Combinatorial Optimization in Computational Protein Design

Thesis by

David Benjamin Gordon

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, CA

2000

(Submitted April 17, 2000)

Acknowledgements

While writing my dissertation, I have most looked forward to writing the acknowledgements. Throughout my graduate career, I have been blessed to be surrounded by amazing people—both personally and professionally. I am grateful for having had each of them pass through my life, and I have been looking forward to formally thanking them for sharing their experience with me.

Chronologically, I should first thank the people who were most instrumental in initiating my training. Fred Lee, Barry Olafson, Scott Ross, Bassil Dahiyat, and Alice Su served as my first mentors in computational protein science. Throughout my early years of graduate school, they were always willing to take the time to answer my questions, debug my code, and give me scientific and professional advice. In addition to the training, I like to think they also initiated a legacy in the lab of a high level of helpfulness and patience. In that spirit, I have tried to follow their example when I find myself in a situation in which I can help new lab members.

As I worked my way out of the computer room and into the laboratory, Sandra Malakauskas became my molecular biology mentor. She was a fantastic teacher, and she became a close friend. As I write this, she is completing her second year of medical school. I know she'll be a fantastic doctor, and I wish her all the best. Dirk Boekenkamp joined the lab a little later, and also became instrumental to my training at the bench-top. Dirk never ceased to amaze me with this almost encyclopedic knowledge of experimental subtleties. Moreover, he's a wonderful person, and I'm glad we had the opportunity to work closely together and become good friends.

For several years, my kindred spirit in the lab has been Arthur Street. Arthur was my sounding board, debugger, sanity-checker, Guan companion (along with Sandy and Leah), and most importantly, the guy whose desk I would always find myself unconsciously wandering toward whenever I needed a break from my own work. I wish to thank Arthur, not only for helping me through various bumps in my research, but also for just being there. Since his graduation, Arthur has decided to return to his Aussie homeland, and I really miss him.

For the majority of my time in the lab, Andrei Marinescu has filled in the southwest corner of the lab with Dirk and me. Bright, fun, and down to earth, Andrei was the type of guy who made it worth it to go to work every day just to be able to hang out with him, and I am thankful to have been able to work with him. I was also fortunate to have Chantal Morgan as a colleague. Chantal was a good friend, and I also wish to thank her for the important ways that she acted as my tie to the graduate student community at large. I would also like to thank Scott Ross for all of his support through the years, which has extended well beyond NMR tutoring to giving me career advice and eagerly reviewing my papers. I have also benefited greatly from having Cathy Sarisky as a fellow chemistry student. I have enjoyed our talks and collaborations, and have relied on her help several times in areas where my own chemistry knowledge has grown rusty. It has also been a privilege to work with Dan Bolon, whose biochemical knowledge (and taste in jazz) have both helped me stay on target from time to time. Pavel Strop has been a fun colleague and tremendous resource for my education about crystallography. I'd like to thank him for taking the time to answer my questions and for being the first one to introduce me to the apparatus.

I would also like Marie Ary to know how much I've learned from the experience of writing the ORBIT manual with her. It has been a surprisingly difficult, but important exercise to try to rearrange my own conceptual models and package them into a written form that anyone can understand. Marie has been my guide through that effort.

I'm not sure she has realized she's doing it, but I've always appreciated how Shannon Marshall has challenged me, always keeping me on my scientific toes. Perhaps because she mistakenly thinks I already know the answers, or maybe just because she'd like my point of view, Shannon is a continual source of original, provocative questions that I suspect are in general likely to be important to the field. I also want to thank her for co-authoring a review paper with me. I have really enjoyed working with her, and I hope that I'll have the opportunity to do so again in the future.

I am deeply indebted to Deepshikha Datta, who along with Arthur and myself comprised the β-sheet attack force. Deepshikha has been a good friend and an exciting person to have as a scientific colleague, as she always is constructing creative ways to think about the problems at hand, and has demonstrated molecular biology efficiency that I can't help but be jealous of. She also has made a significant impression on my son, who chose her name to be his first three-syllable word. I will really miss her company. Possu Huang has also been an important part of my training, in part by taking me through the exercise of reviewing the basics, and in part by saying or doing something almost every day that made me feel like my work was important. Many thanks are also due Chris Voigt, who has been instrumental in filling in the gaps in my understanding in areas of sequence optimization. I have always found Chris fascinating to talk to, and I hope that I will be able to work with him again in the future.

It has been a true honor to be able to work so closely with Niles Pierce. Niles and I worked both together and individually on algorithmic development, and he always surprised me with his brilliant solutions to the obstacles we encountered. Moreover, Niles has always done everything in his power to help me along in my own career, ranging from giving me advice to making contacts on my behalf with potential future employers. It has been a pleasure to work with him, and for all his help and companionship I will be forever thankful.

Due to the cross-disciplinary nature of my research, I have solicited help from various members of the computer science community. In particular, I would like to thank Rajit Manohar, who started pointing me in what became the right direction. I have also benefited from numerous discussions with Marc Riedel. Marc has been a great friend, and he has helped me several times as I've treaded through the computer science literature by checking my footing and helping me find my way when I get disoriented.

There are also are a number of people that have joined the lab relatively recently, and since I have enjoyed working with them, and have benefited significantly from them all, I regret that I am inadvertently cutting short our time together by leaving. To begin with, I want to be sure to thank Julia Shifman, Rhonda DiGustino, Cynthia and Brandon Carlson, Premal Shah, and John Love. In addition, I really enjoyed having Hezekiah McMurray as a brother-in-arms in dealing with the day-to-day reality that, in fact, our work depends on lapses in the de facto flakiness of computers. I have learned a tremendous amount of system administration from Hez, and I hope that he will not hesitate to call on me in the future to help him power-cycle a computer or remove a DAT tape stuck in a drive.

I am also thankful to have had the brief opportunity to get to know J.J. Plecs. J.J. is a wellspring of knowledge, scientific and otherwise, and he has the friendliest way of dispensing it when asked. I am also indebted to J.J. for always being a willing and reliable source of advice on topics ranging from writing up my work (including this dissertation) to applying for a job.

As I draw closer to the end of this long list of acknowledgements, I'm glad that I finally have a chance to offer my appreciation to my advisor, Stephen Mayo. I think I was particularly fortunate to be able to partake of the unique learning environment Steve has created, which provided me with the opportunity to garner as much experience as I could want in theoretical, computational, and experimental methods. Steve has also been an outstanding scientific mentor, a professional advisor, a strong personal advocate, and a friend. I am truly privileged to have been able to work for him, and I am profoundly grateful for all he has done for me.

My science education has roots dating back to my distant past, so this section would not be complete without acknowledgement of all the teachers I've had all along the way. Particular thanks go to Mrs. Sydel Rudner, my first grade teacher, in whose classroom I had my first glimmering of the notion that I might pursue a career in science. I also wish to thank my high school chemistry and biology mentors, Mary Catherine and Rayburn Ray. This talented duo did a fantastic job of preparing me for college, and they have been a continual source of support long since twelfth grade.

While at Caltech, I have also depended on vital support from friends outside of the lab. In particular, I would like to thank Fred and Aidyl Gonzalez-Serricchio for

opening their home to my family, and for always being on call whenever we've needed them. I hope they know that we would never hesitate to return the favor.

I am particularly fortunate to have had my whole family root for me with such enthusiasm throughout my time here. I would like Grandma A to know that her advice for me to "keep my chin up," has been well heeded, and has been an important part of my growth during my time in school. I'm particularly appreciative to my loving little brother, Josh, who had always made me feel that I make him proud. Likewise, I'm exceptionally proud of him.

Words fail me when I consider how to thank my parents. I will never be able to express enough gratitude to Mom and Dad for continually encouraging me to follow my dreams and do what I love, and for doing everything in their power to help enable me at every step of the way.

I would like to thank my son Akiva for all of the fun he has introduced to my life, as well as all the lessons he has taught me. Since he is one of the very few people who is likely to read this set of acknowledgements in fifteen or twenty years, I want to take this opportunity to tell him how much I have loved him as a baby and as a toddler, and how much I look forward to getting to know him as he grows older. Also, Akiva, if you are reading this, clean up your room!

Last, I'd like to thank my wife Leah, but I do not know how I could even begin. Leah has been an endless source of strength, inspiration, and energy, who has infused my life with humor, love, and pride. Leah has been a wellspring of support and I know she is also my biggest fan. I have loved building a life together with her here, and I feel blessed that we will continue to build a future together wherever life might take us.

Abstract

The central objective of computational protein design is to develop computational techniques for selecting amino acid sequences that fold into proteins with desired structures and functions. The work described here is directed toward addressing issues that arise in the development of computational methods for the design of solvent-exposed portions of beta-sheets. However, it is also demonstrated that the results of these investigations extend beyond specific secondary structures and in fact provide a means to address a broad spectrum of design problems. Computational issues arise from the fact that when constructing a representation of protein sequence space for analysis, significant concessions must be made with respect to the physical model and the search criteria in order to ensure that the calculation remains tractable. One of the limiting factors driving these concessions is the sheer number of combinations of amino acid identities and configurations that must be evaluated. We have therefore pursued the development and refinement of high-performance combinatorial search algorithms in order to better enable improvement of computational methods. The consequent algorithmic work consists of enhancement strategies based on combining optimization methods and instilling within them heuristics that manifest specialized knowledge of protein design problems. The results are significant performance enhancements for the well-established Dead-End Elimination algorithm, as well as two new algorithmic approaches, dubbed Branch and Terminate and Hybrid Rotamer Optimization.

Table of Contents

List of Figures and Tables

Figures

Tables

Chapter I

Introduction

Computational protein design has made startling strides in recent years. Several groups have demonstrated that stable, well-behaved proteins can be engineered in silico, using a variety of computational approaches. (See [1] and [2] for a review.) The motivations that have been driving the development of computational approaches to protein design are twofold. First is the expected promise of a general-purpose computational tool that can provide a sequence for any desired protein structure. Such a tool would make it possible to design proteins "to order," heralding a new era of bio-nanotechnology, in which the virtually unlimited number of applications in biotechnology, medicine, and industry that have been lying in wait could finally be fulfilled. The second, and perhaps more profound motivation, is the potential for the development of such a design tool to provide deeper insight into the principle nature of the sequence-structure relationship.

Attainment of both goals is dependent on ongoing refinement of design techniques. This refinement is most reliably approached through the union of simulation and experiment, in which computational strategies are systematically tested, refined, and are then validated through quantitative assessment of the physical properties of designed proteins [3]. The hope is that the quality of the design approach may be gradually improved through iterative refinement of computational techniques with feedback from experiment.

Given this framework for development, choices concerning the refinement of the computational method may be divided into three areas. The first area concerns the way

in which the protein design problem is represented, and is referred to here as the model. There are many possible implementations of the model, both in terms of the physical description and in terms of the types of constraints and liberties one may impose upon it. However, the state of the art for most groups is to use models with a full atomic-representation of the protein. Except for a few specialized cases [4, 5], the models do not allow for flexibility in the protein backbone, but side chain flexibility is incorporated by selecting amino acid rotamers from a library of discrete side chain conformations [6]. High-resolution atomic coordinates are used, which imparts generality to the model, in that arbitrary protein folds may be represented. If necessary, one may use rotamer libraries with different levels of resolution. Independent of the choice of library, the choices of amino acids may be restricted for particular positions, or one may impose sequence coupling between residue positions. Such a model provides a general, albeit rigid, framework within which an enormous variety of design problems may be described.

The second, and currently, the most significant aspect of computational protein design that is targeted for refinement is the energy expression [7]. The energy expression provides a quantitative assay capable of comparing the relative fitness of different amino acid sequences with respect to the desired target protein fold. Presently, sequences are judged by submitting their chemical and geometric information for evaluation by the energy expression. The energy expression is typically composed of potential functions based on traditional molecular mechanics and dynamics force fields. It also often includes additional non-physical potentials designed to account for other quantities suspected to be important for protein design, such as those that help prevent the selection

of sequences that might adopt other, undesired folds (as for negative design). The key to fine-tuning the energy expression is to find the proper balance of all the component potential energy terms. Communication with experiment provides a reliable mechanism through which the proper balance may be discovered.

The remaining area targeted for refinement is the search strategy. For small design problems, the best amino acid sequence may be determined simply by applying the energy expression to all possible sequence combinations and selecting the one with the lowest energy. However, because of the size of the search grows exponentially with the size of the design problem, a more sophisticated search strategy is necessary, which takes the form of a search algorithm. Using various algorithms, it is often possible to find good amino acid sequences without evaluating all possible amino acid combinations. Moreover, it is sometimes possible to find the sequence ranked best by the energy expression. (See [8] for a review.)

Although the three aspects of the computational method are quite distinct from one another, they are significantly coupled in implementation. It is often the case that refinements in one area necessitate change in another. Fundamentally, the interdependence may be ascribed to the practical limits of computing power. Currently, prospects for a computer that can precisely compute the folding of an amino acid sequence into a protein have only begun to appear on the distant horizon [9]. A computer that could perform this computation for all possible amino acid sequences is, accordingly, much further away. Fortunately, it appears that such extreme computing resources are not necessary in order to achieve successful protein design. However, preliminary design successes have relied on making significant concessions. The limits on computational

speed have limited options regarding search strategies, which in turn have imposed significant restrictions on both the model and the energy expression.

Due to this interdependence, improvement of the model and energy expression can often only be accommodated through enhancement of the search strategy. For example, if it were to be determined that a more detailed rotamer library was necessary to address some class of design problem, an associated improvement in the search strategy would also be necessary to accommodate the increase in combinatorial complexity. In this representative case, progress in development is essentially limited by the search strategy. As a result, discovery of refinement techniques that may be applied to search algorithms are of central importance to computational protein design.

This dissertation describes a set of algorithmic enhancements motivated by challenges encountered in designing solvent exposed residues of β-sheets. Chapter 2 provides an overview of the current state of the art with respect to energy expressions used in protein design, and suggests some areas in need of further refinement. Chapter 3 describes the first attempt to apply computational techniques to the design of β-sheet residues on protein surfaces.

Combinatorial optimization difficulties seemingly particular to β-sheet surfaces motivated investigation into methods of developing new algorithms as well as refining algorithms already in use. Chapter IV describes several significant enhancements that may be applied to algorithms based on the Dead-End Elimination theorem, and Chapter V describes the development of a new search algorithm based on the Branch-and-Bound technique, which we have dubbed Branch-and-Terminate.

An unexpected finding that arose from these investigations was that although the algorithmic developments were primarily motivated by difficulties with β-sheets, the resulting methods in fact had general implications for optimization of all types of secondary structure. We found that large and difficult optimization problems inaddressable by any known individual algorithm became tractable when treated with a combination of optimization techniques. At first, this combination took the form of serial application, as described in the latter part of Chapter V. Motivated by the effectiveness of this approach, we investigated the prospects for combining the algorithms at a deeper level, effectively fusing them into one algorithm. The results, as described in Chapter VI, are two versions of a very effective hybrid algorithm, one approximate and one exact.

The successes of these efforts toward algorithmic development and refinement provide the basis for some general conclusions about the application of combinatorial algorithms to protein design problems. Such observations, as summarized in Chapter VII, may be useful for the continuing development of search algorithms for protein design, and they may also have extensions to other unsolved combinatorial search problems in biology, such as the protein structure prediction problem.

In summary, the algorithmic discoveries described here illustrate major improvements in the ways that computing power may be harnessed for protein design. As a result, researchers in the field have had the ability to enhance both the model and the energy expression without being restricted to improvements in computing hardware. Looking forward, future algorithmic enhancements, especially when combined with future improvements in computer hardware, hold the promise of freeing the remaining

computational aspects from current practical restrictions, thereby enabling them to be fine-tuned to address the outstanding problems of computational protein design.

References

1. Degrado WF. 1997. Proteins from scratch. *Science* **278,** 80-81.
2. Street AG and Mayo SL. 1999. Computational Protein Design. *Structure.* **7**, R105-109.
3. Dahiyat BI and Mayo SL. 1996. Protein Design Automation. *Prot. Sci.* **5**, 895-903.
4. Su A and Mayo SL. 1997. Coupling backbone flexibility and amino acid sequence selection in protein design. *Prot. Sci.* **6,** (8) 1701-1707.
5. Harbury PB, Plecs JJ, Tidor B, Kim PS. High-resolution protein design with backbone freedom. *Science.* **282,** 1462-1467.
6. Ponder JW and Richards FM. 1987. Tertiary templates for proteins – Use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.* **193**, 775-791.
7. Gordon DB, Marshal SA, Mayo SL. 1999. Energy functions for protein design. *Curr. Opin. Struct. Biol.* **9,** (4) 509-513.
8. Desjarlais JR and Clarke ND. 1998. Computer search algorithms in protein modification and design. *Curr. Opin. Struct. Biol.* **8,** (4) 471-475.
9. Service RF. 1999 Big Blue aims to crack protein riddle. *Science.* **286**, 2250.

Chapter II

Energy Functions for Protein Design

The text of this chapter is adapted from a published manuscript that was coauthored with Shannon A. Marshall and Professor Stephen L. Mayo.

D. B. Gordon, S. A. Marshall, and S. L Mayo. *Curr. Opin. Struct. Biol.* 1999 **9** (4) 509-513

Introduction

Computational protein design is a general, closed-loop approach for finding the optimal sequence of amino acids for a desired protein fold [1]. A potential energy function that represents the dominant factors, as well as the subtleties, of protein stability is used to predict the energy of each possible amino acid sequence on a target protein structure. Current design efforts have used fixed protein backbones as target structures, with two notable exceptions [2, 3, 4]. Atomic level detail is introduced by using statistically significant sidechain conformations, called rotamers [5], to represent the flexibility of each amino acid. A variety of stochastic and deterministic search algorithms [6] are then used to find the optimal combination of amino acid sidechain rotamers on the target structure as ranked by the potential energy function. Finally, the experimentally determined stability and structure of designed proteins are analyzed and rational improvements to the potential function are implemented.

The purpose of this review is to discuss the development of protein design force fields and to survey the potential energy terms that have been used thus far. The terms fall into five broad categories. First, we discuss the energies describing packing between atoms that are not covalently bonded. Nonbonded polar interactions are considered next. We briefly survey internal coordinate energies, and finally examine solvation and

entropy, which are computed differently than in typical molecular mechanics force fields.

Force Field Requirements

Protein design presents a demanding task for a potential energy function. Design potentials must be sensitive to subtle changes in amino acid identity that are known to perturb the experimental stability of proteins. However, design force fields should not be overly sensitive to small variations in rotamer geometry, since discrete rotamers are used to model sidechain conformations. The force field also must be compatible with the computational requirements of protein design. For example, most search algorithms demand that energy terms be pairwise decomposable, and design problems with large combinatorial complexity require energy terms that can be calculated quickly.

Because the energies produced by design potentials are intended to correlate with the free energy of folding, the force field must also model the unfolded state as well as the folded state. Experimental and theoretical studies [7] indicate that unfolded proteins can sometimes have residual structure, and mutations may alter the properties of the unfolded state ensemble. However, in design calculations, the unfolded state is commonly assumed to have no residual structure: nonbonded interactions between sidechains are considered to be insignificant, the sidechains are assumed to be fully solvated, all rotamers are modeled as being equally probable, and all sequences in the unfolded state are isoenergetic.

Due to the demands posed by protein design, force fields that are widely used to perform molecular mechanics calculations, such as CHARMM [8], AMBER [9-10], and DREIDING [11], are not necessarily appropriate for design. Similarly, statistically derived pair potentials that are quite effective in structure compatibility studies [12] do

not manifest the structural sensitivity necessary for protein design. Instead, new force fields must be developed for protein design that properly balance each factor described by the potential energy function. Over the past few years, the first force fields tailored for design have been constructed. However, very few potential energy terms have been used in these force fields, and even fewer have been evaluated through comparison of design predictions and experimental results. Future progress in protein design force fields will be realized by continued systematic experimental validation of the terms comprising the potential function.

van der Waals

Packing specificity is critical for protein design. For protein core calculations, which comprise the majority of design studies, a force field that models only packing specificity is sufficient to design well-folded proteins [13-16]. Although packing can be evaluated exclusively with interatomic distance restraints [17], most design programs utilize a van der Waals potential. This potential provides a physical basis for sidechain packing specificity, thereby favoring native-like folded states with well-organized cores and selecting against disordered or molten globule states. The van der Waals energy is typically calculated with a Lennard-Jones 12--6 expression.

$$E_{vdW} = D_0\left[\left(\frac{R_0}{R}\right)^{12} - 2\left(\frac{R_0}{R}\right)^{6}\right] \qquad (1)$$

The interatomic distance, R, is computed from atomic coordinates. The equilibrium radii, R_0, and well-depths, D_0, are parameters that are defined within each force field.

Two examinations of van der Waals parameters underscore the need to tune molecular mechanics potential functions for protein design. Lazar and coworkers [16]

compared the predictive ability of variations of Hagler and AMBER van der Waals parameters for a set of ubiquitin variants with redesigned cores. United atom parameters from AMBER95 were markedly superior to the other variations when used in conjunction with a detailed rotamer library. Dahiyat and Mayo [15] generated sequences by systematically varying the scale of the atomic radii, based on the DREIDING parameter set and using rotamers with explicit hydrogen atoms. Scaling the radii by a factor of 0.90 achieved the optimal balance between packing specificity and hydrophobic collapse, as represented by a solvation term (discussed in a later section).

Hydrogen Bonding

Because the majority of computational protein design studies have focused on protein cores, electrostatic and hydrogen bonding terms have not been as thoroughly validated by experiment. Nevertheless, initial forays have proven these terms useful for the design of helical surfaces [18] and for full sequence design [19].

Hydrogen bonds are typically represented with an angle-dependent, 12-10 hydrogen bond potential,

$$E_{HB} = D_0\left[5\left(\frac{R_0}{R}\right)^{12} - 6\left(\frac{R_0}{R}\right)^{10}\right]F(\theta) \qquad (2)$$

where R_0 is the equilibrium distance, D_0 is the well depth, and R is the interatomic distance between donor and acceptor heavy atoms. The angle dependence term, $F(\theta)$, is typically $\cos^4\theta$, where θ is the donor-hydrogen-acceptor angle.

We have observed that calculations performed with the above potential will allow rotameric arrangements with non-physical hydrogen bond geometries, as shown in figure II-1. To circumvent this problem, we employ more restrictive hybridization-

dependent angle-dependence terms that enforce reasonable geometries [18].

sp^3 donor – sp^3 acceptor	$F = \cos^2\theta\cos^2(\phi - 109.5)$	$\theta > 90°$, ϕ-109.5° < 90°	(3)
sp^3 donor – sp^2 acceptor	$F = \cos^2\theta\cos^2\phi$	$\phi > 90°$	(4)
sp^2 donor – sp^3 acceptor	$F = \cos^4\theta$		(5)
sp^2 donor – sp^2 acceptor	$F = \cos^2\theta\cos^2(\max[\phi,\varphi])$		(6)

The angles ϕ and φ refer to the hydrogen-acceptor-base angle (where the base is the atom covalently attached to the acceptor) and the angle of between the normals of the planes defined by the six atoms attached to the two sp^2 centers, respectively.

A potential energy term based on the above equations allows only physically reasonable sidechain/sidechain and sidechain/backbone hydrogen bonds. Unfortunately, using a highly restrictive energy term in combination with a discrete rotamer library causes the force field to predict poor energies for some sequences that may actually form good hydrogen bond interactions.

Electrostatics

The role of electrostatics in protein stability is subject to debate. At moderate temperatures, favorable electrostatic interactions are not thought to be strong enough to compensate for the energy of desolvation [20]. In more extreme conditions, however, salt bridges may stabilize proteins [21-22]. Moreover, electrostatics may play a more significant role in defining the specificity, rather than the stability, of folding and of functional interactions [23-26].

Computational protein design efforts have not yet developed an electrostatic term intended to represent these considerations. Rather, electrostatics are used sparingly, primarily to guard against destabilizing interactions between like-charged residues. The simplest treatment of electrostatic interactions is based on Coulomb's Law, which describes the energy of two charges, Q_i and Q_j, separated by distance, R, in a medium with dielectric constant, ε.

$$E_{elec} = 322.0637\left(\frac{Q_i Q_j}{\varepsilon R}\right) \tag{7}$$

We use a distance-attenuated version of Coulomb's law with an effective dielectric constant value of *40R* and partial atomic charges that give a total coulombic energy of approximately $+/-1$ kcal/mol for the interaction between juxtaposed charged residues. Thus, electrostatic contributions to the total energy are only significant when charged atoms are in close proximity. In sharp contrast, electrostatic energy is often the largest contributor to the total energy in potentials used for molecular mechanics and dynamics calculations.

Internal Coordinate Terms

Typical molecular mechanics force fields have terms that evaluate bonds, angles, torsions, and inversions among atoms that are covalently attached. These internal coordinate or "bonded" energies must be considered when generating rotamers or modifying the protein backbone, and have been used for protein design in some cases [4, 16]. The usefulness of these terms for design, however, has not been rigorously demonstrated. Since rotamers derived from statistical analysis of protein structure databases generally have good internal coordinate energies, many design potential functions do not include them at all.

Solvation

Because the hydrophobic effect drives protein folding [27] , modeling solvation effects is critical for a protein design force field. However, the computational expense of explicitly modeling protein/solvent interactions for all sequences under consideration is prohibitively expensive. Therefore, several groups have employed approximate methods utilizing octanol-water and gas-water free energy of transfer data for each amino acid [28-29]. The experimentally measured free energies of transfer are correlated with the molecular surface area [30], shown in figure II-2. These energies are either used directly for residues in the protein core [31] or they are scaled by the change in solvent exposed surface area associated with protein folding [14, 32].

