Gr9	Gr10	HEA-AD
T0859-D1	28.32	27.66	27.66	26.77	26.55	26.55	26.55	26.55	26.55	26.55	**31.64**
T0886-D1	71.01	48.19	48.19	48.19	47.1	46.38	46.38	41.67	41.67	41.67	**42.39**
T0892-D2	50.23	48.41	47.73	47.73	47.5	47.27	46.82	46.82	46.82	46.82	42.27
T0897-D1	27.9	23.19	23.19	23.19	23.19	22.64	22.28	21.92	21.92	21.74	**28.62**
T0898-D2	75.45	70	70	70	68.64	68.18	68.18	68.18	68.18	68.18	50
T0953s1-D1	57.09	52.24	49.63	48.88	48.88	48.88	48.51	47.76	47.76	47.76	41.42
T0953s2-D3	45.7	44.35	43.01	41.94	41.4	39.78	39.52	39.52	39.52	39.52	31.49
T0957s1-D1	57.18	54.4	54.4	54.4	54.4	54.4	54.4	54.4	54.4	53.01	42.36
T0960-D2	58.63	56.84	56.55	56.55	56.55	56.55	55.95	55.36	55.36	55.36	40.18
T1008-D1	91.23	87.01	87.01	87.01	87.01	87.01	87.01	87.01	86.04	85.39	67.86

Table 6.12: Comparison of the highest TM-score obtained by HEA-AD with top 10 performing groups in CASP competition on each of the 10 CASP targets is shown in Columns 2-12. The CASP ID of each target is shown in Columns 1. The highest TM-score values of HEA-AD that ranks in top 10 are marked in bold.

PDB ID	**Highest TM-score**										
	Gr1	Gr2	Gr3	Gr4	Gr5	Gr6	Gr7	Gr8	Gr9	Gr10	HEA-AD
T0859-D1	0.3	0.29	0.28	0.28	0.28	0.28	0.28	0.28	0.27	0.27	**0.34**
T0886-D1	0.67	0.41	0.4	0.39	0.38	0.37	0.37	0.37	0.37	0.37	0.36
T0892-D2	0.55	0.52	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.47
T0897-D1	0.36	0.28	0.28	0.28	0.27	0.26	0.26	0.26	0.26	0.26	**0.35**
T0898-D2	0.67	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.59	0.56	0.4
T0953s1-D1	0.54	0.46	0.45	0.44	0.44	0.43	0.43	0.43	0.43	0.43	0.37
T0953s2-D3	0.46	0.45	0.42	0.41	0.4	0.4	0.4	0.4	0.4	0.4	0.27
T0957s1-D1	0.59	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.54	0.44
T0960-D2	0.55	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.4
T1008-D1	0.9	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.86	0.67

energy on 9/13 of the proteins in the dataset; Rosetta does so on 2/13 cases, and SP-EA$^+$ on 2/13 cases. HEA-AD comfortably outperforms Rosetta (10 vs. 3 cases) and SP-EA$^+$ (9 vs. 4 cases) in a head-to-head comparison. Panel (a) of Table 6.17 presents the p-values for statistical significance tests. These tests suggest that the performance improvements of HEA-AD in terms of lowest energy are statistically significant at the 95% confidence level (p-values < 0.05) over both algorithms.

Figure 6.4 shows the performance profiles of each of the three algorithms in terms of lowest energy. Figure 6.4 shows that the HEA-AD is the optimal algorithm on 0.7 of the proteins, easily outperforming the other two algorithms. HEA-AD "solves" all targets at a $pr = 1.2$, whereas SP-EA$^+$ and Rosetta do so at pr values of 1.32, and 1.37, respectively.

The rest of the analysis now focuses on evaluating how close each algorithm comes to each of the listed conformations for each target in the metamorphic dataset. We measure distance via lRMSD, TM-Score, and GDT_TS. We expand the list of 13 proteins into 18 test cases, where we list the known conformations as Target 1 and Target 2. This organization

Table 6.13: Comparison of the lowest energy in Rosetta Energy Units (REUs) obtained by each algorithm under comparison on each of the 13 distinct proteins in the metamorphic dataset. The lowest energy value reached is marked in bold.

Lowest Energy (REU)		
Rosetta	SP-EA$^+$	HEA-AD
-111.8	-100.6	**-126.1**
-85.6	-87	**-97.9**
-71.3	-74.9	**-98**
-76	-73.8	-64.6
-52.7	**-56.4**	-53.1
-78	-84.5	**-104.3**
-44.2	-44.6	**-50**
-147.6	-138.1	**-155.8**
-136.2	-132.2	-125.3
-161.8	-164.1	**-169.7**
-124.4	-127.3	**-130.6**
-108.5	**-108.7**	-103
-200.7	-214.7	**-222**

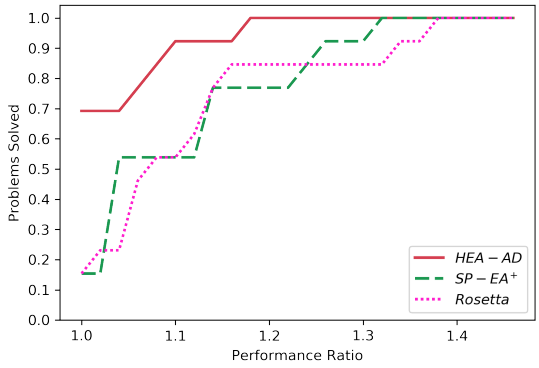


Figure 6.4: Performance profiles for the algorithms on lowest energy on the metamorphic dataset.

facilitates the exposition of our analysis. For Calmodulin, where 4 conformations have been collected, this results in 6 Target 1 – Target 2 pairs: 1cfda – 1clla, 1cfda – 2f3ya, 1clla – f3ya, 1cfda – 1lina, 1clla – 1lina, and 2f3ya – 1lina.

Table 6.14: Comparison of the lowest lRMSD obtained by each algorithm on each of the 18 target pairs in the metamorphic dataset. The lowest lRMSD value reached is marked in bold.

Lowest lRMSD (Å)					
Rosetta		SP-EA$^+$		HEA-AD	
Target 1	Target 2	Target 1	Target 2	Target 1	Target 2
8.5	7	6.6	6.1	**6.5**	**5.4**
2.2	10.2	**1.9**	1.5	2	**1.3**
5.6	7.8	4	3.3	**2.8**	**2.9**
6.1	12.4	6.4	9.2	6.4	**6.9**
11.2	9.3	8.5	**8.5**	**7.6**	9.3
12.1	**6.2**	8.8	7	**6.4**	6.9
12.3	11.9	9.9	**9.4**	**8.7**	**9.4**
6.6	10	6.7	7.9	**5.5**	**7.4**
4	7.6	**3.8**	4.4	4.1	**3.8**
10.1	31.2	9.2	13.6	**7.3**	**12.6**
9.1	17.6	9.9	**12.4**	10.5	**12.4**
6.6	9.1	6.8	6.5	**5.6**	**5.6**
6.8	8.3	3.2	**2.7**	**3**	2.8
6.8	3.8	3.2	3.6	**3**	**3.2**
8.3	3.8	**2.7**	3.6	2.8	**3.2**
6.8	4.3	3.2	3.7	**3**	**3.4**
8.3	4.3	**2.7**	3.7	2.8	**3.4**
3.8	4.3	3.6	3.7	**3.2**	**3.4**

Table 6.14 lists for each algorithm the lowest lRMSD (over all conformations sampled by an algorithm over 5 runs) to each of the targets. Table 6.14 shows that for Target 1, HEA-AD achieves the lowest lRMSD on 12/18 cases, Rosetta in 2/18 cases, and SP-EA$^+$ in 4/18 cases. HEA-AD comfortably outperforms Rosetta (15 vs. 3 cases) and SP-EA$^+$ (13 vs. 6 cases) in a head-to-head comparison. Panel (b) in Table 6.17 shows that these performance improvements are statistically significant. For Target 2, HEA-AD achieves the lowest lRMSD on 15/18 cases, Rosetta in 1/18 cases, and SP-EA$^+$ in 4/18 cases. In a head-to-head comparison, HEA-AD easily outperforms Rosetta (17 vs. 2 cases) and SP-EA$^+$ (16 vs. 4 cases). Table 6.17(c) shows that these performance improvements are statistically significant.