The energy required to transfer a sidechain from a solvated, unfolded protein to a partially or completely desolvated position in the folded protein is not necessarily the same as the transfer energy from water to gas or a nonpolar solvent. But, the approximate linear relationship between transfer energy and change in surface area should be correct for both cases. Dahiyat and Mayo [14] determined the optimal values for polar and nonpolar atomic solvation parameters by fitting to the experimentally determined stability of designed proteins. Inclusion of a hydrophobic burial benefit and a polar burial penalty in the protein design force field provides a significant improvement in predictive power compared to a force field with only a van der Waals term.

Two other considerations have affected the formulation of a protein design solvation potential. First, a negative design term that penalizes exposure of nonpolar surface area is sometimes used [15, 33]. Although nonpolar exposure should not destabilize a protein, it can lead to aggregation or misfolding. Therefore, a nonpolar exposure penalty is required to limit the amount of exposed nonpolar surface area at boundary and surface positions [34]. Second, many optimization algorithms require that energy terms be pairwise decomposable, but pairwise calculation of buried surface areas

leads to significant overcounting. Street and Mayo have developed a pairwise expression with one scalable parameter that closely reproduces both the true buried area and the true exposed solvent accessible surface areas [35].

Entropy

A simple entropy term is sometimes incorporated into protein design potential functions [31, 32]. The change in sidechain entropy upon folding is modeled as the change in number of rotatable bonds, making the assumption that conformational freedom is completely restricted in the folded state. The unfolded state entropies are calculated either by assuming that all rotamers are equally populated or by fitting to semi-empirical estimates [36]. Inclusion of an entropy term based on the number of rotatable bonds did not significantly improve correlation between predicted and observed stabilities of the GCN4-p1 coiled coil core [14]. This simple model for entropy may have failed because it neglects residual sidechain entropy in folded proteins, as well as possible residual structure in the unfolded state.

Looking Forward

Protein design force fields have been successful, in part, because of their stringency. Restrictive functions such as the van der Waals and the hybridization-dependent hydrogen-bond potential, in particular, result in a very high rejection rate, and a significant false-negative rate. Fortunately, many design force fields also show a low false-positive rate. Therefore, sequences that are selected in protein design studies tend to fold properly, even though many other equally acceptable sequences are rejected.

Because of the high false-negative rate, potential functions derived through protein design efforts may not be suitable for folding studies. To gain a

deepernderstanding of the determinants of protein stability, it is therefore important to lower the false-negative rate. Softening of the restrictive potentials could result in design models that more accurately describe the fundamental relationship between sequence, structure, and stability.

Acknowledgements

We wish to thank A. G. Street for helpful comments on the manuscript. This work was supported by the Howard Hughes Medical Institute (SLM), the Helen G. and Arthur McCallum Foundation (DBG), an NIH NRSA Training Grant, and the Caltech Initiative in Computational Molecular Biology program awarded by the Burroughs Wellcome Fund (SAM).

References

1. Street AG and Mayo SL. 1999. Computational protein design. *Structure.* **7,** R105-R109.
2. Harbury PB, Tidor B, Kim PS. 1995. Repacking protein cores with backbone freedom: structure prediction for coiled coils. *Proc. Natl. Acad. Sci. USA.* **92**, 8408-8412.
3. Su A and Mayo SL. 1997. Coupling backbone flexibility and amino acid sequence selection in protein design. *Protein. Sci.* **6**, 1701-1707.
4. Harbury PB, Plecs JJ, Tidor B, Alber T, Kim PS. 1998. High-resolution protein design with backbone freedom. *Science.* **282**, 1462-1467.
5. Ponder JW and Richards FM. 1987. Tertiary templates for proteins - use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.* **193**, 775-791.
6. Desjarlais JR and Clarke ND. 1998. Computer search algorithms in protein modification and design. *Curr. Opin. Struct. Biol.* **8**, 471-475.
7. Dill KA and Shortle D. 1991. Denatured states of proteins. *Ann. Rev. Biochem.* **60**, 795-825.
8. Brooks BR, Bruccoleri RE, Olafson BD, States DJ, Swaminathan S, Karplus M. 1983. CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.* **4**, 187-217.
9. Weiner SJ, Kollman PA, Case DA, Singh UC, Ghio C, Alagona G, Profeta SJ, Weiner P. 1984. A new force field for molecular mechanical simulation of nucleic acids and proteins. *J. Am. Chem. Soc.* **106**, 765-784.
10. Cornell WD, Cieplak P, Bayly CI, Gould IR, Merz KM, Jr., Ferguson DM,

Spellmeyer DC, Fox T, Caldwell JW, Kollman PA. 1995. A second-generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Am. Chem. Soc.* **117**, 5179-5197.

11. Mayo SL, Olafson BD, Goddard WA. III. 1990. DREIDING - a generic force-field for molecular simulations. *J. Phys. Chem.* **94**, 8897-8909.
12. Bowie JU, Luthy R, Eisenberg D. 1991. A method to identify protein sequences that fold into a known three-dimensional structure. *Science* **253**, 164-170.
13. Desjarlais JR and Handel TM. 1995. *De novo* design of the hydrophobic cores of proteins. *Prot. Sci.* **4**, 2006-2018.
14. Dahiyat BI and Mayo SL. 1996. Protein design automation. *Prot. Sci.* **5**, 895-903.
15. Dahiyat BI and Mayo SL. 1997. Probing the role of packing specificity in protein design. *Proc. Nat. Acad. Sci. USA* **94**, 10172-10177.
16. Lazar GA, Desjarlais JR, Handel TM. 1997. *De novo* design of the hydrophobic core of ubiquitin. *Prot. Sci* **6**, 1167-1178.
17. Jiang X, Bishop EJ, Farid RS. 1997. A *de novo* designed protein with properties that characterize natural hyperthermophilic proteins. *J. Am. Chem. Soc.* **119**, 838-839.
18. Dahiyat BI, Gordon DB, Mayo SL. 1997. Automated design of the surface positions of protein helices. *Prot. Sci.* **6**, 1333-1337.
19. Dahiyat BI and Mayo SL. 1997. *De novo* protein design: fully automated sequence selection. *Science* **278**, 82-87.
20. Hendsch ZS and Tidor B. 1994. Do salt bridges stabilize proteins- a continuum electrostatic analysis. *Prot. Sci* **3**, 211-226.
21. Elcock AH. 1998. The stability of salt bridges at high temperatures: implications for hyperthermophilic proteins. *J. Mol. Biol.* **284**, 489-502.

22. de Bakker PIW, Hunenberber PH, McCammon JA. 1999. Molecular dynamics simulations of the hyperthermophilic protein Sac7d from *Sulfolobus acidocaldarius*: contribution of salt bridges to thermostability. *J. Mol. Biol.* **285**, 1811-1830.

23. Lumb KJ and Kim PS. 1995. A buried polar interaction imparts structural uniqueness in a designed heterodimeric coiled coil. *Biochemistry* **34**, 8642-8648.

24. Schneider JP, Lear JD, DeGrado WF. 1997. A designed buried salt bridge in a heterodimeric coiled coil. *J. Am. Chem. Soc.* **119**, 5742-5743.

25. Sindelar CV, Hendsch ZS, Tidor B. 1998. Effects of salt bridges on protein structure and design. *Prot. Sci.* **7**, 1898-1914.

26. Spek EJ, Bui AH, Lu M, Kallenbach NR. 1998. Surface salt bridges stabilize the GCN4 leucine zipper. *Prot. Sci.* **7**, 2431-2437.

27. Dill KA. 1990. Dominant forces in protein folding. *Biochemistry* **29**: 7133-7155.

28. Fauchère J-L and Plicska V. 1983. Hydrophobic parameters of amino-acid side-chains from the partitioning of n-acetyl-amino-acid amides. *Eur. J. Med. Chem.* **18**, 369-375.

29. Ooi T, Oobatake M, Nementhy G, Scheraga HA. 1987. Accessible surface areas as a measure of the thermodynamic parameters of hydration of peptides. *Proc. Natl. Acad. Sci. USA* **84**, 3086-3090.

30. Wesson L and Eisenberg D. 1992. Atomic solvation parameters applied to molecular dynamics of proteins in solution. *Prot. Sci.* **1**, 227-235.

31. Hellinga HW and Richards FM. 1994. Optimal sequence selection in proteins of known structure by simulated evolution. *Proc. Natl. Acad. Sci. USA* **91**, 5803-5807.

32. Kono H, Nishiyama M, Tanokura M, Doi J. 1998. Designing the hydrophobic core of *Thermus flavus* malate dehydrogenase based on Side-chain packing. *Prot. Eng.* **11**,

47-52.

33. Sun S, Brem R, Chan HS, Dill KA. 1995. Designing amino acid sequences to fold with good hydrophobic cores. *Prot. Eng.* **8**, 1205-1213.

34. Malakauskas SM and Mayo SL. 1998. Design, structure and stability of a hyperthermophilic protein variant. *Nat. Struct. Biol.* **5**, 470-475.

35. Street AG and Mayo SL. 1998. Pairwise calculation of protein solvent accessible surface areas. *Fold. Des* **3**, 253-258.

36. Sternberg MJE and Chickos JS. 1994. Protein side-chain conformational entropy derived from fusion data - comparison with other empirical scales. *Prot. Eng.* **7**, 149-155.

Figure II-1: An example of a non-physical hydrogen bond geometry that can be selected when a hydrogen bond potential dependent only on θ is used for protein design. A more restrictive hydrogen bond potential, described in equations 2 through 6, correctly predicts that no favorable interaction is present because $\phi = 90°$.

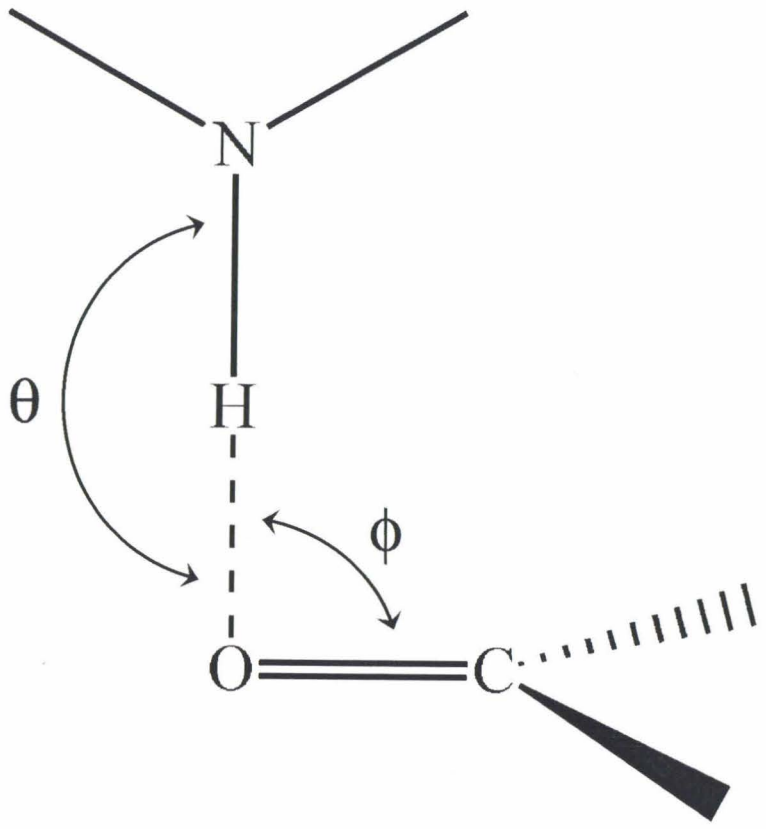
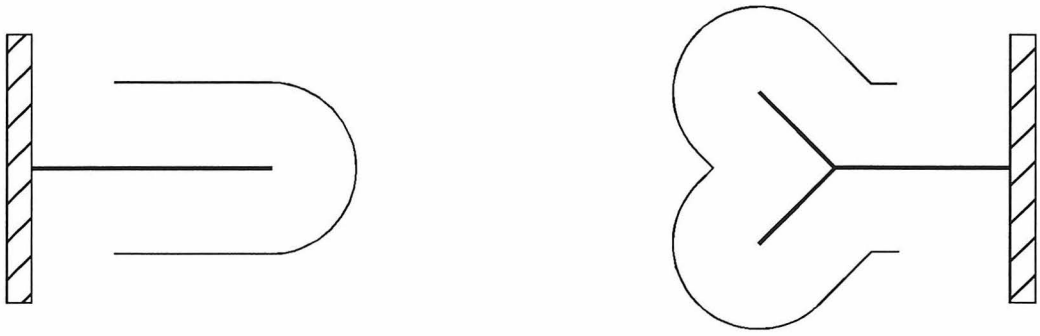
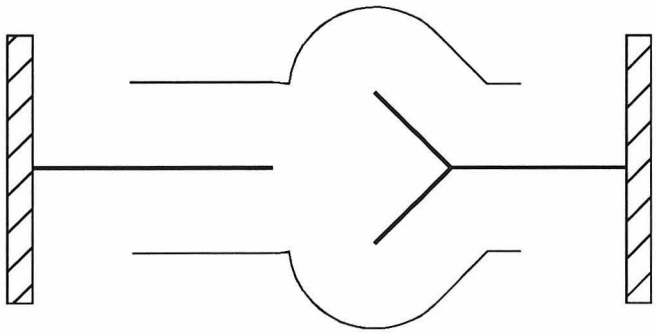

N
θ
H
ϕ
O
C

Figure II-2: (a) Unfolded or reference exposed surface areas for two sidechain rotamers. (b) Folded exposed surface area for the rotamer pair. (c) Buried surface area for the rotamer pair, which is calculated by subtracting (b) from (a).

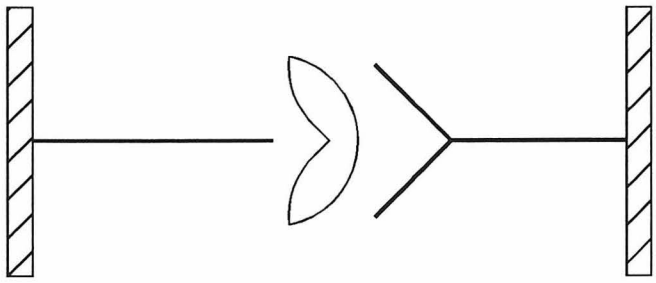
(a)



(b)



(c)



Chapter III

Computational Design of β-sheet Surfaces

Abstract

We have investigated the utility of incorporating secondary-structure propensity and a hydrogen-bond potential into the computational design of β-sheet surfaces. Success in the design of α-helical surfaces suggests that side chain placement algorithms can be applied to solvent-exposed residues by adding a hydrogen-bond potential to the energy expression. However, unlike helices, many of the large polar or charged residues favored by such energy expressions are known to destabilize β-sheets. To compensate, statistically-derived propensity energies were incorporated into a sequence selection algorithm based on the Dead End Elimination Theorem. Twelve surface-exposed β-sheet positions of the immunoglobulin-binding domain of streptococcal protein-G were redesigned using energy potentials in which the contribution of β-sheet propensity was systematically incremented. The resulting designed proteins were obtained via bacterial expression and characterized by circular dichroism and NMR spectroscopy. The designed proteins all folded reversibly, with melting temperatures ranging from 29 °C to 67 °C. Proteins designed with highly scaled propensity had significantly higher melting temperatures than those designed with lower propensity, though they had fewer predicted inter-residue electrostatic interactions. However, none of the designed proteins approached the thermal stability of the wild type protein. We conclude that propensity and conventional electrostatic considerations are both critical for the design of β-sheet

surfaces, but also that they are insufficient for the reproduction of native-like stability, even when used together.

Introduction

Several groups have reported approaches to protein design using computational methods to find amino acid sequences that adopt the conformation of a desired protein fold [1-5]. In general, these techniques have been applied only to the cores of proteins, for which packing and hydrophobic burial considerations are dominant. On protein surfaces, however, interactions between side chains are different than in protein cores. To begin with, side chains make fewer contacts. This is because neighboring side chains are spaced further from each other than they are in protein cores, and they are often oriented pointing in the same direction or even away from one other. Moreover, surface residues may interact with the accessible solvent molecules, which are typically regarded as excluded in core design calculations.

Due to the reduced number of inter-residue contacts, packing interactions cannot impart the same restrictions on sequence that they do for protein core design calculations. Therefore, the contributions from other physical forces, such as those arising from electrostatic interactions or potential interactions with the solvent, are not overwhelmed by packing energies. As a result, inaccuracies in the representation of such energetic considerations have a more significant impact on the quality of the sequence prediction method. Therefore, careful refinement of the energy potential is likely to be necessary.

Our first step toward such refinement has been to enhance the representation of electrostatic interactions with the addition of a hybridization dependent hydrogen-bond

potential into our previous scoring function. An energy expression containing this potential has been successfully applied to the design of surface positions of α-helices [6]. However, we have anticipated that additional considerations may arise when extending this approach to the design of β-sheet surfaces.

The prospects for accurate sequence prediction of the energy expression may be judged through comparison of its predictions to known trends in the composition of the different types of secondary structure. A convenient quantity for comparison is the intrinsic secondary-structure propensity of amino acids towards α-helices and β-sheets. There is good agreement between intrinsic propensity scales derived both from statistical studies [7] and host-guest studies on various model systems [8-11]. These studies have established that β-sheets have clear preferences for amino acid composition.

In its form as applied to helical surface design, the energy expression exhibits a bias toward large and polar or charged amino acids, since they can more easily participate in packing and electrostatic interactions rewarded by the van der Waals and electrostatic potentials. The result is a serendipitous bias toward amino acids that have high helical propensity, which aids helical surface design. Unfortunately, the bias has the side effect of discriminating against amino acids that have high β-sheet propensity, such as small β-branched residues like Thr or Ile, in favor of amino acids thought to disrupt β-sheets, such as Asp. To compensate for this bias, we chose to incorporate β-sheet propensities directly into the energy expression.

To find the proper balance between propensity and electrostatic interactions, we calculated sequences covering a range of relative weightings of electrostatic and propensity considerations. The resulting designed sequences varied from consisting

almost entirely of high-propensity amino acids to others consisting mostly of hydrogen-bond and ion-pair forming side chains.

Method

Propensities

Intrinsic propensities have been calculated by host-guest studies on a zinc-finger peptide [10] and in different host environment on the surface of protein-G [8, 9, 11]. The propensity values for studies with host sites on non-edge strands host-sites correlate closely with each other, but not with energies from the edge strand study. The difference is attributed to differences in tertiary context that are not presently well defined. Thus there are several different data sets from which to select propensity energies.

Munoz and Serrano [7] have produced a set of intrinsic propensities from a statistical survey of the protein database. Pseudo energies were calculated from the fraction of residues having ϕ and ψ angles consistent with β-strand conformations (see figure III-1). This set of propensities was selected for incorporation into the energy expression because it was derived without bias toward any specific protein, and it showed good agreement with the experimental scales.

To make it possible to emphasize the differences in propensity when necessary, an exponential form was used for incorporation into the energy expression,

$$E_{\beta-sheet} = 10^{N_\beta(\Delta G^o_{aa} - \Delta G^o_{val})-1} \tag{1}$$

where ΔG° is the statistically-derived β-sheet propensity and N_β is the scale factor used to control the relative contribution of the propensity energy to the overall energy. The scale

is normalized by subtracting the propensity of the most preferred amino acid for β-sheets, Val. Thus the value of N_β may be considered as the extent to which non-valine residues are penalized for occupying the β-sheet surface (see figure III-2). To determine the best value for the scale factor N_β, the calculation was repeated for different values of N_β ranging from 0.0 to 4.0.

Calculation

The 56-residue immunoglobulin-binding domain of streptococcal protein-G [12] (PDB id: 1pga) was selected as the model system for this study. The protein consists of four β-strands that create a cradle for a single α-helix. The β-sheet surface was considered a good system for design because it is provides a template for a potentially extensive system of interactions, yet it is relatively isolated from the rest of the protein. Sequences were computed for twelve of the fourteen solvent-exposed β-sheet positions arranged in a 3-residue by 4-residue array.

Coordinates for the structure 1pga were obtained from the Protein Data Bank and were used as the template for all calculations. Rotamers were selected from a backbone dependent library [13]. For each value of N_β, an algorithm based on the Dead End Elimination theorem [14] was used to determine the best possible arrangement of hydrophilic rotamers for the twelve designed positions on the β-sheet surface (positions 4, 6, 8, 13, 15, 17, 42, 44, 46, 51, 53, and 55). Propensities were only applied to positions that were flanked on either side by residues also in β-sheet conformations (positions 4, 6, 15, 17, 44, and 53). A Lennard-Jones 12-6 potential using atom radii scaled by 0.9 and well-depths from the DEEIDING force-field [15] was used to represent packing, and

electrostatic interactions were implemented using a weak coulombic potential and a hybridization-dependent hydrogen-bond potential as described in [6].

Protein Expression & Purification

Genes for the designed sequences were obtained through successive applications of inverse PCR [16] and QuickChange™ (Stratagene) techniques to the wild-type gene in a pAED-4 vector (courtesy Dan Minor). Proteins were expressed in BL21DES E. Coli and extracted by freeze-thaw [17]. Crude extracts were mixed with equal volumes of acetonitrile and the precipitants were removed by centrifugation. The supernatant was concentrated and then purified by reverse phase HPLC on a Vydac C8 column. The purified proteins were then lyophilized and stored at –20 °C.

CD and NMR

To measure stability and to confirm native-like structure, designed proteins were studied by CD and NMR. CD spectra were measured on an Aviv 62DS spectrometer at pH 5.0 in 50 mM phosphate. Thermal melts were performed in the same buffer with 2.0°C increments. NMR samples were prepared in the same buffer in 90/10 H_20/D_20 on a Varian Unityplus 600 MHz spectrometer.

Results

Calculations covering a range of values of N_β produced six different amino acid sequences (See Table 1). Designed sequences had between one and five amino acids in common with the wild-type sequence, and they maintained approximately the same

charge balance in each case. Calculation results ranged from sequences having few residues with good β-sheet propensity and many modeled electrostatic interactions, to sequences with high threonine content and few modeled electrostatic interactions (see figure III-3). As N_β was increased, more threonines were selected, which were too short to participate in most inter-strand interactions.

All sequences had native-like CD spectra and folded reversibly, with melting temperatures ranging from 36°C to 67°C (See figure III-4). The sequences calculated with high propensity (s12_1.4 & s12_2.0) were significantly more stable than the other sequences (see figure III-5). Well-dispersed NMR spectra were observed for s12_1.1 and s12_1.3 at low temperatures, demonstrating ordered structure even for the least stable proteins (data not shown).

Discussion

Although the designed sequences did not exhibit native-like stability, a few general trends may be inferred from the dependence of stabilities on the weighting of the force-field. First, with respect to the competing drives toward interaction-forming residues and high-propensity residues, it is clear that both considerations can contribute to protein stability. This is evidenced by the observed rise in melting temperatures toward the N_β extremes, as each consideration precludes the other. Sequences s12_1.3, s12_1.1, and s12_0.0 get progressively more stable as the number of pair interactions increases. Similarly, increasing numbers of threonine residues stabilize sequences s12_1.4, s12_1.9, and s12_2.0.

Secondly, it appears that when confronted with the choice, residues with high propensity are better able to stabilize β-sheet surfaces than are residues that form inter-residue interactions. This is demonstrated by the large gap in thermal stability (20°C) between the most stable proteins at either N_β extreme (see figure III-5). This suggests that design strategies might need to weigh propensities more heavily than electrostatic pairing.

Third, it is clear that even when considered together, propensity and pair interactions supply insufficient criteria for the design of sequences that approach or exceed native-like stability in the context of our computational design method. This could be due to any of a number of possible shortcomings of the method, concerning both the energy expression and the representation of the protein.

One concern is that inaccuracies of the electrostatic and hydrogen-bond potentials are causing the technique to erroneously discard critical interactions. Of particular concern is the occasional loss of interactions between β-sheet side chains and the backbone atoms of neighboring turns. These interactions may be critically important for stability because they involve a stabilizing side chain-backbone interaction, which is not thought to incur the same cost of loss of entropy as a similar side chain–side chain interaction [18]. To test the sensitivity of the turns to β-sheet residues, we analyzed several point mutants (see Appendix A). Based on this work, as well as on the known stability of host sites for propensity that disrupt the same positions [9], we believe that β-sheet residues potentially involved with turns may need to be designed independently of the remainder of the sheet, or at least with special attention paid to the nearby turn.

Also pertaining to the energy expression, it may be necessary to add additional potential terms to better approximate the complex physics on the protein surface. For example, it may help to quantitatively incorporate the context effects described by Minor and Kim [9] in terms of hydrophobic burial and exposure. Moreover, it has been suggested [19] that hydrophobic interactions of the surfaces of proteins can have stabilizing effects. Further motivation comes from the presence of a hydrophobic isoleucine at position 6 in the wild-type sequence, as well as from a host-guest study [20] in which this position was successfully replaced by various other hydrophobic side chains.

Since the time that the work presented here was completed, we have used both suggestions above in conjunction with a Z-score approach to optimize an energy expression for β-sheet surfaces [21]. In the new study, an energy expression that includes hydrophobic burial and exposure terms is optimized on eight β-sheet positions that do not neighbor turns. Proteins designed using the newer approach are more stable than the proteins designed here, validating these two suggestions. However, even these proteins do not achieve native-like thermal stability.

Two remaining issues may be useful for consideration in future design work. First is that the presence of networks of electrostatic interactions may provide more significant contributions to thermal stability than groups of electrostatic pairs. A side chain involved in a network obtains multiple electrostatic benefits, but it is penalized only once for entropy loss. Therefore, sequences that favor the formation of networks may derive greater stability from electrostatics than those that do not. Unfortunately, in its current form, the pair-wise representation of the model, in conjunction with the

restrictions imposed by the DEE algorithm, prevent the direct detection of hydrogen bond networks. Other search algorithms, however, such as branch-and-bound algorithms, may be modified to overcome this obstacle.

Related to the concern that interactions may too complex to be captured by simple two-body expressions is the reality that side chains on protein surfaces are much less ordered than those in protein cores. It is therefore reasonable to question the validity of representing the amino acids on a protein surface with single rotamers. While it has not been demonstrated that the lack of side-chain motion in the model is an important obstacle to computational β-sheet design, it may be worthwhile to consider a flexible-rotamer model [22] approach to attempt to capture some of the disordered nature of solvent-exposed side chains.

Conclusions

We have demonstrated that introduction of either a hydrogen-bond potential or an explicit representation of intrinsic β-sheet propensity may be used to compute β-sheet surface sequences that produce well-behaved proteins. However, these considerations compete with one another, and when both are weighted equally, proteins with low thermal stability result. Moreover, even the most stable of the designed proteins does not approach the thermal stability of the wild-type protein. As a result, it is clear that new considerations and approaches are necessary to appropriately tune computational design methods for the design of β-sheet surfaces.

References

1. Dahiyat BI, Mayo SL. 1996. Protein design automation. Protein Science **5**, 895-903.
2. Desjarlais JR, Handel TM. 1995. De novo design of the hydrophobic cores of proteins. *Protein Science* **4**, 2006-2018.
3. Harbury PB, Tidor B, Kim PS. 1995. Repacking protein cores with backbone freedom: structure prediction for coiled coils. *Proc. Nat. Acad. Sci. USA* **92**, 8408-8412.
4. Hurley JH, Baase WA, Matthews BW. 1992. Design and structural analysis of alternative hydrophobic core packing arrangements in bacteriophage T4 lysozyme. *J. Mol. Biol.* **224**, 1143-1154.
5. Lazar GA, Desjarlais JR, Handel TM. 1997. De novo design of the hydrophobic core of ubiquitin. *Prot. Sci.* **6**, 1167-1178.
6. Dahiyat BI, Gordon DB, Mayo SL. 1997. Automated design of the surface positions of protein helices. *Prot. Sci.* **6**, 1333-1337.
7. Munoz V, Serrano L. 1994. Intrinsic secondary structure propensities of the amino acids, using statistical phi-psi matrices - comparison with experimental scales. *Prot: Struct. Func. Gen.* **20**, 301-311.
8. Smith CK, Withka JM, Regan L. 1994. A thermodynamic scale for the b-sheet-forming tendencies of amino acids. *Biochemistry* **33**, 5510-5517.
9. Minor DL, Kim PS. 1994. Context is a major determinant of β-sheet propensity. *Nature* **371**, 264-267.

10. Kim CA, Berg JM. 1993. Thermodynamic β-sheet propensities measured using a zinc-finger host peptide. *Nature* **362**, 267-270.

11. Minor DL, Kim PS. 1994. Measurement of the β-sheet-forming propensities of amino acids. *Nature* **367**, 660 - 663.

12. Gronenborn AM, Filpula DR, Essig NZ, Achari A, Whilow M, Wingfield PT, Clore GM. 1991. A novel, highly stable fold of the immunoglobulin binding domain of streptococcal protein G. *Science* **253**, 65-661.

13. Dunbrack RL and Karplus M. 1993. Backbone-dependent rotamer library for proteins – application to side-chain prediction. *J. Mol. Biol.* **230**, 543-574.

14. Desmet J, De Maeyer M, Hazes B, Lasters I. 1992. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature* **356**, 539-542.

15. Mayo SL., Olafson BD, and Goddard WA. 1990. DREIDING – A generic force-field for molecular simulations. *J. Phys. Chem.* **94**, 8897-8909.

16. Hemsley A, Arnheim N, Toney MD, Cortopassi G, Galas DJ. 1989. A simple method for site-directed mutagenesis using the polymerase chain reaction. *Nucl. Acid. Res.* **17**, 6545-6551.

17. Johnson BH, Hecht MH. 1994. Recombinant proteins can be isolated from E. coli cells by repeated cycles of freezing and thawing. *Bio/Technology* **12**, 1357-1360.

18. Goldman A. 1995. How to make my blood boil. *Structure* **3,** (12) 1277-1279

19. Van den Burg B, Dijkstra BW, Vriend G, Van der Vinne B, Venema G, Eijsink VGH. 1994. Protein stabilization by hydrophobic interactions at the surface. *Eur J Biochem.* **220**, 981-985.

20. Street AG, Mayo SL. Penalizing hydrophobic exposure as a model for context effects on β-sheet stability. Submitted

21. Street AG, Datta D, Gordon DB, Mayo SL. 2000. Designing protein β-sheet surfaces by Z-score optimization. *Phys. Rev. Lett.* In press.

22. Mendez J, Baptista AM, Carrondo MA, Soares CM. 1999. Improved modeling of side-chains in proteins with rotamer-based methods: A flexible rotamer model. *Prot. Sruct. Func. Gen.* **37,** (4) 530-543

Table I. Summary of designed sequences.

		Sequence position											
Scale	Protein	*4*	*6*	*8*	*13*	*15*	*17*	*42*	*44*	*46*	*51*	*53*	*55*
0.0 - 1.1	s12_0.0	K+	Q	K+	E-	E-	Q	D-	Q	K+	T	E-	R+
1.1-1.2	s12_1.1	K+	T	K+	E-	Q	Q	D-	Q	K+	T	E-	R+
1.3	s12_1.3	T	T	K+	E-	Q	T	D-	Q	K+	T	E-	R+
1.4-1.8	s12_1.4	T	T	K+	E-	Q	T	D-	Q	D-	T	T	R+
1.9	s12_1.9	T	T	K+	E-	Q	T	D-	T	D-	T	T	R+
2.0-4.0	s12_2.0	T	T	K+	E-	T	T	D-	T	D-	T	T	R+
	Wild Type	K+	I	N	K+	E-	T	E-	T	D-	T	T	T

Protein	Charge	#Thr	#Bonds	Tm [°C]
s12_0.0	0	1	7	41
s12_1.1	+1	2	7	36
s12_1.3	0	4	6	29
s12_1.4	-1	5	6	61
s12_1.9	-1	6	5	-
s12_2.0	-1	7	4	67
Wild Type	-1	5	5	86

Figure III-1: Propensity pseudoenergies, as calculated by Munoz & Serrano. Proline (2.61 kcal/mol) is omitted for clarity.

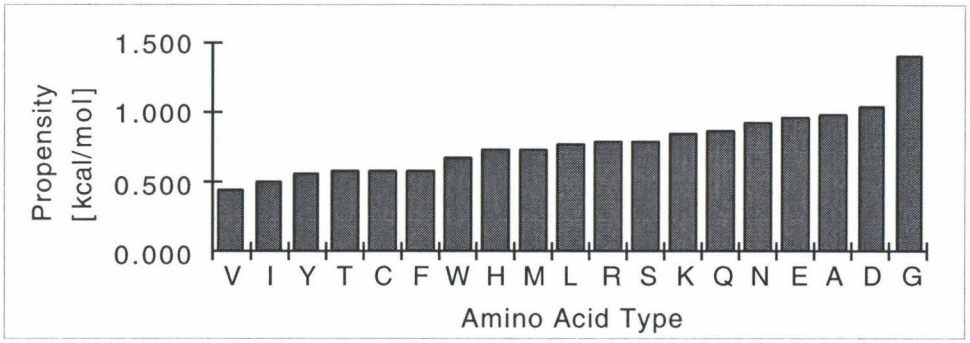

Propensity
[kcal/mol]
1.500
1.000
0.500
0.000
V I Y T C F W H M L R S K Q N E A D G
Amino Acid Type

Figure III-2: Illustration of how penalty energies based on the propensities in figure III-1 are scaled by N_β.

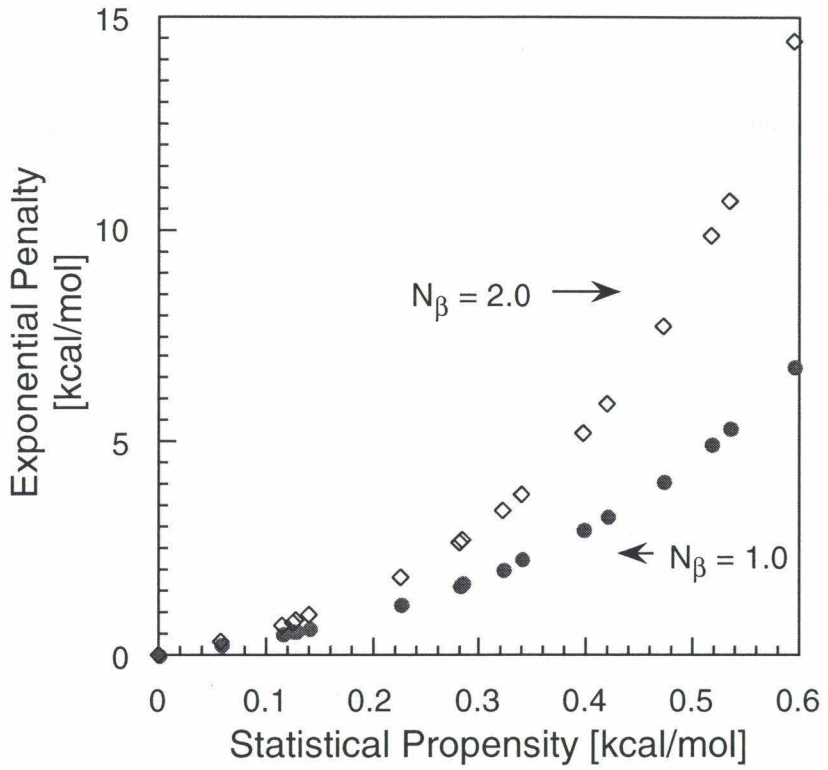

15
10
5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
Exponential Penalty [kcal/mol]
Statistical Propensity [kcal/mol]
N_β = 2.0
N_β = 1.0

Figure III-3: Schematic diagrams of predicted interactions for designed sequences. The wild-type interaction schematic is shown for reference. The interactions observed in these predicted structures were used to determine the values for Table 1.

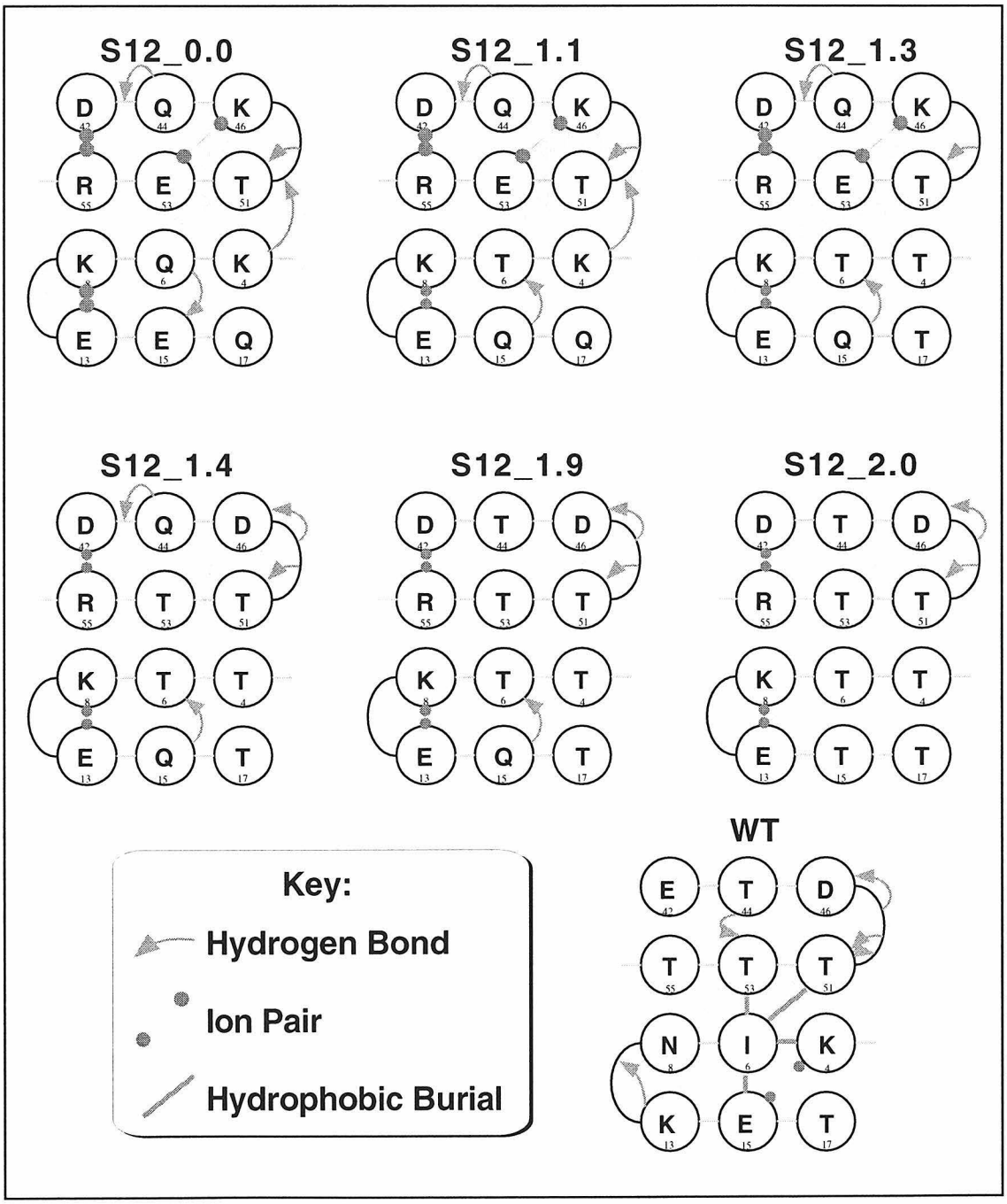

S12_0.0
D 42
Q 44
K 46
R 55
E 53
T 51
K 8
Q 6
K 4
E 13
E 15
Q 17
S12_1.1
D 42
Q 44
K 46
R 55
E 53
T 51
K 8
T 6
K 4
E 13
Q 15
Q 17
S12_1.3
D 42
Q 44
K 46
R 55
E 53
T 51
K 8
T 6
T 4
E 13
Q 15
T 17
S12_1.4
D 42
Q 44
D 46
R 55
T 53
T 51
K 8
T 6
T 4
E 13
Q 15
T 17
S12_1.9
D 42
T 44
D 46
R 55
T 53
T 51
K 8
T 6
T 4
E 13
Q 15
T 17
S12_2.0
D 42
T 44
D 46
R 55
T 53
T 51
K 8
T 6
T 4
E 13
T 15
T 17
Key:
Hydrogen Bond
Ion Pair
Hydrophobic Burial
WT
E 42
T 44
D 46
T 55
T 53
T 51
N 8
I 6
K 4
K 13
E 15
T 17

Figure III-4: Temperature Melts: Normalized CD melt data for designed β-surface sequences. The ellipticities were monitored at 218 nm, and Tm values were calculated from the maxima of the derivatives.

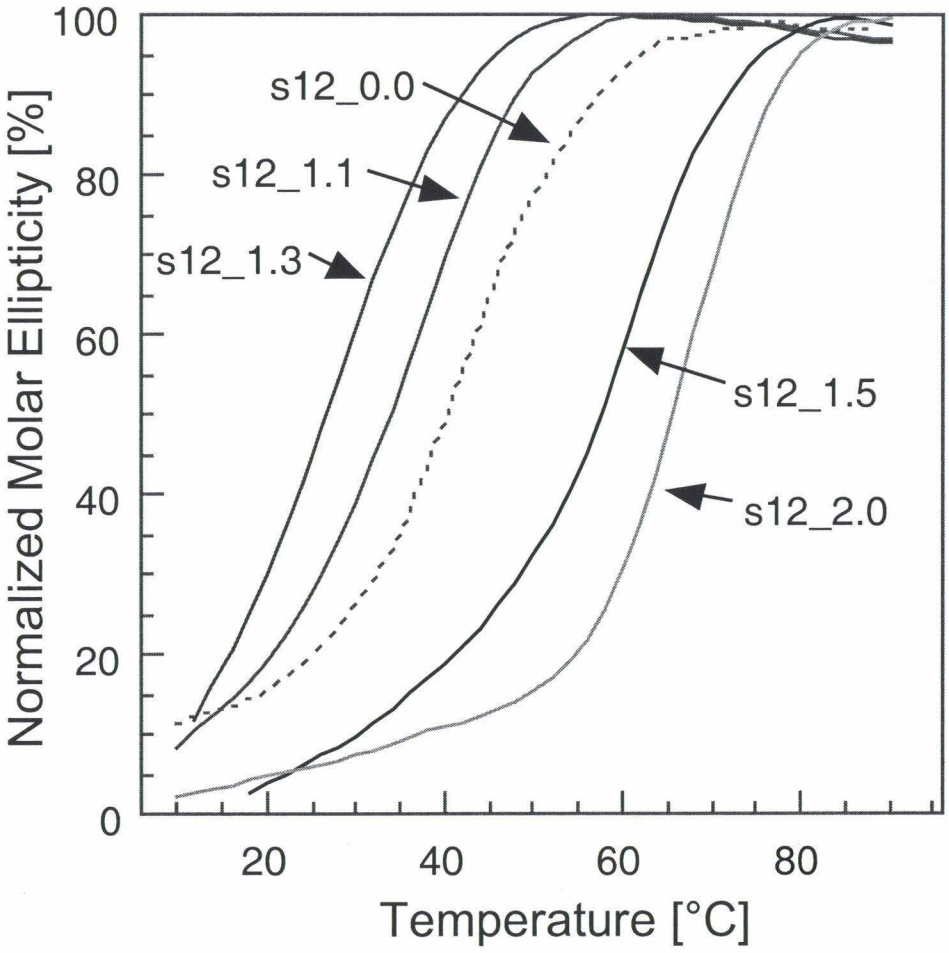
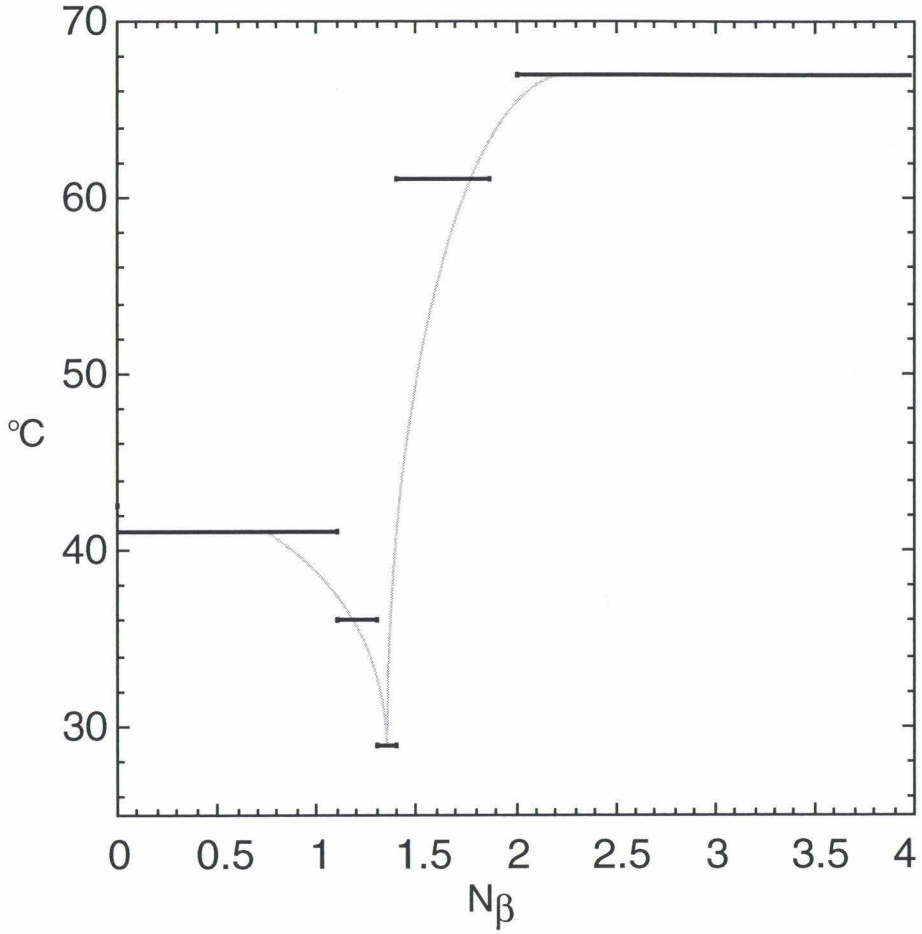

100
80
60
40
20
0
Normalized Molar Ellipticity [%]
s12_0.0
s12_1.1
s12_1.3
s12_1.5
s12_2.0
20
40
60
80
Temperature [°C]

Figure III-5: Propensity Scale vs. Melting Temperature: Lines span intervals of N_β for which the calculation produced the same sequences. Curves are drawn to suggest the trends of increasing number of high-propensity residues (right) and increasing number of inter-residue electrostatic interactions (left).


70
60
50
40
30
°C
0
0.5
1
1.5
2
2.5
3
3.5
4
Nβ

Chapter IV

Radical Performance Enhancements for Algorithms Based on the Dead-End Elimination Theorem

The text of this chapter is adapted from a published manuscript that was coauthored with Professor Stephen L. Mayo.

D. B. Gordon and S. L. Mayo; *J. Comp. Chem.* 1998. 19 (13) 1505-1514

Abstract

Recent advances in protein design have demonstrated the effectiveness of algorithms based on the Dead-End Elimination theorem. The algorithms solve the combinatorial problem of finding the optimal placement of side chains for a set of backbone coordinates. Though powerful tools, these algorithms have severe limitations when the number of rotamers is large. This is due to the high-order time-dependence of the aspect of the calculation that deals with rotamer doubles. We present three independent algorithmic enhancements that significantly increase the speed of the doubles computation. These methods work by using quantities that are inexpensive to compute to forecast which expensive calculations are worthwhile. One of the methods, the comparison of extrema, is derived from theory, and the remaining two, the magic bullet and the q_{rs} and q_{uv} metrics, are based on empirical observation of the distribution of energies. When used together, these methods effect an overall speed improvement of as much as a factor of 47, and for the doubles aspect of the calculation, a factor of 95. Together, these enhancements extend the envelope of inverse-folding to larger proteins by making formerly intractable calculations attainable in reasonable computer time.

Introduction

The accurate placement of side chains on a specified main chain template is of central importance to protein design and protein homology modeling. This placement is typically simplified through discretization of the conformational freedom of side chains into statistically significant representative conformations called rotamers [1,2]. Nevertheless, the sheer number of rotameric combinations makes exhaustive searches of all arrangements computationally intractable. The Dead-End Elimination (DEE) theorem proposed by Desmet et al. [3] solves this problem by providing an effective means of pruning rotamers from the available combinatorial space.

Several enhancements have been proposed since the DEE theorem was first described. Fuzzy-End Elimination [4] and improved forms of the elimination criteria [5] extend the utility of the theorem to more difficult problems. De Maeyer et al. [6] have demonstrated that the calculation speed can be increased by simultaneously implementing an energy threshold and a more detailed rotamer library. Together, these enhancements have enabled homology calculations for proteins as large as 250 residues.

These techniques, in conjunction with systematically derived energy expressions, have been used to perform inverse protein folding, that is, protein design. The hydrophobic cores of coiled-coil [7] and $\alpha+\beta$ proteins [8] have been successfully redesigned, as have α-helical surfaces [9]. Recently, Dahiyat and Mayo [10] have employed the DEE theorem in the complete redesign of an entire 28-residue motif.

Design calculations are significantly more computationally intensive than homology calculations on proteins of the same size. This is due to a high-order time dependence on the number of allowed rotamers per residue position. Design calculations

suffer because rotamers from several different amino acids are possible at each position. We present three speed enhancements that make much larger design calculations attainable in reasonable computer time.

Background

The strength of the DEE theorem is that it can determine that a particular rotamer cannot exist in the global minimum energy conformation (GMEC) without any prior knowledge of the GMEC. A rotamer determined to be incompatible with the minimum energy conformation is termed "dead-ending," and is eliminated from further consideration. The GMEC is then attained through iterative elimination of dead-ending rotamers until only a single rotamer remains at each residue position.

To eliminate a rotamer, one must show that there exists another rotamer that contributes less energy to the GMEC than the candidate rotamer. This is accomplished by finding a rotamer that is lower in energy than the candidate in all possible configurations of the system. The DEE criterion proposed by Desmet et al. [3] confirms the lower energy for all configurations by checking if, for some residue position, i, the minimum energy of the candidate rotamer to be eliminated, i_r, is greater than the maximum energy of another rotamer, i_t:

$$E(i_r) + \sum_{j, j \neq i} \min_s E(i_r j_s) > E(i_t) + \sum_{j, j \neq i} \max_s E(i_t j_s) \qquad (1)$$

The quantity $E(i_r)$ is the interaction energy of the rotamer i_r with the template. The energy of interaction between two rotamers i_r and j_s is denoted $E(i_r j_s)$. Thus the minimum

and maximum energies for all configurations are expressed on the left and right sides of the criterion, respectively.