Figure 6.5(a) and 6.5(b) show the performance profiles of each algorithm in comparison for the lowest lRMSD metric for Target 1 and Target 2 in the metamorphic dataset respectively. Figure 6.5(a) shows that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the lowest lRMSD to Target 1) is about 0.66, considerably more than the other algorithms. HEA-AD "reaches" Target 1 on all cases at $pr = 1.2$, whereas SP-EA$^+$ and Rosetta do so at pr values of 1.4 and 3.1, respectively. Figure 6.5(b) shows that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the lowest lRMSD to Target 2) is about 0.83, considerably more than the other algorithms. HEA-AD "reaches" Target 2 on all cases at $pr = 1.15$, whereas SP-EA$^+$ does so at $pr = 1.3$; in contrast, Rosetta never reaches Target 2 on all cases in this pr range, saturating at 0.9 of the cases at $pr = 3.0$.

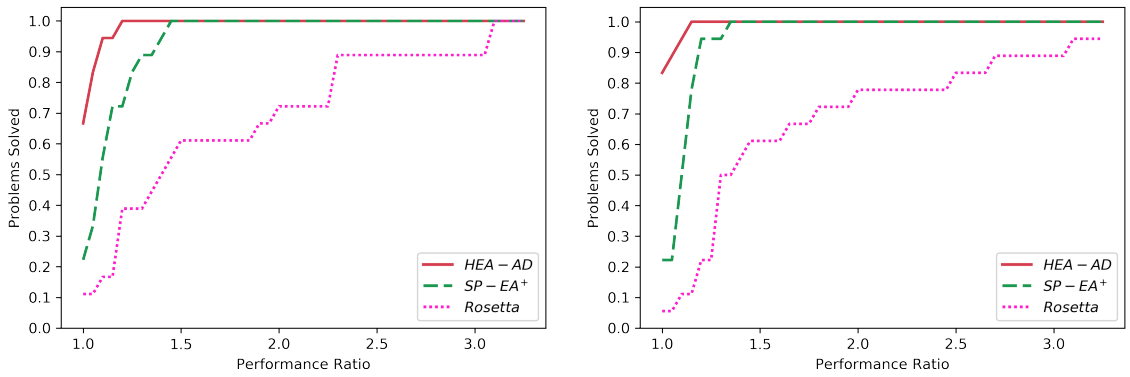


Figure 6.5: Performance profiles for the algorithms on lowest lRMSD for (a) Target 1 and (b) Target 2 on the metamorphic dataset.

Tables 6.15 and 6.16 present the comparison in terms of TM-score and GDT_TS score (higher is better) respectively. Table 6.15 shows that for Target 1, HEA-AD achieves the highest TM-score on 15/18 cases, Rosetta in 4/18 cases, and SP-EA$^+$ in 1/18 cases. In a head-to-head comparison, HEA-AD comfortably outperforms Rosetta (15 vs. 4 cases) and SP-EA$^+$ (16 vs. 2 cases) in a head-to-head comparison. Panel (d) in Table 6.17 shows

that these performance improvements are statistically significant. For Target 2, HEA-AD achieves the highest TM-score on 15/18 cases, Rosetta in 3/18 cases, and SP-EA$^+$ in 2/18 cases. In a head-to-head comparison, HEA-AD easily outperforms Rosetta (16 vs. 3 cases) and SP-EA$^+$ (17 vs. 2 cases). Panel (e) in Table 6.17 shows that these performance improvements are statistically significant.

Table 6.15: Comparison of the highest TM-score obtained by each algorithm on each of the 18 target pairs in the metamorphic dataset. The highest TM-score value reached is marked in bold.

Highest TM-score					
Rosetta		SP-EA$^+$		HEA-AD	
Target 1	Target 2	Target 1	Target 2	Target 1	Target 2
0.33	0.46	0.41	0.48	**0.42**	**0.55**
0.77	0.46	0.73	0.84	0.72	**0.87**
0.58	0.62	0.67	0.66	**0.7**	**0.69**
0.5	**0.44**	0.41	0.39	0.45	**0.44**
0.36	**0.38**	0.34	0.32	**0.36**	0.34
0.32	**0.5**	0.41	0.46	**0.44**	0.47
0.29	0.28	0.31	**0.33**	**0.33**	0.32
0.45	0.34	0.44	0.42	**0.55**	**0.45**
0.66	0.51	**0.66**	0.55	0.64	**0.59**
0.29	0.15	0.38	0.21	**0.48**	**0.23**
0.36	0.29	0.37	**0.4**	**0.4**	**0.4**
0.47	0.35	0.44	0.49	**0.48**	**0.58**
0.48	0.48	0.73	0.76	**0.74**	**0.81**
0.48	0.69	0.73	0.69	**0.74**	**0.77**
0.48	0.69	0.76	0.69	**0.81**	**0.77**
0.48	0.62	0.73	0.71	**0.74**	**0.72**
0.48	0.62	0.76	0.71	**0.81**	**0.72**
0.69	0.62	0.69	0.71	**0.77**	**0.72**

Similarly, Table 6.16 shows that for Target 1, HEA-AD achieves the highest GDT_TS on 12/18 cases, Rosetta in 4/18 cases, and SP-EA$^+$ in 3/18 cases. In a head-to-head comparison, HEA-AD comfortably outperforms Rosetta (15 vs. 4 cases) and SP-EA$^+$ (13 vs. 5 cases) in a head-to-head comparison. Panel (f) in Table 6.17 shows that these performance improvements are statistically significant. For Target 2, HEA-AD achieves the

Table 6.16: Comparison of the highest GDT_TS score obtained by each algorithm on each of the 18 target pairs in the metamorphic dataset. The highest GDT_TS value reached is marked in bold.

Highest GDT_TS (%)					
Rosetta		SP-EA$^+$		HEA-AD	
Target 1	Target 2	Target 1	Target 2	Target 1	Target 2
34.5	43	39.75	47.25	**41.5**	**53.25**
81.25	50.78	80.86	88.28	77.34	**89.84**
59.96	62.99	69.08	70.13	**75**	**71.1**
47	40.5	43.75	38	45.25	**41.5**
33.66	**35.4**	34.16	31.19	**34.65**	33.17
31.67	**48.74**	39.68	43.45	**42.68**	44.19
26.63	27.53	28.05	**30.18**	**28.83**	28.81
42.82	33.35	44.77	41.9	**50.93**	**44.91**
67.91	52.39	66.55	59.8	**67.91**	**62.5**
26.79	12.68	32.36	15.61	**40.89**	**16.43**
29.42	21.25	31.12	32.14	**34.52**	**33.33**
52.48	39.36	46.29	46.53	46.04	**52.72**
43.06	40.8	**61.02**	67.19	60.24	**70.83**
43.06	60.42	**61.02**	58.51	60.24	**65.1**
40.8	60.42	67.19	58.51	**70.83**	**65.1**
43.06	54.34	**61.02**	60.94	60.24	**62.33**
40.8	54.34	67.19	60.94	**70.83**	**62.33**
60.42	54.34	58.51	60.94	**65.1**	**62.33**

highest TM-score on 15/18 cases, Rosetta in 2/18 cases, and SP-EA$^+$ in 1/18 cases. In a head-to-head comparison, HEA-AD easily outperforms Rosetta (16 vs. 2 cases) and SP-EA$^+$ (17 vs. 1 cases). Panel (g) in Table 6.17 shows that these performance improvements are statistically significant.