There can be cases, however, in which the energy profile of a candidate rotamer may be higher than a reference in all conformations, but its minimum may be lower in energy than the maximum of the other rotamer. Although the candidate rotamer should be eliminated, the pair will be overlooked by the above elimination criterion. To treat this case, as well as higher order cases, Goldstein [5] proposed a form of the criterion of arbitrary order. The zeroth order form refers to the criterion proposed by Desmet et al. Goldstein describes a more sensitive, first-order form of the criterion that also detects the special case described above.

$$E(i_r) - E(i_t) + \sum_{j,j\neq i} \min_s \left[E(i_r j_s) - E(i_t j_s)\right] > 0 \qquad (2)$$

This criterion checks that the energy profiles of two rotamers do not cross by verifying that the minimum energy difference upon substitution of any rotamer for the candidate rotamer is always greater than zero.

In practice, the calculation typically reaches a point at which no more rotamers can be eliminated by the above criterion. Lasters and Desmet [4] describe how the calculation can be continued by finding pairs of rotamers, called doubles, that cannot coexist in the GMEC. The variation is obtained by tailoring the zeroth-order criterion (1) to search for dead-ending pairs:

$$\varepsilon([i_r j_s]) + \sum_{k,k\neq j\neq i} \min_t \varepsilon([i_r j_s], k_t) > \varepsilon(i_u j_v) + \sum_{k,k\neq j\neq i} \max_s \varepsilon([i_u j_v], k_t) \qquad (3)$$

where

$$\varepsilon([i_r j_s]) = E(i_r) + E(j_s) + E(i_r j_s)$$

and

$$\varepsilon([i_r j_s], k_t) = E(i_r k_t) + E(j_s k_t).$$

The Goldstein elimination criterion can also be extended to doubles:

$$\varepsilon([i_r j_s]) - \varepsilon([i_u j_v]) + \sum_{k,k \neq j \neq i} \min_t \left[\varepsilon([i_r j_s], k_t) - \varepsilon([i_u j_v], k_t)\right] > 0 \qquad (4)$$

In contrast to singles eliminated by criterion (1), it is possible that one of the individual rotamers that constitutes a dead-ending pair may exist in the GMEC, so the rotamers cannot be eliminated. However, as a unit, dead-ending pairs can be excluded when evaluating the minima in criteria (1) and (2) in subsequent singles calculations, enabling the elimination of more rotamers. Additionally, dead-ending pairs may be eliminated upon residue unification [11] in which a "super-residue" is constructed from all possible rotamer pairs for two positions. The super-residue is treated as a single residue for the remainder of the calculation.

Calculation Speed

Dead-End Elimination calculations filter through enormous numbers of combinations of sequences with remarkable speed when there are few rotamers per residue position. However, calculations proceed more slowly as the number of rotamers increases. This is of primary concern in protein design applications, in which each position has rotamers from many amino acids, often totaling hundreds of rotamers. Additionally, super-residues formed through unification also contribute large numbers of rotamers. The calculation is slowed in part because more rotamers need to be eliminated.

More importantly, however, is that the time to execute each iteration is significantly lengthened, because of a fourth-order dependence on the number of rotamers per residue position.

To clarify this fourth-order dependence, it is convenient to define a comparison matrix. To exhaustively search for all dead-ending rotamers at a residue position i, it is necessary to compare every rotamer to every other rotamer available at i. In the comparison matrix, each column corresponds to a particular rotamer, i_r, as a candidate for elimination, and each row corresponds to one of the possible reference rotamers i_t. If there are n rotamers at position i, then an exhaustive search of $(n^2\text{-}n)$ matrix elements is necessary. Such a matrix is evaluated for each of the p positions that may be represented by i.

The computational bottlenecks, however, are the evaluation of the minimum on the left side and the maximum on the right side of the elimination criterion. The calculation of each extremum requires computation for n rotamers at each of the other residue positions j. For the zeroth-order criterion (1), the same extrema can be used repeatedly within each row or column, and therefore they need only be computed once. The calculation time therefore scales proportionally to the number of rotamers and positions, $n \times p$. However, when the first-order criterion (2) is invoked, the minimum operator is applied to the rotamer pair, and therefore must be repeated for each matrix element. Therefore, an exhaustive search using the Goldstein variation scales as $n^2 \times p$.

The problem is exacerbated when performing doubles elimination. An $i_t j_s$ pair submitted to evaluation by criterion (3) will have n^2 combinations, as will the $i_u j_v$ pair to

which it is compared. Thus the dimension of the comparison matrix is $n^2\times(n^2\text{-}1)$, and such a matrix is constructed for each of the possible $\frac{1}{2}\times p\times(p\text{-}1)$ *i-j* doubles. As with zeroth-order singles, it is only necessary to evaluate the extrema once for each row and column, and so the calculation scales as $n^2\times p^2$. However, when it becomes necessary to progress to criterion (4), a computationally expensive calculation must be performed for every matrix element. Therefore, first-order doubles iterations scale as $n^4\times p^2$.

Performance analysis of our implementation of the DEE algorithm shows that the computation of the first-order doubles criterion (4) dominates the overall calculation time. For example, a doubles calculation with 100 rotamers at each position requires the evaluation of 10^8 matrix elements. At a typical evaluation rate of 10^4 comparisons per second, the zeroth-order calculation will take p^2 seconds, but the computation of an entire matrix for first-order doubles will take p^2 hours.

Optimization

Minima and Maxima

The actual number of dead-ending pairs found when using the first-order doubles criterion is much smaller than the number of comparison matrix elements. The calculation could be made much faster if there were a way to predict which matrix elements were likely to be dead-ending, which would then be confirmed with the DEE criterion. Our approach is to prejudge matrix elements by utilizing the minima and maxima precalculated for the zeroth-order calculation. For convenience we define

$$\varepsilon_{\max}([i_r j_s]) = \varepsilon([i_r j_s]) + \sum_{k\neq i\neq j} \max_t \varepsilon([i_r j_s],k_t) \qquad (5)$$

$$\varepsilon_{\min}([i_r j_s]) = \varepsilon([i_r j_s]) + \sum_{k \neq i \neq j} \min_t \varepsilon([i_r j_s], k_t) \tag{6}$$

$$\varepsilon_{\max}([i_u j_v]) = \varepsilon([i_u j_v]) + \sum_{k \neq i \neq j} \max_t \varepsilon([i_u j_v], k_t) \tag{7}$$

$$\varepsilon_{\min}([i_u j_v]) = \varepsilon([i_u j_v]) + \sum_{k \neq i \neq j} \min_t \varepsilon([i_u j_v], k_t) \tag{8}$$

These quantities are illustrated on energy profiles in figure IV-1. As previously stated, the calculation of these extrema scales as n^2, rather than as n^4, because the values are the same for an entire row or column of the matrix.

Magic Bullet

It is important to emphasize that it is not necessary to discover all possible dead-ending-pairs in the matrix. Although more would be preferable, it is only necessary to find a sufficient number to enable successful elimination in the next singles iteration. It is therefore reasonable to sacrifice the discovery of some pairs to gain calculation speed.

Inspection of the energy distributions in sample matrices has revealed that an $i_u j_v$ pair that dead-end eliminates a particular $i_r j_s$ pair can also eliminate other $i_r j_s$ pairs. In fact, there are often a few $i_u j_v$ pairs, which we call "magic bullets," that eliminate a significant number of pairs. We have found that one of the most potent magic bullets is the pair for which the maximum interaction energy, $\varepsilon_{max}([i_u j_v])$, is least. We refer to this pair as $[i_u j_v]_{mb}$.

Our first speed enhancement is to evaluate the first-order doubles calculation for only the matrix elements in the row corresponding to the $[i_u j_v]_{mb}$ pair. The discovery of $[i_u j_v]_{mb}$ is an n^2 calculation, and the application of criterion (4) to the single row of the

matrix corresponding this rotamer pair is another n^2 calculation, so the calculation time is small in comparison to a full first-order doubles calculation. In practice, this calculation produces a large number of dead-ending pairs, often enough to proceed to the next iteration of singles elimination without any further searching of the doubles matrix.

The magic bullet first-order calculation will also discover all dead-ending pairs that would be discovered by the zeroth-order calculation, thereby making it unnecessary. This stems from the fact that $\varepsilon_{max}([i_u j_v]_{mb})$ must be less than or equal to any $\varepsilon_{max}([i_u j_v])$ that would successfully eliminate a pair by the zeroth-order criterion.

Comparison of Extrema

When the magic bullet doubles calculation fails to produce any more dead-ending pairs, it is necessary to evaluate the full doubles matrix. However, we observe that the remaining doubles that satisfy the first-order doubles criterion are sparse on the matrix. Therefore, many matrix elements must be searched to find a relatively small number of dead-ending pairs. The search odds can be improved, however, by using the minima and maxima precomputed earlier to isolate regions of the matrix for which the probability of finding a dead-ending pair is greater.

We employ a comparison of extrema to effectively reduce the matrix by a factor of four. Matrix elements that satisfy either of the following criteria are skipped:

$$\varepsilon_{min}([i_r j_s]) < \varepsilon_{min}([i_u j_v]) \tag{9}$$

or

$$\varepsilon_{max}([i_r j_s]) < \varepsilon_{max}([i_u j_v]) \tag{10}$$

Figure IV-2 illustrates schematically that when either of these conditions are met, the energy profiles necessarily cross (see appendix for proof). We can therefore be certain that the corresponding matrix element will not be dead-ending.

Because the matrix is symmetrical, half of its elements will satisfy the first inequality (9), and half of those remaining will satisfy the other inequality (10). These three quarters of the matrix need not be subjected to the evaluation of criterion (4), resulting in a theoretical speed enhancement of a factor of four.

Proof of Comparison of Extrema

To simplify the presentation, we show the proof for a singles calculation. Consider a pair of rotamers i_r and i_t for which we observed that

$$E_{min}(i_r) < E_{min}(i_t)$$

which means

$$E(i_r) + \sum_j \min_s E(i_r j_s) < E(i_t) + \sum_j \min_s E(i_t j_s)$$

Rearranging we obtain

$$E(i_r) - E(i_t) + \sum_j \min_s E(i_r j_s) - \sum_j \min_s E(i_t j_s) < 0$$

Let m be the selection of rotamer s for each position j that minimizes $E(i_r j_s)$. By definition, then,

$$\sum_j E(i_r j_m) = \sum_j \min_s E(i_r j_s)$$

Now, since

$$\sum_j E(i_t j_m) \geq \sum_j \min_s E(i_t j_s)$$

we have

$$E(i_r) - E(i_t) + \sum_j E(i_r j_m) - \sum_j E(i_t j_m) \leq 0$$

Because *m* is the same in both pairwise energy expressions, we may write

$$E(i_r) - E(i_t) + \sum_j \left[E(i_r j_m) - E(i_t j_m)\right] \leq 0$$

This shows that there must exist a configuration, *m,* for which the energy difference is negative upon substitution of i_t for i_r. Since the minimum difference upon substitution must be less than or equal to any particular difference,

$$\sum_j \min_s \left[E(i_r j_s) - E(i_t j_s)\right] \leq \sum_j \left[E(i_r j_m) - E(i_t j_m)\right]$$

Substituting the minimum difference yields

$$E(i_r) - E(i_t) + \sum_j \min_s \left[E(i_r j_s) - E(i_t j_s)\right] \leq 0$$

Thus, given that the initial comparison of minima is satisfied, the minimum difference must be less than zero. Therefore, i_t cannot eliminate i_r by first-order elimination. By analogy, the same condition can be derived when the maximum energy of i_t exceeds the maximum of i_r. The proofs are analogous for doubles calculation, confirming the conditions described above.

"q" metrics

Our last enhancement refines the search of the remaining quarter of the matrix. We accomplish this by constructing a metric from the precomputed extrema to detect those matrix elements likely to result in a dead-ending pair.

A metric was found through analysis of matrices from different sample optimizations. We searched for combinations of the extrema that predicted the likelihood that a matrix element would produce a dead-ending pair. Interval sizes (see figure IV-1) for each pair were computed from differences of the extrema. The size of the overlap of the $i_r j_s$ and $i_u j_v$ intervals were also computed, as well as the difference between the minima and the difference between the maxima. Combinations of these quantities as well as the lone extrema were tested for their ability to produce the occurrence of dead-ending pairs. Also, because some of the maxima were very large, the quantities were also compared logarithmically.

Most of the combinations exhibited the ability to predict dead-ending matrix elements to varying degrees. The best metrics were the fractional interval overlap with respect to each pair. We refer to these quotients as q_{rs} and q_{uv}.

$$q_{rs} = \frac{\text{intervaloverlap}}{\text{interval}([i_r j_s])} = \frac{\varepsilon_{\max}([i_u j_v]) - \varepsilon_{\min}([i_r j_s])}{\varepsilon_{\max}([i_r j_s]) - \varepsilon_{\min}([i_r j_s])} \tag{11}$$

$$q_{uv} = \frac{\text{intervaloverlap}}{\text{interval}([i_u j_v])} = \frac{\varepsilon_{\max}([i_u j_v]) - \varepsilon_{\min}([i_r j_s])}{\varepsilon_{\max}([i_u j_v]) - \varepsilon_{\min}([i_u j_v])} \tag{12}$$

These metrics were selected because they had values for which the ratio of the total occurrence to the occurrence of dead-ending matrix elements was lower than any other metric. For example, we observe that there are very few (~2%) matrix elements for which $q_{rs} > 0.98$, yet these elements produce as many as 40% of all of the dead-ending pairs. We only apply the first-order doubles criterion to those doubles for which $q_{rs} > 0.98$ and $q_{uv} < 0.99$.

The sample data analyses predict that by using these two metrics, we may find as many as half of the dead-ending elements by evaluating only two to five percent of the reduced matrix. However, we do not expect to observe the full theoretical enhancement because the analysis does not account for redundant eliminations of a pair.

Method

Energy Expression

The energy expression consists of van-der-Waals, electrostatic, and solvation terms. For van-der-Waals, a Lennard-Jones 6-12 potential is used, with radii scaled [8] by a factor of 0.9. A distance-dependent electrostatic term and a hybridization-dependent hydrogen-bonding term were used [9]. Solvation effects are approximated from hydrophobic surface area burial [7,10]. Atom radii and hydrogen-bond well depths are based on the DREIDING force-field [12].

Algorithm

The basic algorithm was implemented as described in the background section of this paper. Residue unification [11] was performed when first-order doubles failed to facilitate subsequent singles iterations, by clustering the pair of positions that produced the largest fraction of dead-ending pairs. Rotamers are selected from a backbone dependent library [13].

The three speed enhancements were added sequentially. First, calculations were performed using the original algorithm. Next, magic bullet doubles were substituted for the zeroth-order doubles calculation. Then a filter implementing the comparison of

extrema was added to the first-order doubles calculation. Last, the q_{rs} and q_{uv} metrics were added as a final filter to the first-order doubles calculation.

For each calculation, the total CPU time was recorded, as well as the portion of that time spent performing first-order doubles. The time required for the initial first-order doubles was also measured. All calculations were performed on a single R10000 CPU of a Silicon Graphics Origin 2000 server.

Benchmark Cases

It was necessary to test the generality of the speed enhancements, since their viability is, in part, dependent on the distribution of energies. Therefore, three sequence optimization problems representative of different protein structural classes were selected. To test α-helical surfaces, the coiled-coil GCN4-p1 [9,14] was used. The twelve residues occupying **b**, **c**, or **f** locations in the heptad repeat were optimized allowing each position to have the identity of any of the hydrophilic amino acids (D, E, N, Q, K, R, S, T, A, and H). There were 8.5×10^{26} rotameric combinations.

The structure of the β1 domain [15] of streptococcal protein G was used to test the applicability of the enhancements to protein cores and β-sheet surfaces. For the former, 13 positions in the core and at the boundary (3, 5, 7, 26, 30, 33, 34, 37, 43, 50, 52, 54, 56) were optimized from the 2.4×10^{23} combinations of hydrophobic rotamers (A, F, I, L, M, V, W, Y). For the β-sheet surface, 12 positions (4, 6, 8, 13, 15, 17, 42, 44, 46, 51, 53, 55) were optimized from the 1.8×10^{26} combinations of hydrophilic rotamers.

Results

The calculation times for the three benchmark cases are shown in Table I. The enhancements collectively increase the calculation speed by more than an order of magnitude. In some cases, the overall speed increase is nearly a factor of 50, and the speed enhancement for first-order doubles calculations is a factor of 95. All algorithms produce the same solutions for each optimization problem.

The evaluation times of the initial first-order doubles calculations are used as predictors of the speed enhancement for large calculations. It is not feasible to directly measure the speed enhancement for very large problems, due to the prohibitive calculation times for their references. Additionally, the overall enhancement is increased for calculations of large size, due to the larger fraction of the calculation dedicated to the evaluation of large doubles matrices. We therefore focus analysis on the calculation times of the earliest encountered large doubles matrix, though the trends are exhibited by the other performance measures as well.

Similar enhancements were observed for all three structural classes. The fluctuations in time improvement are apparently related to the overall difficulty of the optimizations. Harder calculations, such as those involving only the weakly interacting surface residues of β-sheets, derive the greatest enhancement.

The employment of the magic bullet imparts a speed enhancement factor of 1.3 to 2.9, depending on the nature of the optimization problem. As desired, the enhancement enables the calculation to progress through several additional iterations before requiring the invocation of a full first-order doubles round. The size of the problem is therefore reduced for subsequent expensive doubles calculations. (See Table II.)

The observed benefit of the comparison of extrema exceeds the theoretical enhancement for all the test cases. This is a by-product of an implementation detail that prevents redundant eliminations. It is unnecessary to search remaining $i_u j_v$ pairs after one is found that eliminates a particular $i_r j_s$, so calculations for $i_r j_s$ pairs that are eliminated require less computation time than those that are not. The comparison of extrema filter reduces the relative number of $i_r j_s$ pairs that require comparison against all $i_u j_v$ pairs, thereby further speeding the calculation.

Last, the combination of the metrics q_{rs} and q_{uv} is observed to work well for the different cases, increasing the speed of the preliminary doubles calculation by an additional factor of 5 to 8. Coupled with the similarity of trends observed in the initial matrix analysis, we conclude that the selected metrics, q_{rs} and q_{uv}, are effective for all structural classes.

Conclusions

We have demonstrated the effectiveness of three enhancements for algorithms based on the Dead-End Elimination theorem. When used in concert, these techniques reduce the calculation time of the slowest parts of the algorithm by nearly two orders of magnitude in some cases. We observe that all the techniques are effective for optimizations of different protein structural classes, and that the speed enhancements increase with the difficulty of the problem.

The increase in computational speed has dramatic consequences. Previously unattainable calculations for large protein systems are now tractable in reasonable computer time.

Moreover, the evaluation of a large, well-defined matrix lends itself to easy computational parallelization. We have coupled these enhancements with parallelization of the doubles matrix on a 32 CPU Silicon Graphics Origin 2000, and have observed that total calculation times scale nearly ideally with the number of processors used (see figure IV-3). This coupling has enabled us to perform calculations in one day that previously would have taken years.

The successes of the magic-bullet and metric methods suggest that there is fertile ground in the area of optimization based on empirical observation. More sophisticated metrics may yet exist to better predict which first-order doubles calculations are worthwhile.

Acknowledgements

We wish to thank A. G. Street for helpful discussions and L. S. Gordon for assistance with the mathematical proofs. This work was supported by the Howard Hughes Medical Institute (S.L.M.), grant GM 07616C-19 from the National Institutes of Health (D.B.G), the Rita Allen Foundation, the Chandler Family Trust, the Booth Ferris Foundation, the David and Lucile Packard Foundation, the Searle Scholars Program, and the Chicago Community Trust.

References

1. Janin J, Wodak S, Levitt M, Maigret D. 1978. Conformation of amino acid side-chains in proteins. *J. Mol. Biol.* **125**, 357-386.
2. Ponder J, Richards F. 1987. Tertiary templates for proteins. *J. Mol. Biol.* **193**, 775-791.
3. Desmet J, De Maeyer M, Hazes B, Lasters I. 1992. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature* **356**, 539-842.
4. Lasters I, Desmet J. 1993. The fuzzy-end elimination theorem: correctly implementing the side-chain placement algorithm based on the dead-end elimination theorem. *Protein Engineering* **6,** (7): 717-712.
5. Goldstein RF. 1994. Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophys. J.* **66**, 1335-1340.
6. De Maeyer M, Desmet J, Lasters I. 1997. All in One: A highly detailed rotamer library improves both accuracy and speed in the modeling of sidechains by dead-end elimination. *Fold. Des.* **2**, 53-66.
7. Dahiyat B, Mayo S. 1996. Protein design automation. *Protein Science.* **5**, 895-903.
8. Dahiyat B, Mayo S. 1997. Probing the role of packing specificity in protein design. *Proc. Natl. Acad. Sci. USA* **94**, 10172-10177.
9. Dahiyat B, Gordon DB, Mayo S. 1997. Automated design of the surface positions of protein helices. *Protein Science.* **6**, 1333-1337.
10. Dahiyat B, Mayo S. 1997. De novo protein design: Fully automated sequence selection. *Science.* **278**, 82-87.

11. Desmet J, De Maeyer M, Lasters I. 1994. Chapter 10: The "Dead-End Elimination" Theorem: A New Approach to the Side-Chain Packing Problem. The Protein Folding Problem and Tertiary Structure Prediction. Birkhäuser. Boston. pp. 307-337.

12. Mayo S, Olafson BD, Goddard WA. 1990. DREIDING: A generic force field for molecular simulations. *J. Phys. Chem.* **94**, 8897-8909.

13. Dunbrack RL, Karplus M. 1993. Backbone-dependent rotamer library for proteins-An application to side-chain prediction. *J. Mol. Biol.* **230**, 543-574.

14. Gallagher T, Alexander P, Bryan P, Gilliland GL. 1994. 2 crystal-structures of the β1 immunoglobulin-binding domain of streptococcal protein-G and comparison with nmr. *Biochemistry.* **33**, 4721-4729.

15. O'Shea E, Klemm J, Kim P, Alber T. 1991. X-ray structure of the GCN4 leucine zipper, a two-Stranded, parallel coiled coil. *Science* **254**, 539-544.

Table I.

Observed speed enhancements for magic bullet, comparison of extrema, and q_{rs} and q_{uv} metrics on representative cases of three structural classes.

	Method	Total Optimization Time minutes (factor)		Total First-order Doubles Time minutes (factor)		Earliest First-order Doubles Time[5] minutes (factor)	
Case 1: Core and Boundary	Original[1]	192.6		190.6		24.9	
	MB[2]	165.5	(×1.2)	163.7	(×1.2)	11.2	(×2.2)
	MB + CoE[3]	31.3	(×6.2)	29.8	(×6.4)	2.4	(×10)
	MB + CoE + Q[4]	14.0	(×14)	11.5	(×17)	0.5	(×52)
Case 2: Helical Surface	Original	461.4		436.6		371.2	
	MB	206.5	(×2.2)	204.6	(×2.1)	152.9	(×2.4)
	MB + CoE	49.2	(×9.4)	47.3	(×9.2)	35.5	(×10)
	MB + CoE + Q	13.7	(×34)	11.4	(×38)	6.6	(×56)
Case 3: Beta-Sheet Surface	Original	868.6		866.1		712.7	
	MB	303.5	(×2.9)	300.8	(×2.9)	257.3	(×2.8)
	MB + CoE	71.2	(×12)	68.5	(×13)	59.1	(×12)
	MB + CoE + Q	18.4	(×47)	14.8	(×59)	7.5	(×95)

[1] The original algorithm uses the zeroth-order doubles criterion prior to evaluating the entire first-order doubles matrix.

[2] Magic bullet (MB) first-order doubles are substituted for zeroth-order doubles.

[3] The comparison of extrema (CoE) filter is employed during evaluation of the first-order doubles matrix.

[4] The metrics (Q) are used as additional filters during first-order doubles ($q_{rs} > 0.98$ and $q_{uv} < 0.99$).

[5] Execution time required for the earliest encountered iteration of first-order doubles.

Table II.

Effect of substituting magic bullet doubles on subsequent first-order doubles calculations.

	Number of combinations at beginning of optimization	Doubles method	Number of combinations remaining when evaluating the earliest first-order doubles matrix	Average number of rotamers per position during earliest first-order doubles matrix†	Theoretical speed enhancement‡ (ratio)^4	Observed speed enhancement
Case 1: Core and Boundary	2.4×10^{23}	Zeroth-order	9.9×10^{15}	36.2		
		magic bullet	5.9×10^{14}	30.4	2.0	2.2
Case 2: Helical Surface	8.5×10^{26}	Zeroth-order	6.4×10^{21}	81.2		
		magic bullet	5.0×10^{20}	70.0	1.8	2.4
Case 3: Beta-Sheet Surface	1.8×10^{26}	Zeroth-order	2.3×10^{21}	80.2		
		magic bullet	7.2×10^{19}	64.1	2.4	2.8

†Average number of rotamers calculated by dividing the total number rotamers by the number of residue positions in the optimization.

‡Because the evaluation time of the doubles matrix scales as (number of rotamers per postion)^4, one may approximate the theoretical improvement from the relative numbers of rotamers per residue position.

Figure IV-1: Schematic representation of the quantities defined in Eq. (5-8) that are used to construct speed enhancements. The minima and maxima are utilized directly to find the $[i_u j_v]_{mb}$ pair and for the comparison of extrema. The differences between the quantities, denoted with arrows, are used to construct the q_{rs} and q_{uv} metrics.