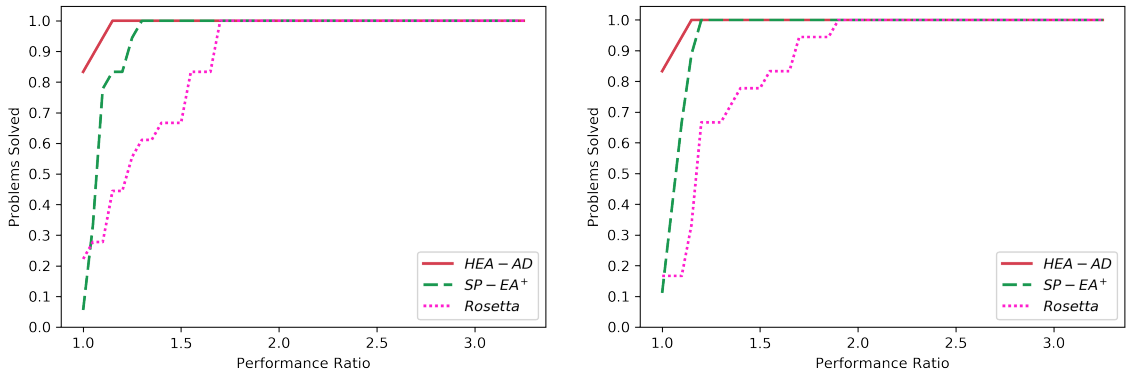
Figure 6.6(a) and 6.6(b) show the performance profiles of each algorithm in terms of the highest TM-score metric for Target 1 and Target 2 on the metamorphic dataset, respectively. Figure 6.6(a) shows that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the highest TM-score to Target 1) is about 0.83, which is considerably more than the other algorithms. HEA-AD "reaches" Target 1 on all cases at $pr = 1.15$, whereas SP-EA$^+$ and Rosetta do so at pr values of 1.3 and 1.7, respectively. Figure 6.6(b)

Table 6.17: Results for the 1-sided Fisher's and Barnard's tests on the comparisons presented in Table 6.13, 6.14, 6.15, and 6.16 on the metamorphic dataset. The tests evaluate the null hypothesis that HEA-AD does not achieve (a) lower lowest energy, (b) lower lowest lRMSD on Target 1, (c) lower lowest lRMSD on Target 2, (d) higher highest TM-score on Target 1, (e) higher highest TM-score on Target 2, (f) higher highest GDT_TS score on Target 1, (g) higher highest GDT_TS score on Target 2 in comparison to a particular algorithm. p-values less than 0.05 are marked in bold.

Test	Rosetta	SP-EA$^+$
(a) Fisher's	**0.008466**	0.05762
Barnard's	**0.004729**	**0.03778**
(b) Fisher's	**7.60E-05**	**0.02186**
Barnard's	**3.48E-05**	**0.01443**
(c) Fisher's	**3.22E-07**	**6.61E-05**
Barnard's	**1.14E-07**	**2.51E-05**
(d) Fisher's	**3.05E-04**	**2.62E-06**
Barnard's	**1.58E-04**	**9.71E-07**
(e) Fisher's	**1.48E-05**	**3.22E-07**
Barnard's	**6.46E-06**	**1.14E-07**
(f) Fisher's	**3.05E-04**	**0.009197**
Barnard's	**1.58E-04**	**0.00569**
(g) Fisher's	**2.62E-06**	**3.58E-08**
Barnard's	**9.71E-07**	**9.71E-09**

shows that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the highest TM-score to Target 2) is about 0.83, which is again considerably more than the other algorithms. HEA-AD "reaches" Target 2 on all cases at $pr = 1.15$, whereas SP-EA$^+$ and Rosetta do so at $pr = 1.2$ and $pr = 1.9$, respectively.

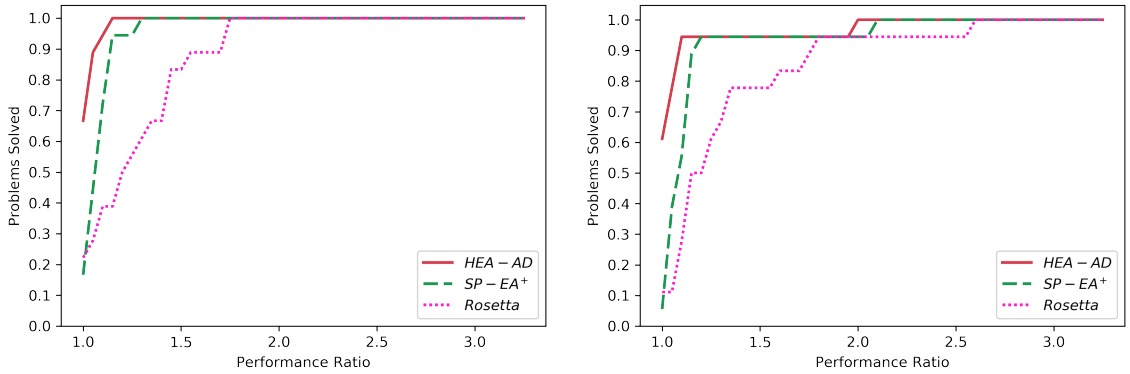
Figure 6.7(a) and 6.7(b) show the performance profiles of each algorithm in terms of the highest GDT_TS metric for Target 1 and Target 2 on the metamorphic dataset, respectively. Figure 6.7(a) shows that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the highest GDT_TS to Target 1) is about 0.66, which is higher than the other algorithms. HEA-AD "reaches" Target 1 on all cases at $pr = 1.15$, whereas SP-EA$^+$ and Rosetta do so at pr values of 1.3 and 1.75, respectively. Figure 6.7(b) shows



(a) (b)

Figure 6.6: Performance profiles for the algorithms on highest TM-score for (a) Target 1 and (b) Target 2 on the metamorphic dataset.

that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the highest GDT_TS to Target 2) is about 0.61, again higher than the other algorithms. At $pr = 1.1$, HEA-AD succeeds on 95% targets. HEA-AD "reaches" Target 2 on all cases at $pr = 2$, whereas SP-EA$^+$ and Rosetta do so at $pr = 2.1$ and $pr = 2.6$, respectively.



(a) (b)

Figure 6.7: Performance profiles for the algorithms on highest GDT_TS score for (a) Target 1 and (b) Target 2 on the metamorphic dataset.

Visualization of Conformations

The quality of the conformations obtained by HEA-AD is shown qualitatively in Fig. 6.8, which draws from the conformations obtained by HEA-AD that are closest to 4 distinct conformations of Calmodulin (PDB ids 1cfda, 1clla, 2f3ya, and 1lina, respectively). Fig. 6.8 shows that HEA-AD captures each of these conformations reasonably well (with lRMSDs shown for each).

1cfda (3Å) 1clla (2.8Å) 2f3ya (3.2Å) 1lina (3.4Å)

Figure 6.8: The HEA-AD conformation closest to the known Calmodulin native conformations under PDB ID 1cfda (left), 1clla (middle-left), 2f3ya (middle-right), and 1lina (right) is drawn in blue; the wet-laboratory conformations are drawn in olive. Rendering is performed with the CCP4mg molecular graphics software [1].

6.2.7 Summary

In this work, we present an adaptive EA for conformation ensemble generation that changes its behavior on the fly towards more exploration or exploitation as needed. The results presented above show that the adaptive selection mechanism in HEA-AD balances the exploitation and exploration effectively and samples regions of the conformation space that contain better-scoring conformations. Analysis over diverse metrics establishes the superiority of HEA-AD not only over other HEA variants, but also Rosetta and other EAs. In particular, the evaluation in the metamorphic setting shows that HEA-AD is superior and can capture diverse conformations several angstroms away when only utilizing the amino-acid sequence of a given protein (and no other conformational information about the protein at hand).

Chapter 7: Conclusions and Future Work

This thesis proposes some methods to improve de novo protein conformation ensemble generation. The focus is on obtaining an ensemble that ideally contains all the functionally-relevant conformations of a protein from the knowledge of its amino-acid sequence rather than obtaining only a single such conformation. This focus addresses the inherently dynamic view of protein systems as they switch between different active conformational states to perform biological functions.

The thesis primarily tackles the challenges of the optimization process in the vast, high-dimensional, and multimodal conformation space in the presence of an inaccurate fitness/energy function by employing evolutionary computation techniques. However, great attention is given to promote the practical use of conformation ensemble generation algorithms as well. In doing all that, the thesis takes inspiration from the algorithmic advances in EC community and protein modeling community, robot motion planning algorithms and their adaptations in molecular motion modeling, and unsupervised machine learning methods.

The work presented in this dissertation addresses three questions that are key to improve the search for functionally-relevant conformations in the rugged and multimodal energy landscape of conformations. Specifically, this thesis pursues the following questions:

- As the available energy functions are inherently inaccurate, how can we mitigate the limitations of the energy functions to generate better ensembles that contain functionally-relevant conformations?

- As conformation ensemble generation algorithms generate numerous conformations, how to effectively reduce the size of the ensemble generated by the conformation ensemble generation algorithms to improve feasibility of such algorithms?

- As it is crucial to balance limited computational resources between exploration and exploration of the multimodal conformation space to sample diverse conformations, how to achieve a proper balance of these two components of the search?

In the context of de novo protein conformation ensemble generation, the first question is addressed in Chapter 4 by first proposing a multi-objective EA that balances multiple energetic objectives to generate better quality conformation ensembles and then exploring the potential of using sequence-predicted contact information as an additional optimization objective. These approaches are shown to be more effective than single energetic objective algorithms and other multi-objective algorithms for conformational sampling.

The second question is addressed in Chapter 5 where we first show that it is possible to represent the originally generated ensemble with a reduced-size ensemble without sacrificing the quality of the original ensemble. Then, we introduce a mechanism through which conformation ensemble generation algorithms can generate such reduced ensembles on the fly. Finally, we show that such an evolving reduced ensemble has the potential to guide the search for active conformations simultaneously to enhance exploration of the conformation space.

The third question is addressed in Chapter 6 where we first focus on mapping the multimodal energy landscape by retaining diversity of the conformations through a sub-population scheme. We then employ an adaptive mechanism to obtain a better balance between exploration and exploitation which adjusts the EA selection pressure on the fly based on the characteristics of the population of conformations. This approach is shown to be very effective in finding diverse functionally-relevant conformations.

This thesis lays the groundwork for lots of further research in de novo conformation ensemble generation. The metamorphic dataset that we introduced in this thesis is the first of its kind in conformation ensemble generation and we hope it will be adopted and enriched by other researchers to serve as a benchmark dataset to evaluate future conformation sampling algorithms. The codebase associated with this thesis is developed in a

completely modular fashion from scratch and it provides a library for different algorithmic components that can be easily combined to develop different EAs and interface them with domain-specific libraries. This will make it convenient for other researchers, starting with the students in the Shehu Lab, to continue this line of work in addition to reproduce the results presented in this thesis. We believe the work in this thesis will be useful to researchers, motivate computation of multiple conformations, and prompt further research on more powerful stochastic optimization algorithms for conformation ensemble generation. An adaptive EA that takes into account complementary information like the contact information can be explored. Incorporating the reduced-size evolving map presented in this thesis to an improved conformation ensemble generation algorithm like the adaptive EA has the potential to achieve both better sampling and efficiency. We also point to the growing work on deep learning in this context. The majority of these methods are not yet able to condition to a given amino-acid sequence. Other deep learning frameworks still consider the narrow setting of one single conformation, leveraging high-inductive bias. We believe that the integration of deep learning models and EAs presents opportunities to further make inroads into what still remains a challenging problem.

Bibliography

Bibliography

[1] S. McNicholas, E. Potterton, K. S. Wilson, and M. E. M. Noble, "Presenting your structures: the CCP4mg molecular-graphics software," *Acta Cryst*, vol. D76, pp. 386–394, 2011.

[2] D. D. Boehr and P. E. Wright, "How do proteins interact?" *Science*, vol. 320, no. 5882, pp. 1429–1430, 2008.

[3] C. Soto and L. D. Estrada, "Protein Misfolding and Neurodegeneration," *Archives of Neurology*, vol. 65, no. 2, pp. 184–189, 02 2008.

[4] V. N. Uversky, "Intrinsic disorder in proteins associated with neurodegenerative diseases," *Front Biosci*, vol. 14, no. 14, pp. 5188–238, 06 2009.

[5] H. Deng, J. Y., , and Y. Zhang, "Protein structure prediction," *Int J Mod Phys B*, vol. 32, no. 18, p. 1840009, 2018.

[6] J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, and A. Tramontano, "Critical assessment of methods of protein structure prediction (casp)—round x," *Proteins: Structure, Function, and Bioinformatics*, vol. 82, pp. 1–6, 2014.

[7] E. Callaway, "'it will change everything': DeepMind's AI makes gigantic leap in solving protein structures," 2020. [Online]. Available: https://www.nature.com/articles/d41586-020-03348-4

[8] D. D. Boehr, R. Nussinov, and P. E. Wright, "The role of dynamic conformational ensembles in biomolecular recognition," *Nature Chem Biol*, vol. 5, no. 11, pp. 789–96, 2009.

[9] E. Callaway, "The revolution will not be crystallized," *Nature*, vol. 525, pp. 172–174, 2015.

[10] K. Jenzler-Wildman and D. Kern, "Dynamic personalities of proteins," *Nature*, vol. 450, pp. 964–972, 2007.

[11] R. Clausen, B. Ma, R. Nussinov, and A. Shehu, "Mapping the conformation space of wildtype and mutant H-Ras with a memetic, cellular, and multiscale evolutionary algorithm," *PLoS Comput Biol*, vol. 11, no. 9, p. e1004470, 2015.

[12] R. Clausen and A. Shehu, "A data-driven evolutionary algorithm for mapping multibasin protein energy landscapes," *J Comp Biol*, vol. 22, no. 9, pp. 844–860, 2015.

[13] E. Sapin, D. B. Carr, K. A. De Jong, and A. Shehu, "Computing energy landscape maps and structural excursions of proteins," *BMC Genomics*, vol. 17, no. Suppl 4, p. 456, 2016.

[14] E. Sapin, K. A. De Jong, and A. Shehu, "From optimization to mapping: An evolutionary algorithm for protein energy landscapes," *IEEE/ACM Trans Comput Biol and Bioinf*, 2017, doi: 10.1109/TCBB.2016.2628745.

[15] T. Maximova, E. Plaku, and A. Shehu, "Structure-guided protein transition modeling with a probabilistic roadmap algorithm," *IEEE/ACM Trans Comput Biol and Bioinf*, 2017, doi: 10.1109/TCBB.2016.2586044.

[16] T. Maximova, Z. Zhao, D. B. Carr, E. Plaku, and A. Shehu, "Sample-based models of protein energy landscapes and slow structural rearrangements," *J Comput Biol*, vol. 25, no. 1, pp. 33–50, 2017.

[17] K. Molloy, R. Clausen, and A. Shehu, "A stochastic roadmap method to model protein structural transitions," *Robotica*, vol. 34, no. 8, pp. 1705–1733, 2016.

[18] K. Molloy and A. Shehu, "A general, adaptive, roadmap-based algorithm for protein motion computation," *IEEE Trans. NanoBioSci.*, vol. 2, no. 15, pp. 158–165, 2016.

[19] P. Koehl, "Minimum action transition paths connecting minima on an energy surface," *J Chem Phys*, vol. 18, no. 145, p. 184111, 2016.

[20] D. Devaurs, K. Molloy, M. Vaisset, and A. Shehu, "Characterizing energy landscapes of peptides using a combination of stochastic algorithms," *IEEE Trans. NanoBioSci.*, vol. 14, no. 5, pp. 545–552, 2015.

[21] K. Molloy and A. Shehu, "Elucidating the ensemble of functionally-relevant transitions in protein systems with a robotics-inspired method," *BMC Struct. Biol.*, vol. 13, no. Suppl 1, p. S8, 2013.

[22] N. Haspel, M. Moll, M. L. Baker, W. Chiu, and K. L. E., "Tracing conformational changes in proteins," *BMC Struct Biol*, vol. 10, no. Suppl1, p. S1, 2010.

[23] H. Huang, E. Ozkirimli, and C. B. Post, "A comparison of three perturbation molecular dynamics methods for modeling conformational transitions," *J Chem Theory Comput*, vol. 5, no. 5, pp. 1301–1314, 2009.

[24] M. Gur, J. D. Madura, and I. Bahar, "Global transitions of proteins explored by a multiscale hybrid methodology: Application to adenylate kinase," *Biophys J*, vol. 105, no. 1643–1652, p. 184111, 2013.

[25] M. Gur, E. Zomot, M. H. Cheng, and I. Bahar, "Energy landscape of leut from molecular simulations," *J Chem Phys*, vol. 143, p. 243134, 2015.