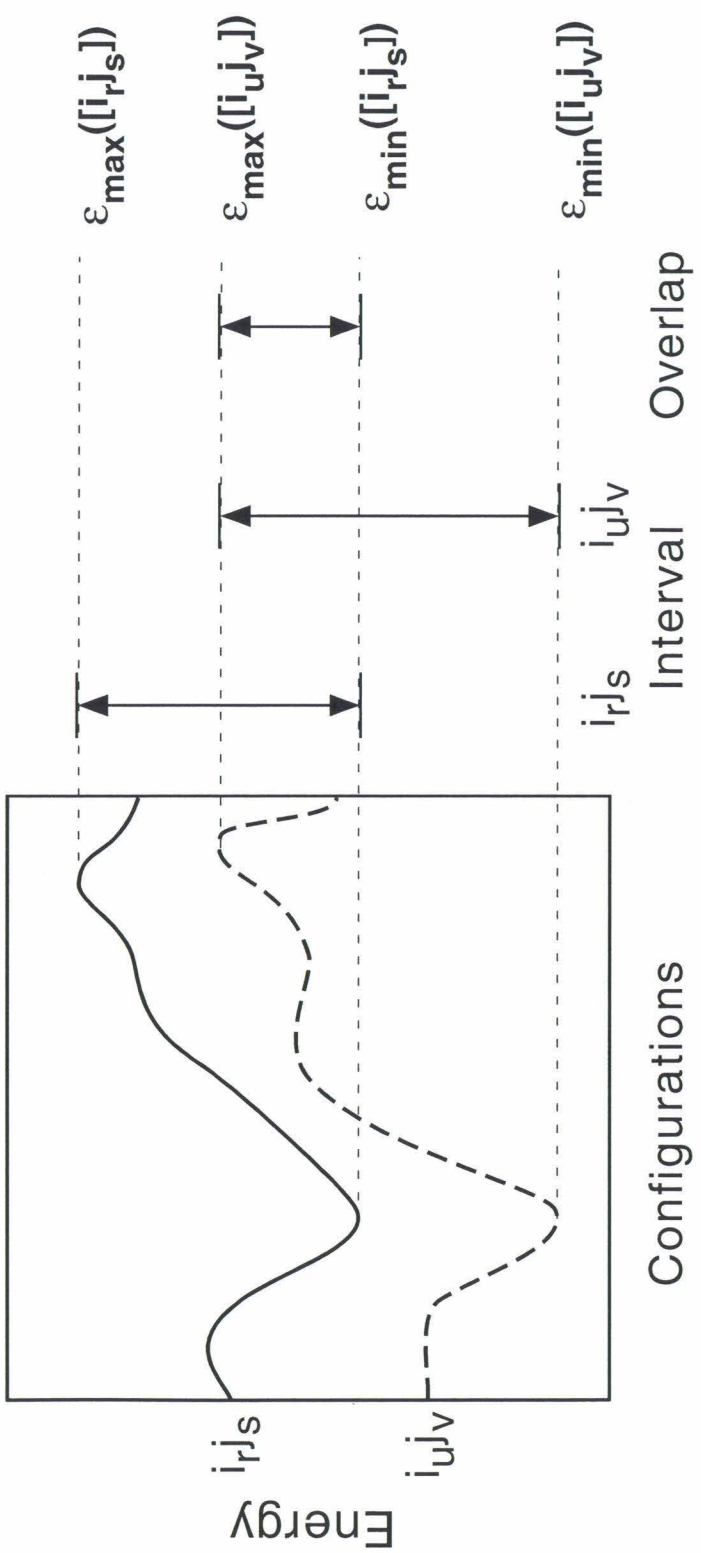
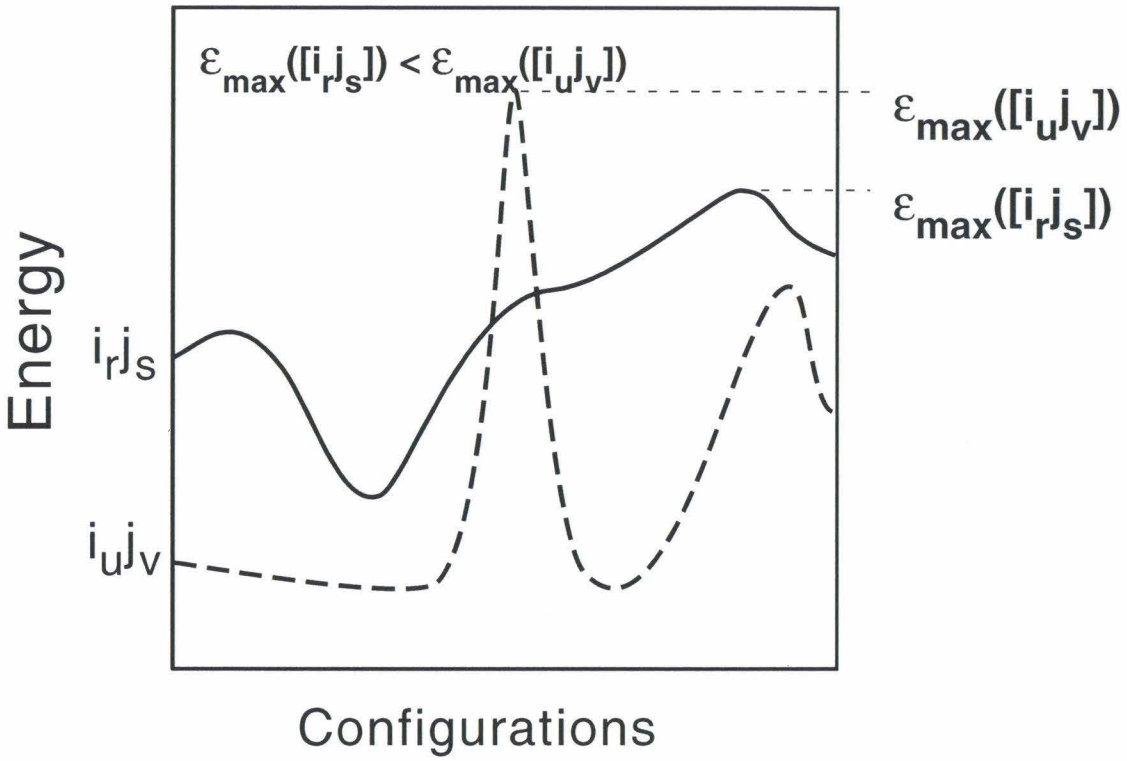

$\varepsilon_{max}([i_r j_s])$
$\varepsilon_{max}([i_u j_v])$
$\varepsilon_{min}([i_r j_s])$
$\varepsilon_{min}([i_u j_v])$
$i_r j_s$
$i_u j_v$
Interval
Overlap
Configurations
Energy
$i_r j_s$
$i_u j_v$

Figure IV-2: These graphs illustrate that the energy profiles of pairs must cross if they satisfy the comparison of extrema of inequalities (9) in figure IV-2a, and (10) in figure IV-2b. One may therefore determine beforehand that these pairs cannot be eliminated by the first-order criterion.

(a)


$\varepsilon_{max}([i_r j_s]) < \varepsilon_{max}([i_u j_v])$
$\varepsilon_{max}([i_u j_v])$
$\varepsilon_{max}([i_r j_s])$
Energy
$i_r j_s$
$i_u j_v$
Configurations

(b)

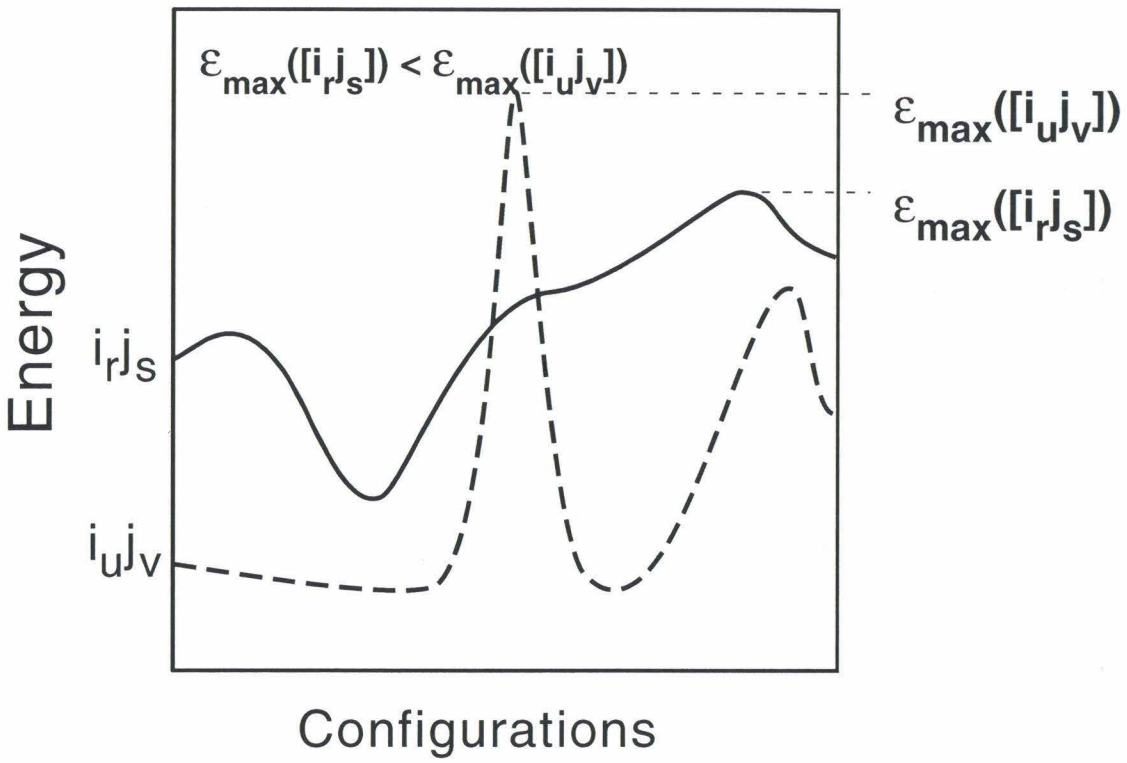
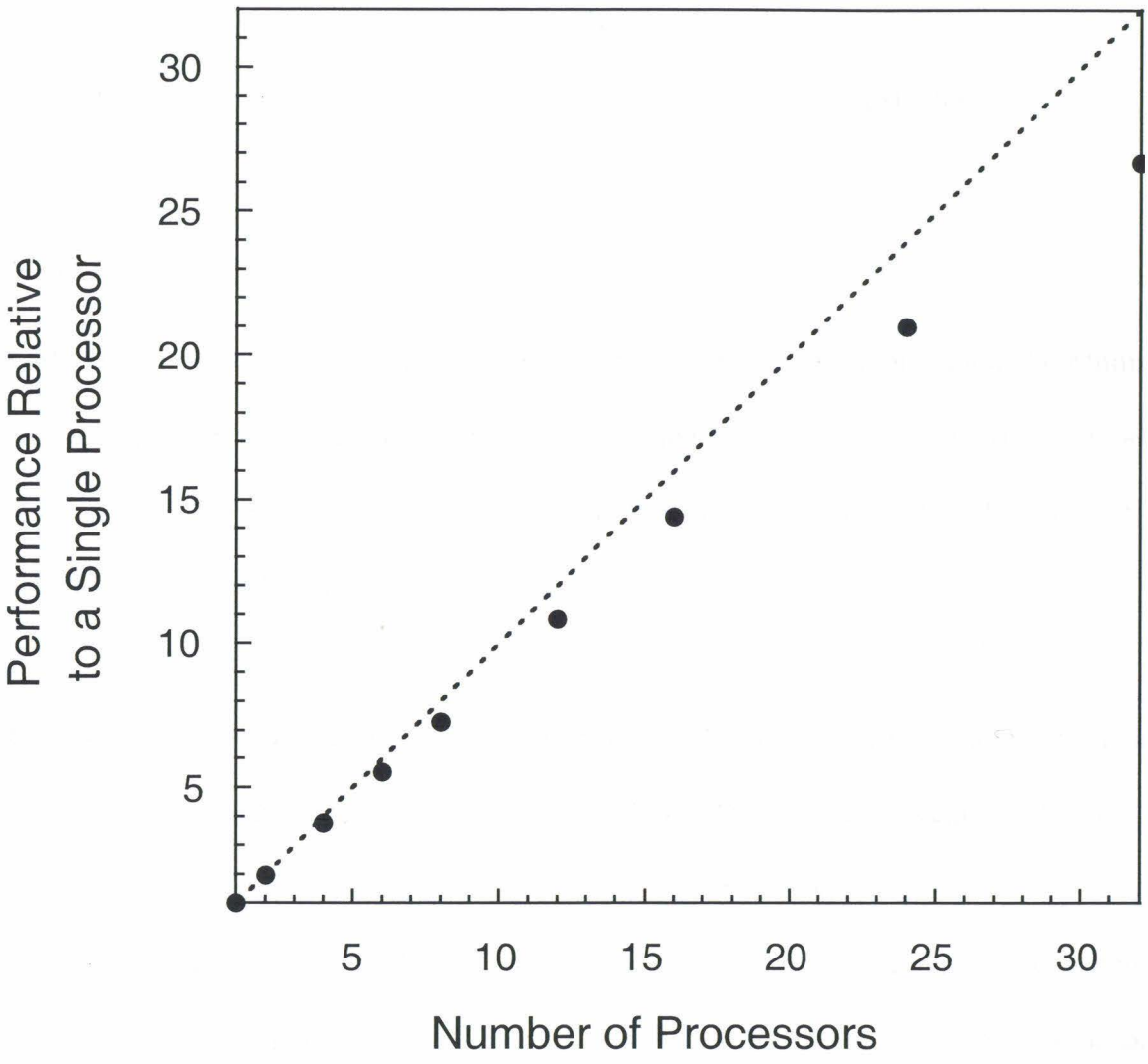

$\varepsilon_{max}([i_r j_s]) < \varepsilon_{max}([i_u j_v])$
$\varepsilon_{max}([i_u j_v])$
$\varepsilon_{max}([i_r j_s])$
Energy
$i_r j_s$
$i_u j_v$
Configurations

Figure IV-3: Increase in calculation speed from parallelization. Times were tabulated for a single first-order doubles iteration during the optimization of the 12 β-sheet surface positions of protien G, allowing all amino acid identities, except proline, at all positions. Speed factors are computed relative to the calculation time on a single CPU.


Performance Relative
to a Single Processor
30
25
20
15
10
5
5
10
15
20
25
30
Number of Processors

Chapter V

Branch and Terminate: A Combinatorial Optimization Algorithm for Protein Design

The text of this chapter is adapted from a published manuscript that was coauthored with Professor Stephen L. Mayo.

D. B. Gordon and S. L. Mayo; *Struture Fold. Des.* 1999. **7** (9) 1089-1098

Abstract

Background: Several deterministic and stochastic combinatorial optimization algorithms have been applied to computational protein design and homology modeling. However, as structural targets get larger, it has become necessary to find more powerful methods to address the increased combinatorial complexity.

Results: We present a new deterministic combinatorial search algorithm, called "Branch and Terminate," (B&T) derived from the Branch-and-Bound search method. The B&T approach is based on the construction of an efficient, but very restrictive bounding expression, which is used for the search of a combinatorial tree representing the protein system. The bounding expression is used both to determine the optimal organization of the tree and to perform a highly effective pruning procedure named "termination." For some calculations, the B&T method rivals the current deterministic standard, Dead-End Elimination (DEE), sometimes finding the solution up to 21 times faster. A more significant feature of B&T algorithm is that it can provide an efficient way to complete

the optimization of problems that have been partially reduced by a DEE algorithm.

Conclusions: The B&T algorithm is an effective optimization algorithm when used alone. Moreover, it can increase the problem size limit of amino acid side chain placement calculations, such as protein design, by completing DEE optimizations that reach a point at which the DEE criteria become inefficient. Together the two algorithms make it possible to find solutions to problems that are intractable by either algorithm alone.

Introduction

Significant advances in protein design [1,2] and protein side chain homology modeling [3] have arisen from the application of optimization algorithms and specialized potentials to the side chain placement problem. In these calculations, one searches for the set of side chain conformations that produce the global minimum energy conformation (GMEC) for the given protein backbone. The energies of side chain interactions are evaluated using empirically based energy potentials, and to reduce the complexity of the calculation, the set of possible side chain orientations is discretized into statistically representative conformations called rotamers [4,5].

The search for the optimal selection of side-chain rotamers for a specified protein fold is necessarily a combinatorial optimization problem; an exhaustive search through all combinations is intractable. As such, the problem has been approached by several different methods, including Monte Carlo [6,7] and simulated annealing [8], mean-field [9,10], and Dead-End Elimination (DEE) [11-14]. In particular, DEE methods have

emerged as powerful tools for more difficult protein design calculations, in which the optimal side chains are selected from rotamers of many different amino acids [1,2].

There are optimizations, however, for which DEE algorithms are not sufficient, due to either the nature of their energy distributions or their sheer size. For example, the optimization of long hydrophilic side chains on β-sheets is typically composed of large numbers of rotamers with interaction energies that are very small in magnitude. DEE is able to reduce the combinatorial size of the problem significantly at the outset, but soon after, elimination becomes inefficient, relying entirely on computationally expensive DEE doubles calculations [12, 14]. This behavior is also observed in the later stages of very large calculations, when after several rounds of unification [15] further eliminations become difficult and the number of super-rotamers at super-residue positions becomes very large. To complete such calculations, a technique consisting of exhaustive combinatorial build-up aided by DEE has been described [3]. However, since the effectiveness of the elimination criteria is poor in these cases, it is advantageous to construct a method that is not dependent on them.

To address these difficult optimization problems, we have developed an enhanced version of a Branch-and-Bound (B&B) algorithm [16] that we have dubbed "Branch-and-Terminate" (B&T). B&B algorithms comprise a sub-class of backtrack algorithms that utilize information about costs (or energies) of complete and partial solutions. Backtrack algorithms are commonly used in atomic-level simulations to construct self-avoiding chains, and they have been used in protein design to engineer metal binding sites into proteins [17].

B&B algorithms are commonly applied to theoretical combinatorial and

scheduling problems, and more recently to combinatorial problems of structural biology ranging from sequence alignment [18] and structural comparison [19], to macromolecular packing [20], ligand design [21], and recently, protein tertiary structure prediction [22]. Toward the study of protein side-chains, Samudrala and Moult [23] have described a graph-theoretic approach to the closely related problem of comparative modeling, in which they represent the search as a clique-finding problem, which they solve using a B&B algorithm. In addition, Leach and Lemon [24] have used a B&B algorithm (called "A*") to explore the conformational energy surface of protein side-chains.

It is straightforward to formulate the side chain optimization problem for direct optimization by a B&B algorithm. All that is necessary is to describe the problem as a search of a combinatorial tree where one searches for the single path through the branches that corresponds to the GMEC set of rotamers. The B&B algorithm is effective because it simultaneously prunes the tree while searching; each branch is tested with a quantitative bounding expression before being searched.

In implementing a B&B algorithm for side chain selection, we have incorporated some novel algorithmic techniques that increase the optimization speed dramatically. First, we describe a bounding function that maximizes the efficiency of pruning for problems in which the total energy can be decomposed into interactions between pairs of rotamers. We also describe a process we call "termination," in which we use the bounding function to deterministically remove rotamers at all amino acid positions, thereby reducing the overall size of the tree before searching. Termination is additionally effective when performed at every level of recursion of the search, sometimes increasing the overall speed of the optimization by an order of magnitude. Last, we demonstrate

how the energetic information produced by the termination process can be used to determine the optimal search order for the remainder of the tree. Because termination effectively replaces the usual bounding process, the resulting breadth-first algorithm is called "Branch-and-Terminate." We also describe a variation of the B&T method that can rapidly find approximate solutions close to the GMEC.

The description of the Branch-and-Terminate algorithm that follows is tailored for rotamer selection, but the algorithm is in fact generalizable to any combinatorial optimization problem in which all the interactions energies are pairwise and pre-computable. The bounding expression we describe is similarly general.

Although the B&T algorithm can be used by itself, greater benefit can often be obtained by using it in concert with a DEE algorithm. Together, the algorithms can solve optimization problems much more quickly than either can accomplish alone. This may make it possible to quickly find the GMEC for protein design problems that were previously insoluble by either algorithm.

Results

Branch and Bound

When a combinatorial tree is used to describe the side chain optimization problem, the root of the tree is placed at the top, and branches extend downward. Each level of depth of the tree corresponds to an amino acid position, and each node represents a particular rotamer choice at that position. (See figure V-1.) Thus a path that extends all the way from the tree root through all levels of branches to a leaf describes a complete rotamer sequence. The problem, then, is to search for the path corresponding to the

sequence with the lowest energy.

A partial path from the root describes a rotamer sequence that is incompletely specified. Alternatively, the path can be interpreted physically as specifying a unique composite rotamer, or "super-rotamer" that occupies a subset of the amino acid positions. Extending the path deeper into the tree corresponds to appending additional rotamers to the super-rotamer, which can be repeated until all positions are specified. According to this interpretation, a full search of the tree would entail the construction of all possible super-rotamers to completion.

It is often possible, however, to determine that a particular partially-specified super-rotamer is not part of the GMEC. In such a case, it is unnecessary to explore any combinations that would result from building up the super-rotamer further. Applied recursively, such observations prune sub-trees from nodes throughout the tree, thereby enabling an exhaustive search without complete enumeration of all possible super-rotamers.

The pruning determination is accomplished by comparing a lower energy bound for the partially-specified rotamer sequence to a known reference energy. Given a reference energy of any plausible sequence, it must be true that the energy of the GMEC is less than or equal to the energy of any plausible sequence.

$$E_{GMEC} \leq E_{reference} \tag{1}$$

One may therefore deduce that the global minimum does not contain a particular super-rotamer upon observing that the energy $E_{super,best}$ of the sequence resulting from optimal completion of the candidate super-rotamer is greater than the reference energy.

$$E_{super,\,best} > E_{reference} \tag{2}$$

Finding the optimal completing sequence, however, can be as difficult as the original problem, so we instead construct an expression for a lower energy bound, $E_{super,bound}$. The expression is constructed to compute an inexpensive lower energy bound based on the partially specified sequence, as well as on the rotamers that are available at the unspecified positions. By definition, the bound must satisfy the inequality,

$$E_{super,\,best} \geq E_{super,\,bound} \tag{3}$$

With this quantity in hand, we may prune any sub-tree for which we observe that the lower bound is greater than the reference energy.

$$E_{super,\,bound} > E_{reference} \tag{4}$$

This is the bounding criterion. The Branch-and-Bound algorithm consists of an exhaustive traversal of the combinatorial tree, applying this criterion to each node as it is encountered. Whenever the search produces a complete path with an energy lower than the current reference energy, the reference energy is updated. This way, the effectiveness of the bounding criterion is increased over the course of the optimization. Moreover, upon completion of the search, the reference energy is the global minimum energy. The corresponding sequence is also stored during each update, which produces the corresponding GMEC.

Bounding Expression

The successful implementation of a B&B type of algorithm depends largely on the

construction of the bounding expression. A bounding expression that is very stringent will produce lower bounds that are high in energy, and therefore will result in more sub-trees that can be pruned by the bounding criterion. The size of the resulting tree will be smaller than one pruned by a less stringent expression, and the search will be faster. It is therefore important to design the bounding expression to most fully utilize the sequence information available.

On the other hand, stringency is obtained at the cost of time. A maximally stringent bound might prune all sub-trees except for the one containing the global minimum, but it would take an impractical amount of time to compute. It is therefore also necessary to temper stringency with speed considerations in order to obtain a bounding expression that is properly balanced for efficient searching.

We describe the construction of such a bounding expression in the Materials and Methods section. Given a partially constructed super-rotamer and the available rotamers at the remaining positions, the approach is to utilize the corresponding energetic information as fully as possible while keeping the computational order of the bounding expression constant. The result is a novel, highly-effective bounding expression that provides the basis for the remaining B&T techniques.

The form of the resulting expression has an additional advantage; it isolates those parts of the expression that are identical for rotamers on the same level of a sub-tree. Thus it is possible to further increase the efficiency of the search by precomputing these shared quantities as each group of nodes is encountered, rather than redundantly evaluating the entire bounding expression for every unique node. This method is described in the Materials and Methods section.

Termination

The enhancements of the B&T algorithm relative to the B&B method are based on a process called "termination." Because all the pairwise interactions are precomputable, the organization of the combinatorial tree is arbitrary (i.e., there is no specific order to which different amino acid positions must be assigned to different levels of the tree). However, organization of the tree can have a significant influence on the speed of the calculation. For example, a greater reduction in the size of the search is derived from pruning a branch at the root of the tree rather than pruning a branch closer to the leaves. Placing a branch at the leaves that would be pruned if placed at the root would be inefficient because the same pruning step would necessarily be repeated for every leaf.

In fact, it commonly occurs that all amino acid positions have some rotamers that could be pruned if placed at the root of the tree. To circumvent the potential loss of efficiency, we implement a pre-processing procedure before determining the tree organization. This procedure consists of temporarily considering each amino acid position to be at the root level and checking if any of its rotamers can be immediately pruned. All rotamers pruned from root positions may be completely discarded for the remainder of the optimization, and are dubbed "terminated" to reflect this fact. The result is an overall reduction of the tree size prior to searching, making the optimization faster.

The selection of the word "terminate" is intended to be contrasted with "eliminate," which is used to describe rotamers that are analogously discarded by using the DEE criterion. Indeed, many of the same rotamers are discarded. As with DEE, termination may be performed iteratively until no further rotamers are terminated.