[26] B. J. Grant, A. A. Gorfe, and J. A. McCammon, "Ras conformational switching: Simulating nucleotide-dependent conformational transitions with accelerated molecular dynamics," *PLoS Comput Biol*, vol. 5, no. 3, p. e1000325, 2015.

[27] C. B. Anfinsen, "Principles that govern the folding of protein chains," *Science*, vol. 181, no. 4096, pp. 223–230, 1973.

[28] R. Nussinov and P. G. Wolynes, "A second molecular biology revolution? the energy landscapes of biomolecular function," *Phys Chem Chem Phys*, vol. 16, no. 14, pp. 6321–6322, 2014.

[29] W. E. HART and S. ISTRAIL, "Robust proofs of np-hardness for protein folding: General lattices and energy potentials," *Journal of Computational Biology*, vol. 4, no. 1, pp. 1–22, 1997, pMID: 9109034.

[30] A. Leaver-Fay, M. Tyka, S. M. Lewis, O. F. Lange, J. Thompson, R. Jacak *et al.*, "ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules," *Methods Enzymol*, vol. 487, pp. 545–574, 2011.

[31] J. A. Maier, C. Martinez, K. Kasavajhala, L. Wickstrom, K. E. Hauser, and C. Simmerling, "ff14sb: Improving the accuracy of protein side chain and backbone parameters from ff99sb," *J Chem Theory Comput*, vol. 11, no. 8, p. 3696–3713, 2015.

[32] A. B. Rubenstein, K. Blacklock, H. Nguyen, D. A. Case, and S. D. Khare, "Systematic comparison of Amber and Rosetta energy functions for protein structure evaluation," *J Chem Theory and Comput*, pp. 6321–6322, 2018, preprint.

[33] R. Das, "Four small puzzles that rosetta doesn't solve." *PLoS ONE*, vol. 6, no. 5, p. e20044, 2011.

[34] K. Molloy, S. Saleh, and A. Shehu, "Probabilistic search and energy guidance for biased decoy sampling in ab-initio protein structure prediction," *IEEE/ACM Trans Comput Biol and Bioinf*, vol. 10, no. 5, pp. 1162–1175, 2013.

[35] A. Shmygelska and M. Levitt, "Generalized ensemble methods for de novo structure prediction," *Proceedings of the National Academy of Sciences*, vol. 106, no. 5, pp. 1415–1420, 2009.

[36] N. Akhter, W. Qiao, and A. Shehu, "An energy landscape treatment of decoy selection in template-free protein structure prediction," *Computation*, vol. 6, no. 2, p. 39, 2018.

[37] N. Akhter and A. Shehu, "From extraction of local structures of protein energy landscapes to improved decoy selection in template-free protein structure prediction," *Molecules*, vol. 23, no. 1, p. 216, 2018.

[38] D. Xu and Y. Zhang, "Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field," *Proteins: Struct, Funct, Bioinf*, vol. 80, no. 7, pp. 1715–1735, 2012.

[39] K. A. De Jong, *Evolutionary Computation: a Unified Approach.* Cambridge, MA: MIT Press, 2006.

[40] B. Olson, K. A. De Jong, and A. Shehu, "Off-lattice protein structure prediction with homologous crossover," in *Conf on Genetic and Evolutionary Computation (GECCO).* New York, NY: ACM, 2013, pp. 287–294.

[41] B. Olson and A. Shehu, "Multi-objective stochastic search for sampling local minima in the protein energy surface," in *ACM Conf on Bioinf and Comp Biol (BCB)*, Washington, D. C., September 2013, pp. 430–439.

[42] J. Skolnick, "In quest of an empirical potential for protein structure prediction," *Curr Opin Struct Biol*, vol. 16, no. 2, pp. 166–7196, 2006.

[43] T. Maximova, R. Moffatt, B. Ma, R. Nussinov, and A. Shehu, "Principles and overview of sampling methods for modeling macromolecular structure and dynamics," *PLoS Comp. Biol.*, vol. 12, no. 4, p. e1004619, 2016.

[44] V. Cutello, G. Narzisi, and G. Nicosia, "A multi-objective evolutionary approach to the protein structure prediction problem," *J Roy Soc Interface*, vol. 3, no. 6, p. 0083, 2006.

[45] J. Handle, D. B. Kell, and J. Knowles, "Multiobjective optimization in bioinformatics and computational biology," *IEEE/ACM Trans Comput Biol and Bioinf*, vol. 4, no. 2, pp. 279–292, 2007.

[46] B. Olson and A. Shehu, "Multi-objective optimization techniques for conformational sampling in template-free protein structure prediction," in *Intl Conf on Bioinf and Comp Biol (BICoB)*, Las Vegas, NV, 2014, pp. 143–148.

[47] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis, "Protein structure prediction using multiple deep neural networks in the 13th critical assessment of protein structure prediction (casp13)," *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 12, pp. 1141–1148, 2019.

[48] H. M. Berman, K. Henrick, and H. Nakamura, "Announcing the worldwide Protein Data Bank," *Nat Struct Mol Biol*, vol. 10, no. 12, pp. 980–980, 2003.

[49] B. Adhikhari, J. Hou, and J. Cheng, "DNCON2: improved protein contact prediction using two-level deep convolutional neural networks," *Bioinformatics*, vol. 34, no. 9, pp. 1466–1472, 2018.

[50] P. Bradley, K. M. S. Misura, and D. Baker, "Toward high-resolution de novo structure prediction for small proteins," *Science*, vol. 309, no. 5742, pp. 1868–1871, 2005.

[51] V. Cutello, G. Narzisi, and G. Nicosia, "A multi-objective evolutionary approach to the protein structure prediction problem," *J Roy Soc Interface*, vol. 3, no. 6, p. 0083, 2005.

[52] G. J. Zhang, G. Zhou, X, X. F. Yu, H. Hao, and L. Yu, "Enhancing protein conformational space sampling using distance profile-guided differential evolution," *IEEE/ACM Trans Comput Biol and Bioinf*, vol. 14, no. 6, pp. 1288–1301, 2017.

[53] G. Zhang, L. Ma, X. Wang, and X. Zhou, "Secondary structure and contact guided differential evolution for protein structure prediction," *IEEE/ACM Trans Comput Biol and Bioinf*, 2018, preprint.

[54] A. Shehu, "A review of evolutionary algorithms for computing functional conformations of protein molecules," in *Computer-Aided Drug Discovery*, ser. Methods in Pharmacology and Toxicology, W. Zhang, Ed. Springer Verlag, 2015.

[55] D. T. Jones, D. W. A. Buchan, D. Cozzetto, and M. Pontil, "Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments," *Bioinformatics*, vol. 28, no. 2, pp. 184–190, 2011.

[56] D. T. Jones, T. Singh, T. Kosciolek, and S. Tetchner, "Metapsicov: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins," *Bioinformatics*, vol. 31, no. 7, pp. 999–1006, 2014.

[57] S. Wang, S. Sun, Z. Li, R. Zhang, and J. Xu, "Accurate de novo prediction of protein contact map by ultra-deep learning model," *PLOS Computational Biology*, vol. 13, no. 1, pp. 1–34, 01 2017.

[58] J. Schaarschmidt, B. Monastyrskyy, A. Kryshtafovych, and A. Bonvin, "Assessment of contact predictions in CASP12: Co-evolution and deep learning coming of age," *Proteins*, vol. 86, no. Suppl 1, pp. 51–66, 2018.

[59] K. B. Santos, G. K. Rocha, F. L. Custódio, H. J. C. Barbosa, and L. E. Dardenne, "Improving de novo protein structure prediction using contact maps information," in *2017 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 2017, pp. 1–6.

[60] N. J. Cheung and W. Yu, "De novo protein structure prediction using ultra-fast molecular dynamics simulation," *PLOS ONE*, vol. 13, pp. 1–17, 11 2018.

[61] C. Zhang, S. M. Mortuza, B. He, Y. Wang, and Y. Zhang, "Template-based and free modeling of I-TASSER and QUARK pipelines using predicted contact maps in CASP12," *Proteins: Struct, Funct, and Bioinf*, vol. 86, no. S1, pp. 136–151, 2018.

[62] M. Gao, H. Zhou, and J. Skolnick, "DESTINI: A deep-learning approach to contact-driven protein structure prediction," *Sci Reports*, vol. 9, no. 3514, 2019.

[63] J. Meiler and D. Baker, "Coupled prediction of protein secondary and tertiary structure," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 21, pp. 12 105–12 110, 2003.

[64] J. DeBartolo, G. Hocky, M. Wilde, J. Xu, K. F. Freed, and T. R. Sosnick, "Protein structure prediction enhanced with evolutionary diversity: SPEED," *Protein Sci*, vol. 19, no. 3, pp. 520–534, 2010.

[65] N. Chen, M. Das, A. LiWang, and L.-P. Wang, "Sequence-based prediction of metamorphic behavior in proteins," *Biophysical Journal*, vol. 119, no. 7, pp. 1380–1390, 2020.

[66] B. Olson and A. Shehu, "An evolutionary-inspired algorithm to guide stochastic search for near-native protein conformations with multiobjective analysis," in *AAAI Workshop*, Bellevue, Washington, July 2013, pp. 32–37.

[67] A. D. McLachlan, "A mathematical procedure for superimposing atomic coordinates of proteins," *Acta Cryst A*, vol. 26, no. 6, pp. 656–657, 1972.

[68] Y. Zhang and J. Skolnick, "Scoring function for automated assessment of protein structure template quality," *Proteins: Structure, Function, and Bioinformatics*, vol. 57, no. 4, pp. 702–710, 2004.

[69] J. Xu and Y. Zhang, "How significant is a protein structure similarity with tm-score = 0.5?" *Bioinformatics*, vol. 26, no. 7, pp. 889–95, 2010.

[70] A. Zemla, "Lga: a method for finding 3d similarities in protein structures," *Nucleic acids research*, vol. 31, no. 13, pp. 3370–3374, 2003.

[71] R. A. Fisher, "On the interpretation of χ^2 from contingency tables, and the calculation of P," *J Roy Stat Soc*, vol. 85, no. 1, pp. 87–94, 1922.

[72] G. A. Barnard, "A new test of 2x2 tables," *Nature*, vol. 156, p. 177, 1945.

[73] E. Dolan and J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, 03 2001.

[74] A. Zaman and A. Shehu, "Balancing multiple objectives in conformation sampling to control decoy diversity in template-free protein structure prediction," *BMC Bioinformatics*, vol. 20, no. 1, p. 211, 2019.

[75] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans Evol Comput*, vol. 6, no. 2, pp. 182–197, 2002.

[76] A. Zaman, P. Parthasarathy, and A. Shehu, "Using sequence-predicted contacts to guide template-free protein structure prediction," in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, ser. BCB '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 154–160.

[77] A. Zaman, P. Kamranfar, C. Domeniconi, and A. Shehu, "Reducing ensembles of protein tertiary structures generated de novo via clustering," *Molecules*, vol. 25, no. 9, p. 2228, 2020.

[78] A. Zaman, P. Kamranfar, C. Domeniconi, and A. Shehu, "Decoy ensemble reduction in template-free protein structure prediction," in *ACM Conf on Bioinf and Comput Biol Workshops (BCBW): Comput Struct Biol Workshop (CSBW)*, Niagara Falls, NY, 2019, pp. 562–567.

[79] Y. Zhang and J. Skolnick, "Spicker: A clustering approach to identify near-native protein folds," *Journal of Computational Chemistry*, vol. 25, no. 6, pp. 865–871, 2004.

[80] W. Qiao, T. Maximova, X. Fang, E. Plaku, and A. Shehu, "Reconstructing and mining protein energy landscape to understand disease." Kansas City, MO: IEEE, 2017.

[81] W. Qiao, N. Akhter, X. Fang, T. Maximova, E. Plaku, and A. Shehu, "From mutations to mechanisms and dysfunction via computation and mining of protein energy landscapes," *BMC Genomics*, vol. 19, no. Suppl 7, p. 671, 2018.

[82] P. J. Ballester and W. G. Richards, "Ultrafast shape recognition to search compound databases for similar molecular shapes," *J Comput Chem*, vol. 28, no. 10, pp. 1711–1723, 2007.

[83] A. Shehu, "An ab-initio tree-based exploration to enhance sampling of low-energy protein conformations," in *Robotics: Science and Systems V*, J. Trinkle, Y. Matsuoka, and C. J. A., Eds., Seattle, WA, USA, June 2009, pp. 241–248.

[84] A. Shehu and B. Olson, "Guiding the search for native-like protein conformations with an ab-initio tree-based exploration," *Intl. J. Robot. Res.*, vol. 29, no. 8, pp. 1106–1127, 2010.

[85] K. Molloy and A. Shehu, "Interleaving global and local search for protein motion computation," in *LNCS: Bioinformatics Research and Applications*, R. Harrison, Y. Li, and I. Mandoiu, Eds., vol. 9096. Norfolk, VA: Springer International Publishing, 2015, pp. 175–186.

[86] P. Mani, M. Vazquez, J. R. Metcalf-Burton, C. Domeniconi, H. Fairbanks, G. Bal, E. Beer, and S. Tari, "The hubness phenomenon in high-dimensional spaces," in *Research in Data Sciences*, ser. Association for Women in Mathematics, E. Gasparovic and C. Domeniconi, Eds. Cham, Switzerland: Springer, 2019, vol. 17, pp. 15–45.

[87] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," in *New Directions in Statistics Physics*, L. T. Wille, Ed. Berlin, Heidelberg: Springer, 2004, pp. 273–309.

[88] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos, "Locally adaptive metrics for clustering high dimensional data," *Data Mining and Knowledge Discovery*, vol. 14, no. 1, pp. 63–97, 2007.

[89] "gmx cluster." [Online]. Available: http://manual.gromacs.org/documentation/2018/onlinehelp/gmx cluster.html

[90] X. Daura, K. Gademann, B. Jaun, D. Seebach, W. F. van Gunsteren, and A. E. Mark, "Peptide folding: When simulation meets experiment," *Angewandte Chemie International Edition*, vol. 38, no. 1-2, pp. 236–240, 1999.

[91] Q. Zhao, V. Hautamaki, and P. Fränti, "Knee point detection in bic for detecting the number of clusters," in *International conference on advanced concepts for intelligent vision systems*. Springer, 2008, pp. 664–673.

[92] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[93] G. J. McLachlan and K. E. Basford, *Mixture models: Inference and applications to clustering*. M. Dekker New York, 1988, vol. 84.

[94] D. Geary, "Mixture models: Inference and applications to clustering," *Journal of the Royal Statistical Society Series A*, vol. 152, no. 1, pp. 126–127, 1989.

[95] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in *KDD*, vol. 2000, 2000, pp. 407–416.

[96] K. C. Gowda and G. Krishna, "Agglomerative clustering using the concept of mutual nearest neighbourhood," *Pattern recognition*, vol. 10, no. 2, pp. 105–112, 1978.

[97] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[98] A. Zaman and A. Shehu, "Building maps of protein structure spaces in template-free protein structure prediction," *Journal of Bioinformatics and Computational Biology*, vol. 17, no. 06, p. 1940013, 2019.

[99] ——, "Equipping decoy generation algorithms for template-free protein structure prediction with maps of the protein conformation space," in *Proceedings of 11th International Conference on Bioinformatics and Computational Biology*, ser. EPiC Series in Computing, vol. 60. EasyChair, 2019, pp. 161–169.

[100] A. Shehu and E. Plaku, "A survey of computational treatments of biomolecules by robotics-inspired methods modeling equilibrium structure and dynamics," *J Artif Intel Res*, vol. 597, pp. 509–572, 2016.

[101] A. Zaman, K. A. De Jong, and A. Shehu, "Using subpopulation eas to map molecular structure landscapes," in *Conf on Genetic and Evolutionary Computation (GECCO)*. New York, NY: ACM, 2019, pp. 1–8.

[102] E. Sapin, K. A. De Jong, and A. Shehu, "Evolutionary search strategies for efficient sample-based representations of multiple-basin protein energy landscapes," in *IEEE Intl Conf Bioinf and Biomed*, 2015, pp. 13–20.