Iterative termination is executed as the preprocessing step before search of the tree.

Recursive Termination

Although termination serves as an effective preprocessing step, the hallmark of the B&T algorithm is that termination is employed at every level of recursion. At any point of the search, the rotamers defined at levels above the level of the current amino-acid position may be considered a root comprised of a single, partially specified super-rotamer. Termination, then, consists of temporarily considering each of the rotamers at all the remaining positions as candidates for the next appendage of the super-rotamer and applying the bounding criterion to each one. All rotamers terminated this way may be discarded from the optimization of the sub-tree with this partially specified super-rotamer root.

In contrast, the recursive step in a B&B search consists of application of the bounding criterion to the rotamers at only one amino acid position. The benefits of the extra reductions in the sizes of sub-trees far outweigh the costs of calculation of extra bounds for termination. The resulting increase in efficiency makes the B&T search significantly faster than a similarly constructed B&B search.

Figure V-2 illustrates the benefit of performing termination at every level of recursion. By varying the depth to which recursive termination is applied in place of conventional bounding, it is clearly demonstrated that the extra pruning effects significant reductions in overall search time.

We have observed that it is not necessary to perform iterative termination at every level of recursion, unlike termination preprocessing. A single iteration per branch

generally yields the best performance.

Search Order

When traversing the combinatorial tree, it is necessary to determine (1) the order in which to explore rotamers at each position, and (2) the sequence in which to explore the different positions. For both cases, we utilize the bounding energies calculated for each rotamer during termination.

We have observed an empirical correlation between low bounding energy and membership in the GMEC. Therefore, the rotamers at each position are searched in order of increasing bounding energy. Conducting the search in this way increases the chance that solutions close to the GMEC are found quickly, thereby providing stringent reference energies early in the calculation.

With respect to the ordering of the different positions, we construct a heuristic based on both the termination bounding energies and the size of the rotamer lists. In a conventional tree search, the positions should be organized in order of increasing number of rotamers per position in order to minimize the total number of nodes in the tree. However, in a B&T search, there are other organization schemes that favor high-level pruning by termination, which reduce the tree size more significantly. We use the bounding energy of the top-ranked (lowest bounding energy) rotamer at each position to indicate which positions are likely to restrict the rest of the system, and consequently favor high-level termination if placed at the super-rotamer root. Because the minimum operators at a node are applied over a set including the subset corresponding to the sub-tree nodes, bounding energies of sub-tree nodes must be higher than or equal to their

parent nodes. Therefore, placing positions with high lowest-energies at the top of the tree promotes high bounding energies for their descendents. Since the rotamer lists of a sub-tree can be significantly different than those of its parent, residue ordering is performed at every level of recursion depth.

We have observed that an optimal ordering can be obtained by combining energetic and list-size sorting criteria using the following heuristic. Positions are sorted in descending order according to a rank index, as computed by the expression,

$$Rank\ Index = (1-f)\frac{1}{1+\ln N} + f\frac{E_{top} - E_{top,min}}{E_{top,max} - E_{top,min}} \tag{5}$$

where N is the number of rotamers at the position, E_{top} is the bounding energy of the top-ranked rotamer of that position, and $E_{top,min}$ and $E_{top,max}$ are the minimum and maximum top-ranked bounding energies of all positions, respectively. The expression $1/(1+\ln N)$ is constructed to produce an attenuated weighting inversely proportional to the number of rotamers that evaluates to unity when N=1. The quantity f is selected to control the relative weighting of the two criteria. A value of zero for f corresponds to sort based entirely on the number of residues per position, and a value of one produces a ranking based entirely on bounding energies.

Approximate Algorithm

A solution that is very close to the GMEC sequence can be found very rapidly by using an approximate variation of the B&T method. Approximate calculations are particularly useful for providing a fast way to obtain low reference energies for exact B&T optimizations. Moreover, the approximate calculation is often sufficient to produce the GMEC energy.

The approximation is based on the observation that the GMEC rotamers are often among those with the lowest termination bounding energies according to the bounding expression (Eq. 21 in Materials and Methods). This indicates that the bounding expression has predictive properties. To rapidly find an approximate solution, the ranked rotamer lists are arbitrarily truncated after the pre-processing termination step, and the B&T search is conducted on the abbreviated set of rotamers.

A more reliable solution can be found by repeating the approximate optimization with more lenient truncation, using the solution from the preceding run for the initial reference energy.

DEE Preprocessing

Perhaps the most practical use of the B&T algorithm is to complement DEE when dealing with optimization problems that are too difficult to solve using either algorithm alone. In such cases, the algorithms are used in succession. DEE is used to eliminate rotamers and to perform unification until the optimization reaches iterations that are inefficient. Inefficiency typically occurs after several unifications when the total number of rotamers and unified super-rotamers gets very large (>5000) and very few eliminations result even from lengthy Goldstein doubles calculations. At this stage, the DEE optimization is aborted, and the state information is transferred to a B&T implementation. Rotamer lists and energy tables are transferred directly, including references to unified super-rotamers, which are transparently represented as ordinary rotamers in the B&T algorithm.

An additional performance improvement is obtained by also passing the list of

dead-ending pairs (DEP). DEP's are pairs of rotamers (or super-rotamers) whose members cannot simultaneously exist in the GMEC. These pairs may therefore be safely omitted from the minimum operators in Eq. 21 (see Materials and Methods).

Benchmarks

To assess the generality of the B&T approach, different incarnations of the algorithm were applied to benchmark problems representing different structural classes, as described in Materials and Methods. Optimization times were heavily dependent on the sorting heuristic, as shown in figure V-3. The performance improvement, as measured by dividing the total optimization times, ranged from a factor of three for the β–sheet case to a factor of over forty for the "mixed" case. Remarkably, very similar values of the sorting factor f produced the fastest optimization times for all structural classes. Initially, values at intervals of 0.1 were tested, but since all benchmark cases exhibited minima near $f = 0.1$, values at intervals of 0.01 were sampled near this value. At this level of refinement, the different cases had different optimal sorting factor values, but a value of $f = 0.08$ was close to optimal for all of them. We also observe that optimizations with the fastest times had the fewest nodes in their pruned combinatorial trees.

The total calculation times for the benchmarks using a sorting factor of 0.08 are competitive compared to times of a highly optimized DEE algorithm, and are significantly faster than the optimized B&B search (Table 1). For the β-sheet surface and the small core-boundary calculations, the B&T method is approximately twenty times faster than DEE. For the mixed case, it is nearly eight times faster. For the α-helical

case, however, the B&T method is more than two times slower. This is likely a reflection of the linear topological arrangement of the system, in which it difficult to select positions to place at the tree root that both restrict large parts of the system and are themselves restricted.

The approximate form of the algorithm proved to be exceptionally effective. For the four cases above, B&T calculations that used only the 30 top-ranked rotamers at each position all took less than fifteen seconds and produced the correct GMEC solutions. For the more difficult core-boundary case, the calculation took five minutes, and also produced the correct GMEC solution. For this case, a more aggressive calculation using only the top 15 rotamers at each position took 25 seconds and produced a solution whose energy was in error by less than 1%. This energy was used as the initial bound for the remaining calculations on the system.

To illustrate the potential for combining DEE and B&T methods by way of DEE preprocessing, we selected a problem computable by either algorithm to enable us to perform quantitative comparisons. In practice, however, the technique is applied to problems that are not currently computable in reasonable computer time by either algorithm, for which the benefit is obviously much greater. Figure V-4 illustrates the total calculation times partitioned into DEE and B&T times for optimization of the difficult benchmark consisting of core and boundary residues. The calculations differ in the amount of time allotted to DEE reduction before completion with the B&T algorithm. At the best timing, the combined algorithms complete the optimization eight times faster than DEE alone. Moreover, we have observed that, in practice, the B&T method is generally effective on completing large problems that DEE can reduce to as high as 10^{30}

remaining sequences.

Discussion

We have described a deterministic search method for rotamer optimization and have demonstrated that for some cases it is as fast as the current standard algorithm for protein design, and that for other cases it is much faster. The success of the Branch-and-Terminate method rests on the construction of a novel pairwise bounding expression, which is used both to perform termination and to supply energetic information with which to determine the search order. Although the algorithm is tailored to protein systems, it is generalizable to any problem that can be similarly described.

Although the B&T algorithm is quite effective when used alone, it is perhaps more important that it increases the problem size limit of DEE calculations by providing an efficient way to complete optimizations for which elimination criteria have become less effective at removing rotamers. This makes it possible to perform optimizations on larger proteins and on systems with large numbers of interacting residues.

The size limit may be raised even higher once the limitations of the approximate form of the algorithm become better understood. For the benchmark cases, the approximate algorithm found the GMEC solutions up to a thousand times faster than either of the exact methods. Even the DEE implementation to which the B&T method is compared incorporates some conservative approximations in the form of high energy threshold rejection (HETR) criteria [3]. Analogous techniques may provide a way to construct a faster, approximate B&T algorithm with a clearly defined accuracy. Along the same line of reasoning, truncation based on bounding energies might be an effective

replacement for HETR cutoffs in DEE.

There is also room for improvement in the heuristic for determining search order. Heuristics that are even more effective may exist that make use of structural information, in addition to energetics and size considerations.

In addition, we are currently exploring features of the B&T algorithm that are common to all backtrack searches. First, it is possible to exhaustively sample the amino acid and rotamer sequence space near the GMEC. This is accomplished by modifying the algorithm so that it refrains from lowering the initial minimum energy upon finding low energy combinations [24]. The result is a full enumeration of all sequences with energies below the specified initial minimum energy, provided that this energy is close enough to the GMEC energy that the calculation remains tractable.

Also, it is straightforward to adapt backtrack algorithms for parallel computation by dispatching branches to different computational nodes. We observe a scaling efficiency between 60%-80%, depending on the type of problem. Another advantage of the tree representation is that it makes it possible to estimate how much time the optimization will require. This is accomplished using a well-known tree estimation technique [25] in which statistics are compiled for random sample trajectories through the tree. This has helped us to predict when it is best to transfer DEE problems to B&T for completion.

In practice, we believe that the best way to use the B&T method is to first attempt to optimize a problem using DEE. Upon observing that DEE begins to produce very few eliminations or dead-ending pairs, the state information should be transferred to an approximate form of the B&T algorithm. Using the energy from this calculation as the

initial upper bound, the approximate algorithm may be repeated again with successively more conservative truncations. The final energy should then be used as the initial bound for the exact B&T calculation.

Biological Implications

Protein design and protein homology calculations typically use combinatorial optimization algorithms to compute the optimal placement of amino acid side chain rotamers on protein backbones. The capabilities of exhaustive search algorithms are currently limited by protein size and energy landscapes. The Branch and Terminate variation of the Branch and Bound search algorithm described here provides a way to optimize these problems, both alone and used in conjunction with well established algorithms based on the Dead End Elimination theorem.

Materials and Methods

Benchmark Cases

We test the generality of the algorithm by applying it to a suite of optimization problems representative of different protein structural classes. Rotamers were selected from a backbone dependent library [26]. To test α-helical surface positions, the 12 residues occupying the **b**, **c**, and **f** locations in the heptad repeat of one helix of the coiled-cold GCN4-p1 dimer [27] were optimized from the set of rotamers corresponding to hydrophilic amino acids (A, D, E, H, K, N, Q, R, S, and T). There were 9.1×10^{22} rotameric combinations.

The β1 domain of streptococcal protein G [28] was used for the remaining cases. As a representative of core and boundary optimization problems, a subset of positions determined to be in the core and boundary according to our residue classification scheme (positions 3, 5, 7, 12, 23, 25, 26, 30, 34, 43, 45, 52, 54) were optimized from the 3.4×10^{25} combinations of hydrophobic rotamers (amino acids A, F, I, L, M, V, W, and Y). For β-sheet surfaces, a subset of the β-sheet surface residues (positions 4, 6, 15, 17, 42, 44, 53, 55) were optimized from the 4.9×10^{17} combinations of hydrophilic rotamers.

To represent problems consisting of a mixture of different structural types, including turns, we also included the optimization of the residues containing any atoms within 10 Å of the side-chain atoms of Val 21. Of these 14, the core residues (positions 3, 20, 36) were allowed to have any of the hydrophobic identities, the surface residues (positions 2, 19, 21, 22, 24) had hydrophilic identities, and the remaining boundary residues (positions 1, 18, 23, 25, 27, 29) were selected from a group of hydrophilic and hydrophobic residues, excluding methionine (amino acids A, D, E, F, H, I, K, L, N, Q, R, S, T, V, W, and Y). There were 1.3×10^{29} possible rotameric combinations.

The most difficult benchmark consisted of all 18 non-glycine core and boundary residues [2]. The core residues (positions 3, 5, 7, 20, 26, 30, 34, 39, 52, 54) were selected from the set of hydrophobic amino acids, and the boundary residues (positions 1, 12, 23, 33, 37, 45, 50, 56) were selected from the composite list of hydrophilic and hydrophobic residues. There were 1.9×10^{34} possible rotameric combinations.

Energy Expression

We employ an energy expression that consists of van der Waals, electrostatic, and

solvation terms. For van der Waals, a Lennard-Jones 6-12 potential was used, with radii scaled by a factor of 0.9 [29]. Electrostatics were computed using a distance dependent dielectric and a hybridization-dependent hydrogen-bonding term [30]. Solvation effects were approximated from hydrophobic surface area burial [31]. Atom radii and hydrogen-bond well depths were based on the DREIDING force-field [32].

Calculation

For reference, calculation times were recorded using a fully optimized DEE algorithm incorporating high energy threshold reduction (HETR) [3], and magic bullets and other doubles optimizations [14]. Calculations were also performed using an enhanced B&B implementation that employed the efficient bounding criteria and termination preprocessing.

For the first three benchmark cases, all calculations were performed using an initial upper bound of 0.0 kcal/mol, since our energy expression typically results in optimal sequences with negative energies. For the remaining two cases, initial bounds were obtained by first running the approximate version of the algorithm, in which the rotamer lists were truncated to the 15 rotamers with the lowest bounding energies at each residue position. These provided initial bounds of –153.0 and –250.0 kcal/mol, respectively.

The generality of the sorting criteria was demonstrated by performing optimizations with values of f in Eq. 5 ranging from 0 to 1.

To illustrate the reliability of the approximate form of the algorithm, optimizations were also performed using only the top 30 rotamers at each position as

ranked after a single round of termination.

The larger benchmark problem consisting of core and boundary residues was used to demonstrate how DEE and B&T methods can work in concert. The problem was optimized using a DEE algorithm, and upon every reduction of complexity by at least an order of magnitude, the state of the diminished problem was recorded. A B&T algorithm was used to complete the calculation for each reduced state. The calculations were performed using the optimal sorting factor as determined from the previous benchmarks.

For all calculations, the total CPU time was recorded, as well as the portions of that time spent performing termination preprocessing and the actual recursive search. The total number of nodes comprising the final pruned tree was also recorded by tallying the number of nodes remaining after termination at every level of recursion. Calculations were performed on a single R10000 CPU of a Silicon Graphics Origin 2000.

Pairwise Bounding Expression

This section describes the construction of a stringent expression for a lower bound for a system composed only of one and two-body interactions in terms of both a partially specified sequence and the set of rotamers available at its unspecified positions.

For a system consisting only of two-body interactions, the total potential energy can be expressed as the sum of energies between all pairs.

$$E_{\text{total}} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E(i,j) = \sum_{i} \sum_{\substack{j \\ j>i}} E(i,j) \qquad (6)$$

In a protein, i and j refer to amino-acid positions, and $E(i,j)$ is energy of interaction between amino-acids at those positions.

A protein system also consists of single-body interactions. Because each body is an amino-acid side chain at a particular position on the protein backbone, there is an energy contribution both from side chain interactions with other side chains as well as interactions with the protein template scaffolding. Both energies of interaction depend on the side chain position, amino acid identity, and configuration. Thus the total potential energy can be expressed,

$$E_{\text{total}} = \sum_{i} E(i_c, \text{template}) + \sum_{i} \sum_{\substack{j \\ j>i}} E(i_c, j_c) \tag{7}$$

where c is a position-specific index describing a side chain rotamer of a particular amino acid type and configuration.

For the purposes of deriving an expression for a lower bound, it is desirable to alter the indices to allow redundancy.

$$E_{\text{total}} = \sum_{i} E(i_c, \text{template}) + \tfrac{1}{2} \sum_{i} \sum_{\substack{j \\ j \neq i}} E(i_c, j_c) \tag{8}$$

To ensure that the bounding expression satisfies the condition in Eq. 3, we use the following inequalities:

$$\min_{r} \left[E(i_r, \text{template}) \right] \leq E(i_g, \text{template}) \tag{9}$$

and

$$\min_{r} [E(i_r, j_g)] \leq E(i_g, j_g) \tag{10}$$

in which the indices r and s refer to all of the possible rotamers available at each position, and the minimum operator selects the single rotamer that minimizes the sub-expression. The index g denotes the rotamer found at the specified position in the global minimum

combination. A simple expression for the lower bound is therefore obtained by summing minimal interaction energies between positions by discovering minimal rotamer-pairs.

$$E_{\text{bound}}^{(0)} = \sum_i \min_r \left[E(i_r, \text{template})\right] + \tfrac{1}{2} \sum_i \min_r \sum_{\substack{j \\ j \neq i}} \min_s \left[E(i_r, j_s)\right] \tag{11}$$

The derivation above represents a generic strategy for producing a bounding expression from any total energy expression. For example, more restrictive bounds can be obtained from energy expressions that sum over three or four-body interactions. However, the computational cost to implement such bounds on a protein system is very high. Fortunately, there are variations of Eq. 7 that are equivalent in terms of computational cost yet yield better bounds.

An alternative way to express the total energy of the system is to distribute the template energies into the pair calculation. Given an energy quantity for a pair of rotamers,

$$E_{\text{pair}}(i_c, j_c) = \frac{E(i_c, \text{template}) + E(j_c, \text{template})}{2p - 2} + \frac{E(i_c, j_c)}{2} \tag{12}$$

in which p is the number of amino acid positions, the total energy can be expressed as

$$E_{\text{total}} = \sum_i \sum_{\substack{j \\ j \neq i}} E_{\text{pair}}(i_c, j_c) \tag{13}$$

which, in turn, can be used to produce the following bounding expression:

$$E_{\text{bound}}^{(1)} = \sum_i \min_r \sum_{\substack{j \\ j \neq i}} \min_s \left[E_{\text{pair}}(i_r, j_s)\right] \tag{14}$$

Because the minima must be evaluated with respect to single-body and pair energies simultaneously, this bounding expression is necessarily greater than or equal to

the expression in Eq. 11. Therefore, the new bound is more restrictive. The computational requirements for both expressions, however, are of the same order. Each requires n^2p^2 calculations, where n is the average number of available rotamers per position, and p is the number of positions.

One can derive a lower bound that is even more restrictive by performing an expansion of Eq. 13 before applying the minimization operators. When testing a particular node during traversal of the combinatorial tree, the positions corresponding to nodes above (and including) the current node have uniquely specified rotamers, whereas the remaining, deeper nodes are not yet uniquely specified. The set of all amino acid positions can therefore be decomposed into two subsets, fixed (F) and variable (V). Eq. 13 can be rewritten,

$$E_{\text{total}} = \sum_{i \in \{F,V\}} \sum_{\substack{j \in \{F,V\} \\ j \neq i}} E_{\text{pair}}(i_c, j_c) \tag{15}$$

Next, we expand the summation.

$$E_{\text{total}} = \sum_{i \in F} \sum_{\substack{j \in F \\ j \neq i}} E_{\text{pair}}(i_c, j_c) + \sum_{i \in F} \sum_{j \in V} E_{\text{pair}}(i_c, j_c) + \sum_{i \in V} \sum_{j \in F} E_{\text{pair}}(i_c, j_c) + \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} E_{\text{pair}}(i_c, j_c) \tag{16}$$

Application of the minimum operators to this expression would yield a bounding expression equivalent to Eq. 14. To increase the stringency, we utilize the inequality,

$$\min_r \sum_j E_{pair}(i_r j_s) \geq \sum_j \min_r E_{pair}(i_r j_s) \tag{17}$$

The middle two terms of Eq. 16 differ only in their indices, and are therefore equivalent to one another. However, there is a difference once the minimum operators are applied, since the rotamers of the fixed subset (F) will restrict the selection of the minimum

energy rotamer pair for the minimized third term, but not for the second. Therefore, we reverse the order of the summation for the second term and combine it with the third term to make use of (17) such that the minimum will be as large as possible.

$$E_{\text{total}} = \sum_{i\in F}\sum_{\substack{j\in F\\ j\neq i}} E_{\text{pair}}(i_c, j_c) + 2\sum_{i\in V}\sum_{j\in F} E_{\text{pair}}(i_c, j_c) + \sum_{i\in V}\sum_{\substack{j\in V\\ j\neq i}} E_{\text{pair}}(i_c, j_c) \tag{18}$$

Now we apply the minimum operators to all sums over positions that are not uniquely specified.

$$E^{(2)}_{\text{bound}} = \sum_{i\in F}\sum_{\substack{j\in F\\ j\neq i}} E_{\text{pair}}(i_r, j_s) + 2\sum_{i\in V}\min_r \sum_{j\in F} E_{\text{pair}}(i_r, j_s) + \sum_{i\in V}\min_r \sum_{\substack{j\in V\\ j\neq i}} \min_s E_{\text{pair}}(i_r, j_s) \tag{19}$$

To achieve further stringency, we rearrange Eq. 18 before applying the minimum operators.

$$E_{\text{total}} = \sum_{i\in F}\sum_{\substack{j\in F\\ j\neq i}} E_{\text{pair}}(i_c, j_c) + \sum_{i\in V}\left\{2\sum_{j\in F} E_{\text{pair}}(i_c, j_c) + \sum_{\substack{j\in V\\ j\neq i}} E_{\text{pair}}(i_c, j_c)\right\} \tag{20}$$

From which we obtain,

$$E^{(\text{final})}_{\text{bound}} = \sum_{i\in F}\sum_{\substack{j\in F\\ j\neq i}} E_{\text{pair}}(i_r, j_s) + \sum_{i\in V}\min_r\left\{2\sum_{j\in F} E_{\text{pair}}(i_r, j_s) + \sum_{\substack{j\in V\\ j\neq i}} \min_s\left[E_{\text{pair}}(i_r, j_s)\right]\right\} \tag{21}$$

The expression is generalizable to any system consisting only of two-body interactions such that the total energy of the system can be expressed as in Eq. 13.

Efficient Implementation of Bounding Expression

The computational cost of evaluating Eq. 21 is proportional to p^2n^2, where p is the number of positions and n is the average number of rotamers at each position. When

performing termination, the bound is evaluated for all pn rotamers, so that the total calculation order for a round of termination is p^3n^3.

Termination consists of evaluating the bounding expression for rotamers at all the unspecified positions. Therefore, a position is temporarily considered a member of set F while its rotamers are being evaluated. Since the expensive second term of the final summation is dependent only on V, its possible values may be precomputed for all rotamers i_r once per position and placed into a table for lookup during the evaluation of Eq. (21).

The cost of performing p^2n^2 calculations for assembling the table for the termination of all p positions scales as p^3n^2. The bounding expression now only requires order pn calculations for each of the pn times it is performed, for an overall order of p^2n^2. The overall calculation time therefore scales approximately as p^3n^2, which is nearly n times faster than the direct implementation. Since n is often as large as 100-200, the speed increase can be drastic.

Acknowledgements

We wish to thank A. G. Street for invaluable feedback throughout the process of developing the algorithm. We also wish to thank E. M. Reingold, R. Manohar, and N. Pierce for helpful discussions. This work was supported by the Howard Hughes Medical Institute (S.L.M.), training grant GM 07616C-19 from the National Institutes of Health (D.B.G), the Rita Allen Foundation, and the David and Lucile Packard Foundation.

References

1. Dahiyat BI, and Mayo SL. 1997. De novo design: Fully automated sequence selection. *Science.* **278**, 82-87.

2. Malakaukas S. and Mayo SL. 1998. Design, structure and stability of a hyperthermophilic protein variant. *Nat. Struc. Biol.* **5**, 470-475.

3. De Maeyer M, Desmet J. Lasters I. 1997. All in one: A highly detailed rotamer library improves both accuracy and speed in the modeling of side-chains by dead-end elimination. *Fold. Des.* **2**, 53-56.

4. Janin J, Wodak S, Levitt M, Maigret D. 1978. Conformation of amino acid side-chains in proteins. *J. Mol. Biol.* **125**, 357-386.

5. Ponder J and Richards F. 1987. Tertiary templates for proteins – use of packing criteria in the enumeraion of allowed sequences for different structural classes. *J. Mol. Biol.* **193**, 775-791.

6. Lee C and Levitt M. 1991. Accurate prediction of the stability and activity effects of site-directed mutagenesis on a protein core. *Nature.* **352**, 448-451.

7. Hellinga HW and Richards FM. 1994. Optimal sequence selection in proteins of known structure by simulated evolution. *Proc. Natl. Acad. Sci. USA.* **91**, 5803-5807.

8. Dahiyat BI and Mayo SL. 1994. Protein Design Automation. *Protein Science.* **5**, 895-903.

9. Koehl P and Delarue M. 1994. Application of a self-consistent mean-field theory to predict protein side-chain's conformation and estimate their conformational entropy. *J. Mol Biol.* **239**, 249-275.

10. Lee C. 1994. Predicting protein mutant energetics by self-consistent ensemble

optimization. *J. Mol. Biol.* **236**, 918-939.