[103] U. Kamath, K. A. De Jong, and A. Shehu, "An evolutionary-based approach for feature generation: Eukaryotic promoter recognition," in *IEEE CEC*, A. E. Smith, Ed. IEEE Press, 2011, pp. 277–284.

[104] D. Veltri, U. Kamath, and A. Shehu, "A novel method to improve recognition of antimicrobial peptides through distal sequence-based features," in *IEEE Intl Conf on Bioinf and Biomed (BIBM)*, 2014, pp. 371–378.

[105] U. Kamath, A. Shehu, and K. A. De Jong, "Using evolutionary computation to improve svm classification," in *WCCI: IEEE World Conf. Comp. Intel.*, Barcelona, Spain, July 2010.

[106] U. Kamath, K. A. De Jong, and A. Shehu, "Selecting predictive features for recognition of hypersensitive sites of regulatory genomic sequences with an evolutionary algorithm," in *GECCO*. ACM, 2010, pp. 179–186.

[107] D. Beasley, D. R. Bull, and R. R. Martin, "A sequential niche technique for multimodal function optimization," *Evolutionary Computation*, vol. 1, no. 2, pp. 101–125, June 1993.

[108] S. Y. Yuen and C. K. Chow, "A genetic algorithm that adaptively mutates and never revisits," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 454–472, 2009.

[109] J. Zhang and A. C. Sanderson, "Jade: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[110] J. E. Vitela and O. Castanos, "A real-coded niching memetic algorithm for continuous multimodal function optimization," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, June 2008, pp. 2170–2177.

[111] J. P. K. Doye, "The network topology of a potential energy landscape: A static scale-free network," *Phys Rev Lett*, vol. 88, no. 23, p. 238701, 2002.

[112] D. J. Wales, M. A. Miller, and T. R. Walsh, "Archetypal energy landscapes," *Nature*, vol. 394, no. 6695, pp. 758–760, 1998.

[113] P. G. Debenedetti and F. H. Stillinger, "Supercooled liquids and the glass transition," *Nature*, vol. 410, no. 6825, pp. 259–267, 2001.

[114] C. L. I. Brooks, J. N. Onuchic, and D. J. Wales, "Taking a walk on a landscape," *Science*, vol. 293, no. 5530, pp. 612–613, 2001.

[115] P. Charbonneau, J. Kurchan, G. Parisi, P. Urbani, and F. Zamponi, "Fractal free energy landscapes in structural glasses," *Nat Commun*, vol. 5, no. 4725, p. 3725, 2013.

[116] L. C. Smeeton, J. D. Farrell, M. T. Oakley, D. J. Wales, and R. L. Johnston, "Structures and energy landscapes of hydrated sulfate clusters," *J Chem Theory Comput*, vol. 11, no. 5, p. 2377–2384, 2015.

[117] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," Master's thesis, University of Michigan, 1975.

[118] W. Chen and K. Y. Szeto, "Complex energy landscape mapping by histogram assisted genetic algorithm," in *Intl Conf Genet Algorithms (ICGA)*, 1987, pp. 44–49.

[119] R. Clausen and A. Shehu, "A multiscale hybrid evolutionary algorithm to obtain sample-based representations of multi-basin protein energy landscapes," in *ACM Conf on Bioinf and Comp Biol (BCB)*, Newport Beach, CA, September 2014, pp. 269–278.

[120] W. M. Spears, "Simple subpopulation schemes," in *Evolutionary Programming Conf.* World Scientific, 1994, pp. 1429–1430.

[121] Y. B. Guo and K. Y. Szeto, "Landscape mapping by multi-population genetic algorithm," in *Nature Inspired Cooperative Strategies for Optimization*, ser. Studies in Computational Intelligence. Springer, 2009, vol. 236, ch. 14, pp. 165–176.

[122] B. Ma and Y. Xia, "A tribe competition-based genetic algorithm for feature selection in pattern classification," *Applied Soft Computing*, vol. 58, pp. 328–338, 2017.

[123] A. Zaman, T. Inan, and A. Shehu, "Protein decoy generation via adaptive stochastic optimization for protein structure determination," in *IEEE Intl Conf on Bioinformatics and Biomedicine (BIBM)*, 2020.

[124] A. Zaman, T. Inan, K. A. De Jong, and A. Shehu, "Adaptive stochastic optimization to improve protein conformation sampling," *IEEE/ACM Trans Comput Biol and Bioinf*, 2021, under review.

[125] B. W. Goldman and D. R. Tauritz, "Meta-evolved empirical evidence of the effectiveness of dynamic parameters," in *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, ser. GECCO '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 155–156. [Online]. Available: https://doi.org/10.1145/2001858.2001945

[126] J. Hesser and R. Männer, "Towards an optimal mutation probability for genetic algorithms," in *Parallel Problem Solving from Nature*, H.-P. Schwefel and R. Männer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 23–32.

[127] J. Smith and T. C. Fogarty, "Self adaptation of mutation rates in a steady state genetic algorithm," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 318–323.

[128] T. Bäck, "The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm," in *Parallel Problem Solving from Nature*. Elsevier, 1992.

[129] J. Cervantes and C. R. Stephens, "Limitations of existing mutation rate heuristics and how a rank ga overcomes them," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 369–397, 2009.

[130] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. SpringerVerlag, 2003.

[131] E. Mezura-Montes and A. G. Palomeque-Ortiz, *Self-adaptive and Deterministic Parameter Control in Differential Evolution for Constrained Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 95–120.

[132] T. Bäck and M. Schütz, "Intelligent mutation rate control in canonical genetic algorithms," in *Foundations of Intelligent Systems*, Z. W. Raś and M. Michalewicz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 158–167.

[133] T. Bäck, A. E. Eiben, and N. A. L. van der Vaart, "An emperical study on gas "without parameters"," in *Parallel Problem Solving from Nature PPSN VI*, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 315–324.

[134] K. Deb and H. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover," *Evolutionary Computation*, vol. 9, no. 2, pp. 197–221, 2001.

[135] H.-P. Schwefel, "Numerische optimierung von computermodellen mittels der evolutionsstrategie," vol. 26, 1977.

[136] ——, *Evolution and Optimum Seeking: The Sixth Generation.* USA: John Wiley amp; Sons, Inc., 1993.

[137] D. B. Fogel, L. J. Fogel, and J. W. Atmar, "Meta-evolutionary programming," in *[1991] Conference Record of the Twenty-Fifth Asilomar Conference on Signals, Systems, and Computers*, 1991, pp. 540–545 vol.1.

[138] S. Meyer-Nieberg and H.-G. Beyer, *Self-Adaptation in Evolutionary Algorithms.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 47–75.

[139] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, ser. Problemata (Stuttgart). Frommann-Holzboog, 1973. [Online]. Available: https://books.google.com/books?id=-WAQAQAAMAAJ

[140] D. Thierens, "An adaptive pursuit strategy for allocating operator probabilities," in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 1539–1546.

[141] D. Schlierkamp-Voosen and H. Mühlenbein, "Strategy adaptation by competing subpopulations," in *Parallel Problem Solving from Nature — PPSN III*, Y. Davidor, H.-P. Schwefel, and R. Männer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 199–208.

[142] L. Davis, "Adapting operator probabilities in genetic algorithms," in *Proceedings of the Third International Conference on Genetic Algorithms.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, p. 61–69.

[143] F. G. Lobo and D. E. Goldberg, "Decision making in a hybrid genetic algorithm," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, 1997, pp. 121–125.

[144] A. Aleti and I. Moser, "A systematic literature review of adaptive parameter control methods for evolutionary algorithms," *ACM Comput. Surv.*, vol. 49, no. 3, Oct. 2016.

[145] K. De Jong, *Parameter Setting in EAs: a 30 Year Perspective.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–18.

[146] J. Lis, "Parallel genetic algorithm with the dynamic control parameter," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 324–329.

[147] R. S. Rosenberg, "Simulation of genetic populations with biochemical properties," Ph.D. dissertation, University of Michigan, 1967.

[148] L. Davis (ed.), *Handbook of Genetic Algorithms.* New York: Van Nostrand Reinhold, 1991.

[149] J. Lis and M. Lis, "Self-adapting parallel genetic algorithm with the dynamic mutation probability, crossover rate and population size," in *Proc. 1st Polish Nat. Conf. Evolutionary Computation.* Oficina Wydawnica Politechniki Warszawskiej, 1996, p. 324–329.

[150] B. A. Julstrom, "What have you done for me lately? adapting operator probabilities in a steady-state genetic algorithm," in *Proceedings of the 6th International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, p. 81–87.

[151] R. E. Smith, "Adaptively resizing populations: An algorithm and analysis," in *Proceedings of the 5th International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, p. 653.

[152] J. Arabas, Z. Michalewicz, and J. Mulawka, "Gavaps-a genetic algorithm with varying population size," in *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, 1994, pp. 73–78 vol.1.

[153] A. E. Eiben, E. Marchiori, and V. A. Valkó, "Evolutionary algorithms with on-the-fly population size adjustment," in *Parallel Problem Solving from Nature - PPSN VIII*, X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiňo, A. Kabán, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 41–50.

[154] D. Schlierkamp-Voosen and H. Muhlenbein, "Adaptation of population sizes by competing subpopulations," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 330–335.

[155] N. Hansen, A. Gawelczyk, and A. Ostermeier, "Sizing the population with respect to the local progress in (1,/spl lambda/)-evolution strategies-a theoretical analysis," in *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, vol. 1, 1995, pp. 80–.

[156] T. Jansen, K. A. D. Jong, and I. Wegener, "On the choice of the offspring population size in evolutionary algorithms," *Evolutionary Computation*, vol. 13, no. 4, pp. 413–440, 2005.

[157] A. Nwamba and D. Tauritz, "Futility-based offspring sizing," in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 1873–1874.

[158] C. G. Shaefer, "The argot strategy: Adaptive representation genetic optimizer technique," in *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*. USA: L. Erlbaum Associates Inc., 1987, p. 50–58.

[159] D. Whitley, K. Mathias, and P. Fitzhorn, "Delta coding: An iterative search strategy for genetic algorithms," in *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1991, pp. 77–84.

[160] D. Goldberg, K. Deb, and B. Korb, "Don't worry, be messy," in *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1991, p. 24–30.

[161] B. McGinley, J. Maher, C. O'Riordan, and F. Morgan, "Maintaining healthy population diversity using adaptive crossover, mutation, and selection," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 692–714, 2011.

[162] R. Huber-Mörk and T. Schell, "Mixed size tournament selection," *Soft Comput.*, vol. 6, pp. 449–455, 09 2002.

[163] V. Filipović, J. Kratica, D. Tošić, and I. Ljubic, "Fine grained tournament selection for the simple plant location problem," 01 2000.

[164] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.

[165] K. De Jong, *Parameter Setting in EAs: a 30 Year Perspective.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–18.

[166] R. Hinterding, Z. Michalewicz, and A. E. Eiben, "Adaptation in evolutionary computation: a survey," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, 1997, pp. 65–69.

[167] A. E. Eiben and J. E. Smith, *Parameter Control in Evolutionary Algorithms.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 129–151.

[168] H. Xie and M. Zhang, "Tuning selection pressure in tournament selection," in *Technical Report Series, School of Engineering and Computer Science.* Victoria University of Wellington, New Zealand, 2009.

[169] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.

[170] M. Sewell, J. Samarabandu, R. Rodrigo, and K. Mcisaac, "The rank-scaled mutation rate for genetic algorithms," *Information Technology - IT*, 01 2006.

[171] M. Giger, D. Keller, and P. Ermanni, "Aorcea - an adaptive operator rate controlled evolutionary algorithm," *Comput. Struct.*, vol. 85, no. 19–20, p. 1547–1561, Oct. 2007.

[172] J. Maturana, Á. Fialho, F. Saubion, M. Schoenauer, F. Lardeux, and M. Sebag, *Adaptive Operator Selection and Management in Evolutionary Algorithms.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 161–189.

[173] L. Budin, M. Golub, and D. Jakobovic, "Parallel adaptive genetic algorithm." 01 1998, pp. 157–163.

Curriculum Vitae

Ahmed Bin Zaman is currently working towards his PhD degree in Computer Science at George Mason University, USA. He received his B.S. degree in Computer Science and Engineering from Shahjalal University of Science and Technology, Bangladesh, in 2012. He worked as a team leader, researcher, and developer at Technext Limited, Bangladesh, before joining Metropolitan University, Bangladesh, as a lecturer in 2015. He received his M.S. degree in Computer Science at George Mason University, USA, in 2020. He is completing his Ph.D. in the summer of 2021. His research interests include evolutionary computation, optimization, and computational biology.

Education

- Master of Science, George Mason University, 2020
- Bachelor of Science, Shahjalal University of Science and Technology, 2012

Awards

- Distinguished Academic Achievement Award, Doctor of Philosophy in Computer Science, George Mason University, 2021.
- Distinguished Academic Achievement Award, Master of Science in Computer Science, George Mason University, 2021.
- NSF Travel Award, ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM BCB), 2019.
- Student Travel Award, ACM, Genetic and Evolutionary Computation Conference (GECCO), 2019.
- Outstanding Graduate Teaching Assistant Award, Computer Science, George Mason University, 2017.
- Runner up in National Code Warriors' Challenge, the most prestigious software development competition of Bangladesh, arranged by Ministry of Information and Communication Technology, Bangladesh, 2015.
- Shahjalal University of Science and Technology Undergraduate Scholarship for the last two years of undergraduate study for outstanding academic results.

Journal Publications

- **Ahmed Bin Zaman**, Toki Tahmid Inan, Kenneth De Jong, and Amarda Shehu. "Adaptive Stochastic Optimization to Improve Protein Conformation Sampling", IEEE /ACM Transactions on Computational Biology and Bioinformatics, 2021 (under review).
- **Ahmed Bin Zaman**, Parastoo Kamranfar, Carlotta Domeniconi, and Amarda Shehu. "Reducing Ensembles of Protein Tertiary Structures Generated De Novo via Clustering", Molecules, 2020.
- **Ahmed Bin Zaman** and Amarda Shehu. "Building Maps of Protein Structure Spaces in Template-free Protein Structure Prediction", Journal of Bioinformatics and Computational Biology, 2019.
- **Ahmed Bin Zaman**, and Amarda Shehu. "A Multi-Objective Stochastic Optimization Approach for Decoy Generation in Template-Free Protein Structure Prediction", Biophysical Journal, 2019.
- **Ahmed Bin Zaman**, and Amarda Shehu. "Balancing Multiple Objectives in Conformation Sampling to Control Decoy Diversity in Template-free Protein Structure Prediction", BMC Bioinformatics, 2019.

Conference and Workshop Publications

- **Ahmed Bin Zaman**, Toki Tahmid Inan, and Amarda Shehu. "Protein Decoy Generation via Adaptive Stochastic Optimization for Protein Structure Determination.", IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Seoul, South Korea, 2020.
- **Ahmed Bin Zaman**, Prasanna Parthasarathy, and Amarda Shehu. "Using Sequence-Predicted Contacts to Guide Template-free Protein Structure Prediction", ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM BCB), Niagara falls, NY 2019.
- **Ahmed Bin Zaman**, Parastoo Kamranfar, Carlotta Domeniconi, and Amarda Shehu. "Decoy Ensemble Reduction in Template-free Protein Structure Prediction", Computational Structural Bioinformatics Workshop (CSBW) - ACM BCB Workshops, Niagara falls, NY 2019.
- **Ahmed Bin Zaman**, Kenneth De Jong, and Amarda Shehu. "Using Subpopulation EAs to Map Molecular Structure Landscapes", Genetic and Evolutionary Computation Conference (GECCO), Prague, Czech Republic 2019.
- **Ahmed Bin Zaman**, and Amarda Shehu. "Equipping Decoy Generation Algorithms for Template-free Protein Structure Prediction with Maps of the Protein Conformation Space", International Conference on Bioinformatics and Computational Biology (BICOB), Honolulu, HI 2019 (finalist for best paper award).
- Steven Meckl, Gheorghe Tecuci, Dorin Marcu, Mihai Boicu, and **Ahmed Bin Zaman**. "Collaborative Cognitive Assistants for Advanced Persistent Threat Detection", AAAI Fall Symposium, Arlington, VA, 2017.