11. Desmet J, De Maeyer M, Hazes B, Lasters I. 1992. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature.* **356**, 539-542.

12. Lasters I and Desmet J. 1993. The fuzzy-end elimination theorem – correctly implementing the side-chain placement algorithm based on the dead-end elimination theorem. *Prot. Eng.* **6**, 717-722.

13. Goldstein RF. 1994. Efficient rotamer elimination applied to protein side-chains and related spin-glasses. *Biophys. J.* **66**, 1335-1340.

14. Gordon DB and Mayo SL. 1998. Radical Performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *J. Comp. Chem.* **19**, 1505-1514.

15. Desmet J, De Maeyer M, Lasters I. 1994. In *The Protein Folding Problem and Tertiary Structure Prediction.* Merz Jr., K. & Le Grand, S. Ed. Birkhäuser, Boston. p. 307

16. Reingold EM, Nievergelt J, Deo N. 1977. *Combinatorial Algorithms. Theory and Practice*. Prentice-Hall, New Jersey.

17. Hellinga HW and Richards FM. 1991. Construction of new ligand binding sites in proteins of known structure. *J. Mol. Biol.* **222**, 763-785.

18. Lathrop RH and Smith TF. 1996. Global optimum protein threading with gapped alignment and empirical pair score functions. *J. Mol. Biol.* **255**, 641-665.

19. Escalier V, Pothier J, Soldano H, and Viari A. 1998. Pairwise and multiple identification of three-dimensional common substructures in proteins. *J Comp. Biol.* **5**, 41-56.

20. Wang CSE, Lozano-Perez T, Tidor B. 1998. A systematic algorithm for packing of macromolecular structures with ambiguous distance constraints. *Proteins.* **32**, 26-42.

21. Todorov NP and Dean PM. 1998. A branch-and-bound method for optimal atom-type assignment in de novo ligand design. *J. Comp. Aid. Mol. Des.* **12**, 335-349.

22. Eyrich VA, Standley DM, Felts AK. Friesner RA. 1999. Protein Tertiary Structure Prediction Using a Branch and Bound Algorithm. *Proteins.* **35**, 41-57.

23. Samudrala R and Moult J. 1998. A graph-theoretic algorithm for comparative modeling of protein structure. *J. Mol. Biol.* **279**, 287-302.

24. Leach AR and Lemon AP. 1998. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. *Proteins.* **33**, 227-239.

25. Knuth DE. 1975.Estimating the efficiency of backtrack programs. *Math. Comput.* **29**, 121-136.

26. Dunbrack RL and Karplus M. 1993. Backbone-dependent rotamer library for proteins – application to side-chain prediction. *J. Mol. Biol.* **230**, 543-574.

27. O'Shea EK, Klemm, JD, Kim PS Alber T. 1991. X-ray structure of the GCN4 leucine zipper, a 2-stranded, parallel coiled coil. *Science.* **254**, 539-544.

28. Gallagher T, Alexander P, Bryan P, Gilliland GL. 1994. 2 Crystal structures of the β1 immunoglobulin binding domain of streptococcal protein-G and comparison with NMR. *Biochemistry.* **33**, 4721-4728.

29. Dahiyat BI and Mayo SL. 1997. Probing the role of packing specificity in protein design. *Proc. Natl. Acad. Sci. USA.* **94**, 10172-10177.

30. Dahiyat BI, Gordon DB, and Mayo SL. 1997. Automated design of the surface position of protein helices. *Prot. Sci.* **6**, 1333-1337.

31. Street AG and Mayo SL. 1998. Pairwise calculation of protein solvent-accessible surface areas. *Fold. Des.*. **3**, 253-258.

32. Mayo SL, Olafson BD, Goddard WA. 1990. DREIDING – A generic force-field for molecular simulations. *J. Phys. Chem.* **94**, 8897-8909.

Table 1.

Benchmark Times

	Benchmark Cases				
	small C-B[a]	α-surface	β-surface	Mixture	Core-Boundary
Total Times [min]					
DEE[b]	177.4	2.2	40.5	101.6	1154.0
B&B[c]	70.7	294.8	44.4	544.9	>30000[d]
B&T[e]	8.4	6.1	2.1	13.0	745.8
B&T Component Times [min]					
Preprocessing	0.1	0.1	0.1	0.4	0.6
Search	8.3	6.0	2.0	12.4	744.8
Approximation[f]				0.2	0.4
B&T Total Nodes	3829	1697	1546	845	34634

[a]Refers to the benchmark comprised of a small set of core and boundary positions.

[b]DEE was performed using the speed enhancements in [13] and [14].

[c]The B&B algorithm uses the novel bounding expression and includes termination preprocessing.

[d]For the difficult Core-Boundary case, the incomplete B&B optimization was aborted after 30,000 minutes.

[e]Total B&T time is computed of as the sum of the approximation, preprocessing, and search times.

[f]An approximate B&T algorithm was used to obtain initial bounds for the Mixture and difficult Core-Boundary cases. These calculations used only the top 30 rotamers at each position according to their bounding energy.

Figure V-1: Schematic of combinatorial tree representing a side chain optimization problem. Four positions, numbered 1-4 are represented, each having either two or three rotameric choices. A complete rotameric arrangement can be described according to the nodes in the tree that specify a path to the lowest level. Search time may be reduced if it can be determined that a partial arrangement, as represented by a partial path through the tree, cannot have any sub-branches that describe the GMEC. In such a case the sub-branches need not be searched (grayed branches).

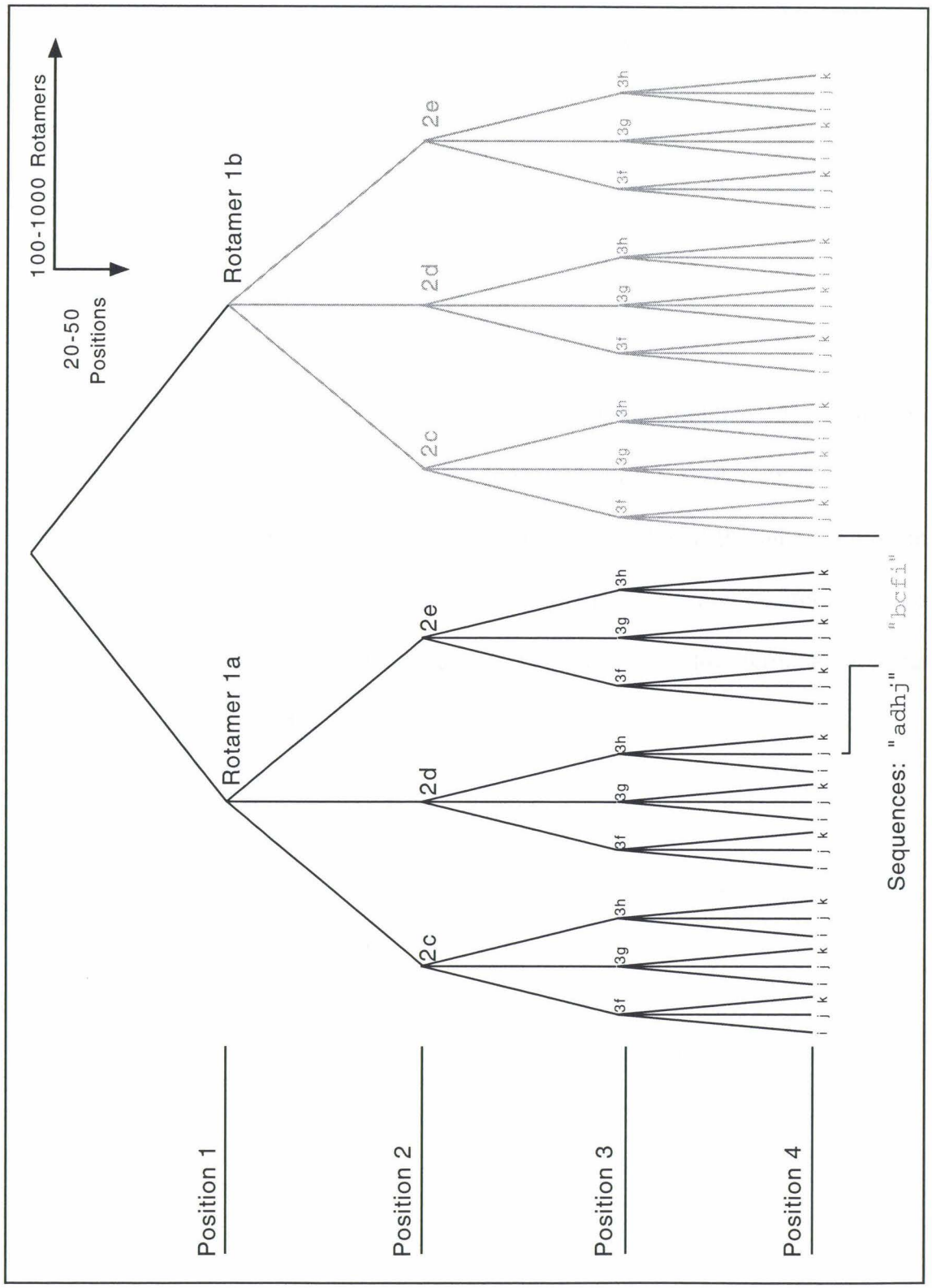

100-1000 Rotamers
20-50 Positions
Rotamer 1a
Rotamer 1b
2c
2d
2e
3f
3g
3h
i j k
Position 1
Position 2
Position 3
Position 4
Sequences: "adhj"
"bcfi"

Figure V-2: Total optimization time vs. level of depth in the combinatorial tree to which the termination procedure was recursively applied for a 12 residue optimization. A termination depth of zero corresponds to a standard Branch-and-Bound algorithm that uses the novel bounding expression and includes a single termination preprocessing step. A termination depth of 12 corresponds to a calculation that employs termination instead of bounding at every opportunity.

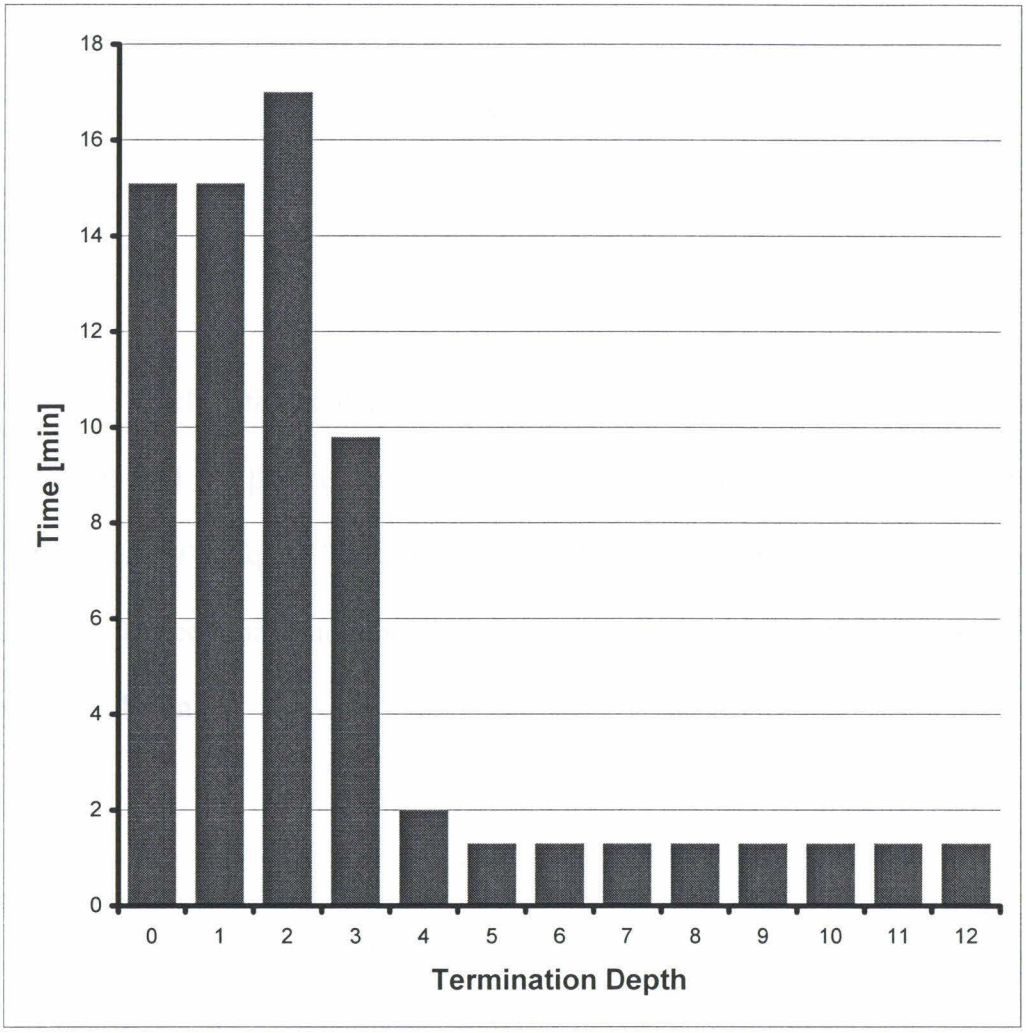
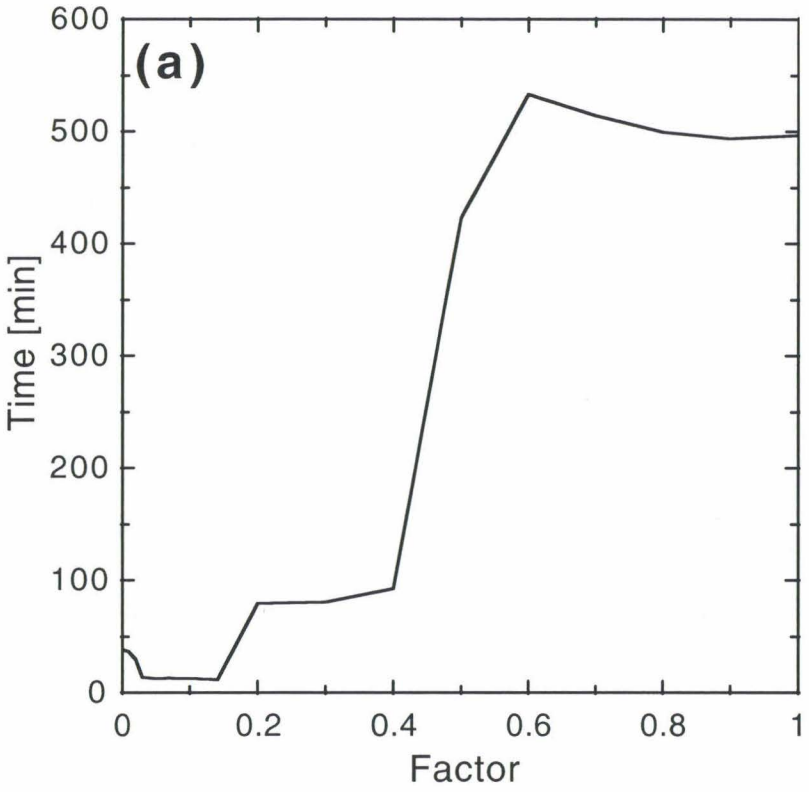

18
16
14
12
10
8
6
4
2
0
Time [min]
0
1
2
3
4
5
6
7
8
9
10
11
12
Termination Depth

Figure V-3: Total optimization time vs. value of sorting factor for the (a) mixed structural type and (b) β-sheet surface benchmark cases. Sorting is determined by the value of the factor f in Eq. 5. The cases exhibit different dependencies on the value of the sorting factor, but both have minima in the vicinity of $f = 0.08$. This trend is observed for all cases (not shown).


(a)
600
500
400
300
200
100
0
Time [min]
0
0.2
0.4
0.6
0.8
1
Factor

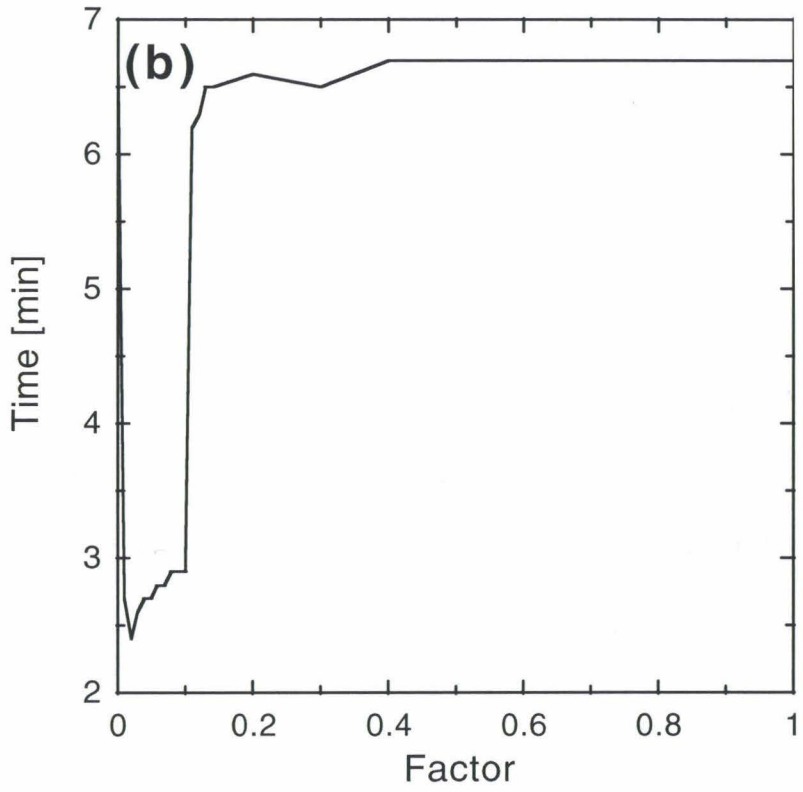
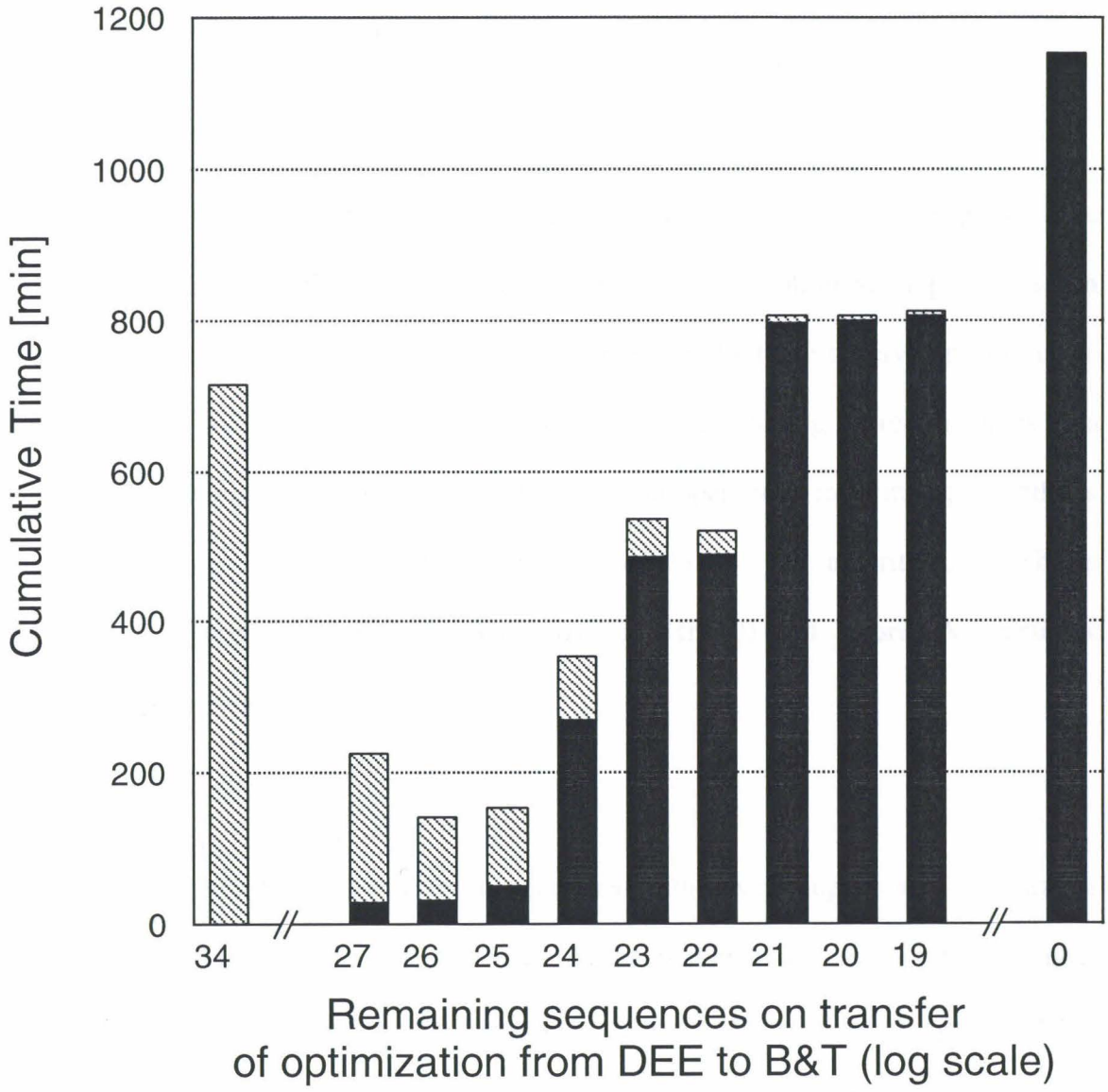

(b)
Time [min]
Factor
7
6
5
4
3
2
0
0.2
0.4
0.6
0.8
1

Figure V-4: Optimization times resulting from the combination of B&T (hashed bars) and DEE (solid bars) algorithms. The bars on the extreme left and right of the figure are the times for lone B&T and DEE optimization, respectively. The remaining bars are the cumulative B&T and DEE optimization times when the two algorithms are used in succession. The sudden jumps in DEE times arise from lengthy Goldstein doubles calculations.


1200
1000
800
600
400
200
0
Cumulative Time [min]
34
27
26
25
24
23
22
21
20
19
0
Remaining sequences on transfer
of optimization from DEE to B&T (log scale)

Chapter VI

Hybrid Algorithms for Rotamer Optimization

This chapter summarizes ongoing work performed in collaboration with Professor Niles A. Pierce and Professor Stephen L. Mayo.

Abstract

In developing combinatorial optimization algorithms for protein design, we have observed that using multiple algorithms in series to treat optimization problems can dramatically reduce the required search time. Motivated by these observed performance enhancements, we sought to determine the utility of combining search methods at a deeper level by assembling algorithmic elements and operations from multiple methods into a hybrid algorithm. This exploration has yielded two new algorithms, which we refer to as Hybrid Exact Rotamer Optimization (HERO) and Hybrid Approximate Rotamer Optimization (HARO).

Introduction

Computational protein design is concerned with developing a set of quantitative search criteria that can be used to evaluate the compatibility of amino acid sequences with the protein backbones of design targets [1]. The set of search criteria typically takes the form of an energy expression comprised of potential energy terms inspired by traditional molecular mechanics and dynamics force fields [2]. Often the energy expression also includes additional, non-physical terms, which are also thought to be important for protein design. Design consists of finding the amino acid sequence most compatible with the target fold according to the energy expression.

The number of possible amino acid sequences is exponentially dependent on the number of designed positions. For small protein design problems, the number of possible sequences may be relatively small. In such cases, it may be feasible to find the optimal sequence by enumerating all amino acid combinations. However, for the majority of design calculations, the number of combinations is far larger than can be fully enumerated in a reasonable amount of time. The number is further amplified by the inclusion of side chain flexibility, which is implemented by subdividing amino acids into statistically significant rotamer conformations [3, 4]. For these larger problems, it is necessary to employ a combinatorial search algorithm to find the best sequences without explicit evaluation of every possible sequence [5].

To this end, several combinatorial search strategies have been applied to protein design. These search algorithms may be broadly categorized as being either approximate or exact. Emergent approximate algorithms for protein design include Monte Carlo (MC) [6, 7], genetic algorithms (GA) [9], and self-consistent mean field (MF) approaches [10, 11]. Exact algorithms have been largely based on the Dead-End Elimination theorem (DEE) [12-16], and have been successfully used for large high-resolution protein design problems. In our laboratory, the tree search algorithm Branch-and-Terminate (B&T) is also used [17], primarily to extend the utility of DEE to the exact solution of larger problems.

When an exact algorithm completes its search, it produces the amino acid sequence that best satisfies the energy expression. In terms of searching the energy landscape of sequence space, this sequence corresponds to the global minimum energy conformation (GMEC). The limitation of exact algorithms is that they are not guaranteed

to complete their search within reasonable amounts of computer time. Conversely, approximate algorithms enable control over their execution time, but cannot guarantee discovery of the GMEC. Rather, they find local minima on the energy landscape, which may or may not correspond to the GMEC.

Because the accuracy of approximate search algorithms cannot be known a priori, it has been important to develop exact algorithms both to avoid ambiguities in interpreting design results and to validate approximate methods. Computational protein design relies on using experimental feedback from designed sequences in order to refine the search criteria, so it is critical that sequences accurately represent all the features of the energy expression. Otherwise, refinement of the search criteria may erroneously reflect artifacts of the optimization process rather than the energy expression. This issue is most directly circumvented by using exact search algorithms. A source of furthur motivation stems from the fact that approximate algorithms may also represent the energy expression accurately, but whether an approximate algorithm has this quality can only be definitively determined by comparative validation with an exact algorithm. We have recently described such an evaluation of several approximate methods [18].

To accommodate protein design problems as they increase in complexity, the exact DEE algorithm has undergone several refinements since its introduction as a tool for homology modeling. The original embodiment outlines the basic algorithmic approach, which consists of the iterative dead-end elimination of both individual rotamers [12] and pairs of rotamers [13], and the occasional unification of pairs of rotamers into super-rotamers [19]. Goldstein [14] showed that original DEE criterion was in fact a special case of a more general expression, and that higher order elimination criteria could

achieve more effective rotamer elimination. To address the increased time demands incurred by using the Goldstein elimination criteria, particularly for eliminating pairs of rotamers, we developed metrics [16] to increase the efficiency of their application. Recently, an additional, more sophisticated method for applying the elimination criteria to individual rotamers has been described, using a technique called conformational splitting [20].

Concurrently with these DEE refinements, conventional tree-search algorithms also have been developed for protein design, based on Branch-and-Bound (B&B) approaches [17, 21]. B&B search consists of incremental assembly of a rotamer sequence, coupled with backtracking upon determination that the partial sequence cannot be completed to form the GMEC. This determination is accomplished through use of a quantitative bounding expression, which is constructed to compute a lower energy bound on the possible ways of completing a partially assembled sequence. Our group has focused on an implementation of the Branch-and-Bound approach called Branch-and-Terminate. Among the enhancements over standard breadth-first B&B is the recursive application of the bounding criterion to all rotamers at all positions as a preprocessing step. This procedure is dubbed "termination" because like "elimination," rotamers removed via this mechanism are certain not to be members of the GMEC.

Despite the similarity, dead-end elimination criteria and bounding criteria remove rotamers using fundamentally different principles. A bounding criterion removes a rotamer by comparing the lower energy bound of possible sequences containing it to the total energy of a known reference sequence. The DEE criteria, on the other hand, attempt to show that one rotamer is preferred over another in all circumstances. This type of

operation is an example of a "dominance relation" [22]. Dominance relations in combinatorial search focus on relationships between alternatives, rather than comparison of choices to a reference state. In the case of DEE, rotamers are removed by finding others that "dominate" them in all circumstances. Dominance relations are commonly used in tree-search algorithms in conjunction with, and sometimes instead of, bounding criteria. Lasters et al. [15] describe such a search for rotamer placement using only DEE criteria, which they call "combinatorial build-up." In our hands, however, DEE dominance criteria did not perform as well in tree-search as BAT bounding criteria.

Motivated by the success of bounding criteria in the context of the BAT algorithm, we sought to determine the utility of transplanting bounding criteria into the DEE algorithm. As with the DEE dominance criteria, bounding criteria may be used to eliminate both individual rotamers and pairs of rotamers. Thus they can provide the same types of information as conventional DEE criteria. Since the bounding criteria use different measures than dominance, they have the potential to augment the DEE reductions and enhance the performance of the algorithm. However, bounding criteria also require an additional item of information for their application: the energy of a reference sequence to which bounding energies may be compared. An approximate algorithm such as a stochastic MC search may be used to determine this reference energy since it need only be close to the GMEC.

With these tools in hand, we have constructed a new algorithm that unifies the most effective optimization strategies from DEE, B&T, and MC search methods. We refer to the approach as the Hybrid Rotamer Optimization method (HRO).

In experimenting with different HRO implementations, two interesting variations have emerged, one exact (HERO) and one approximate (HARO). As described below, the exact HERO implementation exhibits significant improvement in optimization speed as compared to DEE. Moreover, it is able to solve problems that were previously intractable for DEE. The approximate HARO variation is interesting because of its speed and accuracy on problems inaddressable by any known exact algorithm. For the majority of test cases, the approximate HARO algorithm finds the GMEC sequence, as determined by an exact algorithm, in a fraction of the search time. Moreover, for problems presently too difficult to be addressed by exact algorithms, the approximate HERO implementation often finds lower energies than other approximate algorithms.

Descriptions of both the exact HERO and approximate HARO algorithms follow. However, the exact HERO algorithm is currently still under development in collaboration with Dr. Niles Pierce. Therefore, the description presented here will be limited to its underlying algorithmic principles and initial observations.

Bounding Expression

Bounding expressions are used in tree search algorithms to assess whether a branch of a tree, corresponding to a fixed arrangement of rotamers at a subset of the residue positions, may be pruned [22]. Pruning a branch close to the root of the tree saves considerable search time by determining that it is unnecessary to individually examine any of the leaves emanating from the branch. Such examination would correspond to evaluating all the combinations of rotamer arrangements that share the fixed subset of rotamers. Making the determination that exhaustive examination is unnecessary is accomplished by first computing an underestimate of the best energy that

could be obtained in the context of the fixed rotameric arrangement using optimal selection of rotamers at the remaining residue positions. The estimated energy, which is interpreted as a lower bound, is then compared to the energy of a reference sequence. In the event that the lower bound is higher than the reference energy,

$$E_{\text{bound}} > E_{reference} \tag{1}$$

it can be known with certainty that the arrangement of fixed rotamers do not comprise a subset of the rotamers found in the GMEC.

There are many possible ways of constructing an expression to compute the lower energy bound for a given set of rotamers. The expression that yields the best performance in the B&T algorithm requires the definition of the quantity E_{pair},

$$E_{\text{pair}}(i_r, j_s) = \frac{E(i_r, \text{template}) + E(j_s, \text{template})}{2p-2} + \frac{E(i_r, j_s)}{2} \tag{2}$$

where $E(i_r, \text{template})$ and $E(j_s, \text{template})$ are the energies of interaction between the template and rotamers r at position i and s at position j, respectively, and $E(i_r, j_s)$ is the interaction energy between the two rotamers. The quantity p is the total number of residue positions involved in optimization, and is necessary to normalize the E_{pair} energies. The bounding expression is:

$$E_{\text{bound}} = \sum_{i \in F} \sum_{\substack{j \in F \\ j \neq i}} E_{\text{pair}}(i_r, j_s) + \sum_{i \in V} \min_r \left\{ 2 \sum_{j \in F} E_{\text{pair}}(i_r, j_s) + \sum_{\substack{j \in V \\ j \neq i}} \min_s \left[E_{\text{pair}}(i_r, j_s) \right] \right\} \tag{3}$$

The set of residue positions F are those that contain the fixed set of rotamers under scrutiny. The remaining positions, V, are considered as variable. In all cases, the "min" operator is used to find the rotamer that minimizes the energy of the specified subexpression.

To use the bounding expression in the context of a DEE algorithm, all that is necessary is to tailor it so that it may be simply applied to either individual rotamers or pairs of rotamers. This is accomplished by considering the set F as consisting of only one or two fixed rotamers. For the case of an individual rotamer t at a position k, the expression is

$$E_{\text{bound}}^{\text{single}} = \sum_{i \neq k} \min_{r} \left\{ 2E_{\text{pair}}(i_r, k_t) + \sum_{\substack{j \neq i \\ j \neq k}} \min_{s} \left[E_{\text{pair}}(i_r, j_s) \right] \right\} \quad (4)$$

where i and j refer to all positions that are not k. As with B&T, these expressions are best implemented by first precomputing the possible values for the second subexpression inside the large summation since the values are invariant for different rotamers t at position k. To test a pair of rotamers k_t and m_u, the expression is

$$E_{\text{bound}}^{\text{pair}} = 2E_{pair}(k_t, m_u) + \sum_{i \neq k \neq m} \min_{r} \left\{ 2E_{\text{pair}}(i_r, k_t) + 2E_{\text{pair}}(i_r, m_u) + \sum_{\substack{j \neq k \neq m \\ j \neq i}} \min_{s} \left[E_{\text{pair}}(i_r, j_s) \right] \right\} \quad (5)$$

where, like the expression for single rotamers, i and j are indexes to all variable positions, which are all positions that are not k or m. Similarly, the final subexpression is invariant with selection of rotamers t and u for a single pair of positions k and m, and may be precomputed for better efficiency.

HARO Results

Threshold by Bounding Energy

The philosophy behind the HARO algorithm may be understood in terms of a previously described method of enhancing DEE performance. De Mayer et al. [15]

described an operation called high-energy threshold reduction (HETR), which provided a means to quickly remove a large number of rotamers unlikely to be members of the GMEC. This was accomplished by applying a high-energy cutoff relative to the self-energy of the lowest-energy rotamer at each position. Although the technique introduces the possibility for error into the algorithm, it was deemed safe when used in conjunction with a highly detailed rotamer library.

Indeed, we have observed that with the selection of a threshold energy that is sufficiently high, DEE with HETR applied to both rotamers and pairs of rotamers is fast and accurate. The calculation time may be reduced by lowering the threshold, but at the expense of a possible loss of accuracy. There is typically a critical threshold value under which the calculation runs quickly, but does not produce the correct answer.

The inaccuracies of the HETR approach become significant when addressing optimization problems that are difficult or very large. In such cases, the DEE algorithm often cannot be made to converge unless unreliable threshold values are employed. To understand this behavior, the HETR approach may be interpreted as using the self-energies of candidate rotamers as metrics to predict their likelihood of participation in the GMEC. Inaccuracies arise from the fact that interaction energies between side chains may overshadow self-energies, preventing them from acting as a reliable indicator of GMEC membership.

It would therefore be desirable to construct a metric that could more reliably predict whether a particular rotamer has prospects for membership in the GMEC. With such a tool, more aggressive thresholds could be applied, potentially making it possible to solve larger and harder optimization problems. To this end, we have already observed

empirically that the bounding criteria have predictive properties, which serve as the basis for the approximate B&T algorithm [17]. In fact, for some small problems, we have observed that the top-ranked rotamers according to the bounding expression are the GMEC rotamers. Therefore, by substituting bounding energies for self-energies as more reliable metrics for HETR, it should be possible to apply more aggressive cutoffs.

The potential benefit of using bounding energies instead of self-energies for approximating the likelihood of GMEC membership may be inferred from figure VI-1. A significant number of rotamers with low self-energies have high bounding energies. As a result, although there is a weak correlation between the two metrics, there is significant enough skew that similar threshold values (20 kcal/mol in the figure) result in significantly different levels of reduction. Because bounding energies include self-energies as well as information about possible interactions with other positions, the set of rotamers that survive the bounding energy threshold is significantly smaller, but is at the same time likely to be comprised of better guesses.

Because the HARO algorithm is approximate, its utility is best evaluated in comparison to other approximate algorithms. To be useful, HARO needs to demonstrate advantages over other approximate algorithms for cases that are too large or difficult for exact algorithms. We therefore selected a benchmark case representing a search size greater than 10^{100}, which we have thus far been unable to search with any exact algorithm.

For this benchmark, the results of substituting bounding energies for HETR operations are shown in figure VI-2. Although there are threshold values that enable the HARO algorithm to achieve convergence in a short amount of time, the resulting energies

are much higher than those produced by MC search, which also takes significantly less time. Although the energies produced by HARO improve as the threshold is increased, it does not appear that the algorithm in this form will be able to compete with MC.

Threshold by Rank

The HARO approach is improved by altering the way in which the threshold is implemented. As illustrated in figure VI-3, there is an even greater disparity between predicted rankings by bounding energies and self-energies than there is between the energies themselves. This is because the distribution of energies is not uniform, particularly in the case of bounding energies.

To incorporate ranking information into the cutoff, two additional quantities are required. We define a preservation fraction f_p and a minimum number of rotamers to preserve n_p. At the beginning of the optimization, and after every round of unification, the rotamers are ranked according to their bounding energies, and a fraction f_p of the rotamers are preserved, while the remaining fraction $1\text{-}f_p$ are discarded for the remainder of the calculation. However, in the case that at least n_p rotamers would not remain, the top n_p rotamers are preserved instead, and the rest are discarded. This prevents the premature reduction to a single rotamer based solely on approximate rankings. To account for the increased representation required by super-residues formed by unification, the value of n_p is scaled at each residue or super-residue position by the number of constituent residues.

The result of using the more sophisticated rotamer selection scheme on the difficult benchmark case is illustrated in figure VI-4. Here, the HARO algorithm is demonstrated to be faster and more accurate than the other approximate algorithms,

finding a lower energy than was found by any other algorithm, and in less time. Moreover, this computation achieved such accuracy even though it employed fairly aggressive values of n_p and f_p. HARO therefore has the advantage that the accuracy of the calculation may be improved by increasing the values of these parameters. The relative increases in search time and accuracy for the difficult benchmark case are illustrated in figure VI-5.

HERO Results

Although the exact HERO algorithm employs bounding energies, it uses them for a very different purpose than the HARO does. The HERO technique is based on the observation that with repeated rounds of unification, the number of rotamer pairs that can be terminated using the bounding expression increases. Coincidentally, the number of pairs capable of being bounded often becomes significant at the same point in the optimization at which the number of new dead-ending pairs, as determined by doubles DEE criteria, becomes small. Thus the bounding criteria can provide a significant way to augment the list of rotamer pairs that cannot exist in the GMEC. As additional benefit, the time dependence of the doubles bounding condition on the number of positions and rotamers is different than DEE Goldstein doubles, making it a significantly faster calculation in many instances.

Proper use of a bounding expression requires a low reference energy to which bounding energies may be compared. Because the reference energy may be approximate, an MC search will suffice, and MC has the additional advantage that it can find good reference energies relatively quickly. We have therefore inserted a short MC search into the algorithm that is performed before every iteration of bounding pairs. As more

rotamers are eliminated by other criteria, the quality of the MC reference energy calculation improves, thereby increasing the efficacy of the following computation of bounding pairs.

The resulting hybrid algorithm is capable of addressing problems of size and complexity that have not been addressable with any known exact algorithm or combination of exact algorithms. Figure VI-6 illustrates one such calculation for which there has been no exact solution found. The search time is compared to a highly optimized DEE algorithm.

Conclusions

We have demonstrated that elements from different combinatorial search methods can be combined to produce new algorithms that outperform previous optimization approaches. In particular, two algorithms have been validated, one exact and one approximate. The exact algorithm, HERO, has been shown to be able to determine the optimal solution for problems previously insoluble by other known algorithms. For problems too difficult to be optimized exactly, the approximate algorithm, HARO, has demonstrated the ability to find approximate solutions faster and with better accuracy than other known approximate methods. Due to the success of these initial explorations, we believe that there is great opportunity for further development as methods for algorithmic hybridization are refined in the future.

References

1. Street AG and Mayo SL. 1999. Computational Protein Design. *Structure.* **7**, R105-109.
2. Gordon DB, Marshal SA, Mayo SL. 1999. Energy functions for protein design. *Curr. Opin. Struct. Biol.* **9**, 509-513.
3. Janin J, Wodak S, Levitt M, Maigret D. 1978. Conformation of amino acid side-chains in proteins. *J. Mol. Biol.* **125**, 357-386.
4. Ponder J and Richards F. 1987. Tertiary templates for proteins – use of packing criteria in the enumeraion of allowed sequences for different structural classes. *J. Mol. Biol.* **193**, 775-791.
5. Desjarlais JR and Clarke ND. 1998. Computer search algorithms in protein modification and design. *Curr. Opin. Struct. Biol.* **8** (4), 471-475.
6. Lee C and Levitt M. 1991. Accurate prediction of the stability and activity effects of site-directed mutagenesis on a protein core. *Nature.* **352**, 448-451.
7. Hellinga HW and Richards FM. 1994. Optimal sequence selection in proteins of known structure by simulated evolution. *Proc. Natl. Acad. Sci. USA.* **91**, 5803-5807.
8. Dahiyat BI and Mayo SL. 1994. Protein Design Automation. *Protein Science.* **5**, 895-903.
9. Lazar GA, Desjarlais JR, Handel TM. 1997. De novo design of the hydrophobic core of ubiquitin. *Prot. Sci.* **6**, 1167-1178.
10. Koehl P and Delarue M. 1994. Application of a self-consistent mean-field theory to predict protein side-chain's conformation and estimate their conformational entropy. *J. Mol Biol.* **239**, 249-275.
11. Lee C. 1994. Predicting protein mutant energetics by self-consistent ensemble

optimization. *J. Mol. Biol.* **236**, 918-939.

12. Desmet J, De Maeyer M, Hazes B, Lasters I. 1992. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature.* **356**, 539-542.
13. Lasters I. and Desmet J. 1993. The fuzzy-end elimination theorem – correctly implementing the side-chain placement algorithm based on the dead-end elimination theorem. *Prot. Eng.* **6**, 717-722.
14. Goldstein RF. 1994. Efficient rotamer elimination applied to protein side-chains and related spin-glasses. *Biophys. J.* **66**, 1335-1340.
15. De Maeyer M, Desmet J, Lasters I. 1997. All in one: A highly detailed rotamer library improves both accuracy and speed in the modeling of side-chains by dead-end elimination. *Fold. Des.* **2**, 53-56.
16. Gordon DB and Mayo SL. 1998. Radical Performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *J. Comp. Chem.* **19**, 1505-1514.
17. Gordon DB and Mayo SL. 1999. Branch and Terminate: A combinatorial optimization algorithm for protein design. *Structure*. **7** (9) 1089-1098.
18. Voigt CA, Gordon DB, Mayo SL. 2000. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *J. Mol. Biol.* In press.
19. Desmet J, De Maeyer M, Lasters I. 1994. In *The Protein Folding Problem and Tertiary Structure Prediction.* Merz Jr., K. & Le Grand, S. Ed. Birkhäuser, Boston. p. 307.
20. Pierce NA, Spiret JA, Desmet J, Mayo SL. 2000. Comformational splitting: A more powerful criterion for dead-end elimination. *J. Comp.Chem.* In press.

21. Leach AR and Lemon AP. 1998. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. Proteins. 33, 227-239.

22. Papadimitriou CH and Steiglitz K. 1982. Combinatorial Optimization: Algorithms and complexity. Prentice Hall. New Jersey.

Figure VI-1: Comparison of self-energy and bounding energies for a set of 1800 rotamers. The highlighted regions delimit rotamers that would be accepted by a 20 kcal/mol cutoff by self energy (S) and lower bounding energy (LB). The darker region of overlap highlights rotamers that would be accepted by either metric. The lower bound avoids unnecessary inclusion of rotamers that have low self-energies but are nevertheless poor choices according to their bounding energies.

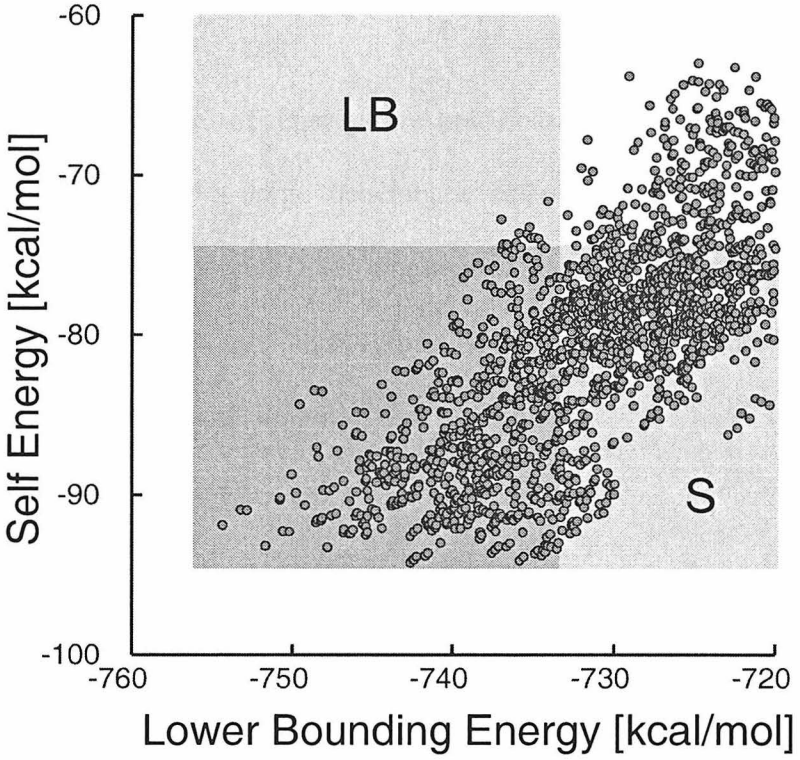
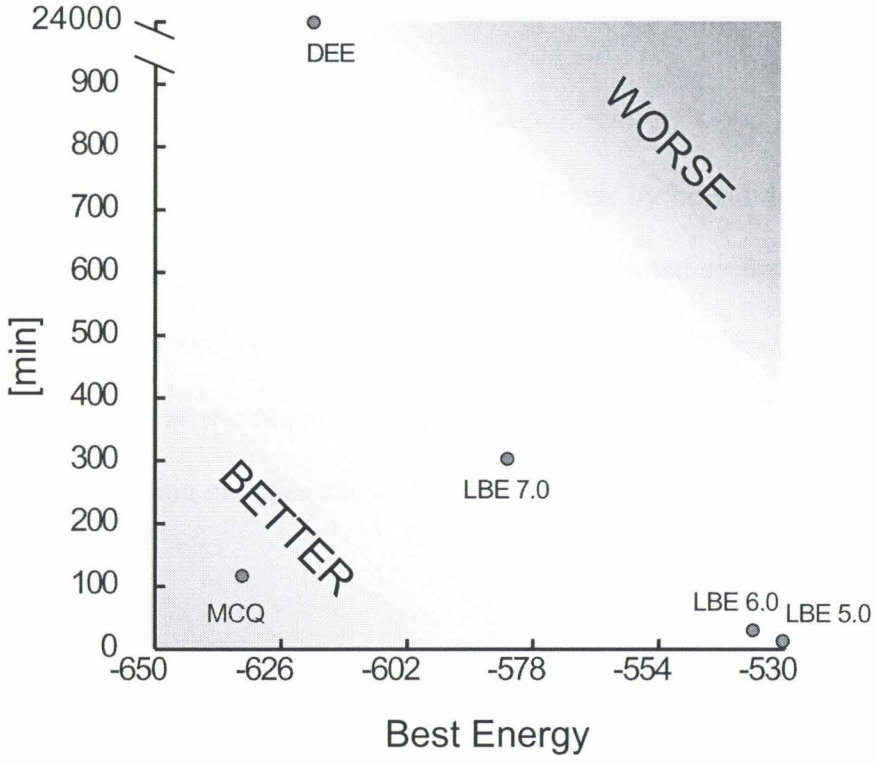

LB
S
-60
-70
-80
-90
-100
-760
-750
-740
-730
-720
Self Energy [kcal/mol]
Lower Bounding Energy [kcal/mol]

Figure VI-2: Performance of energy-threshold based HARO compared to other approximate algorithms for a large benchmark calculation. DEE refers to a DEE algorithm using a low HETR cutoff for singles and pairs. MCQ refers to a quenched simulated annealing Monte Carlo algorithm. The LBE captions denote the threshold energy used for each HARO calculation.


24000
900
800
700
600
500
400
300
200
100
0
[min]
DEE
WORSE
LBE 7.0
BETTER
MCQ
LBE 6.0
LBE 5.0
-650
-626
-602
-578
-554
-530
Best Energy

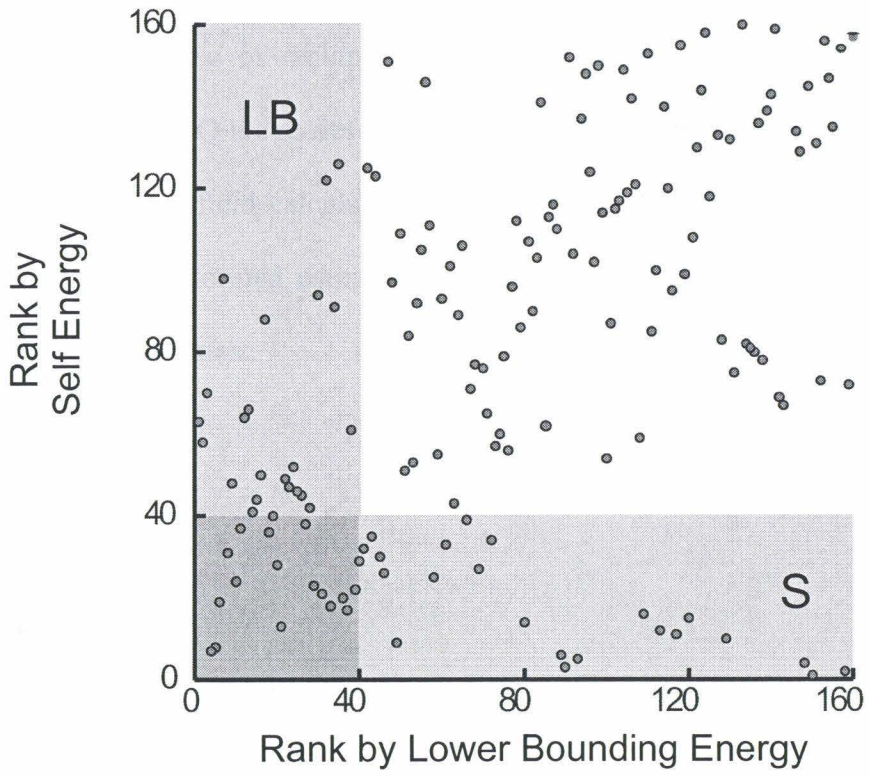

LB
S
160
120
80
40
0
0
40
80
120
160
Rank by
Self Energy
Rank by Lower Bounding Energy

Figure VI-4: Performance of ranking-based HARO compared to other approximate algorithms. DEE and MCQ are as described in figure VI-2. The genetic algorithm (GA) and self-consistent mean field calculations were performed as described in [18]. The HARO calculation is performed using a f_p of 0.5 and a n_p of 20 rotamers. The total calculation time is 22 minutes.

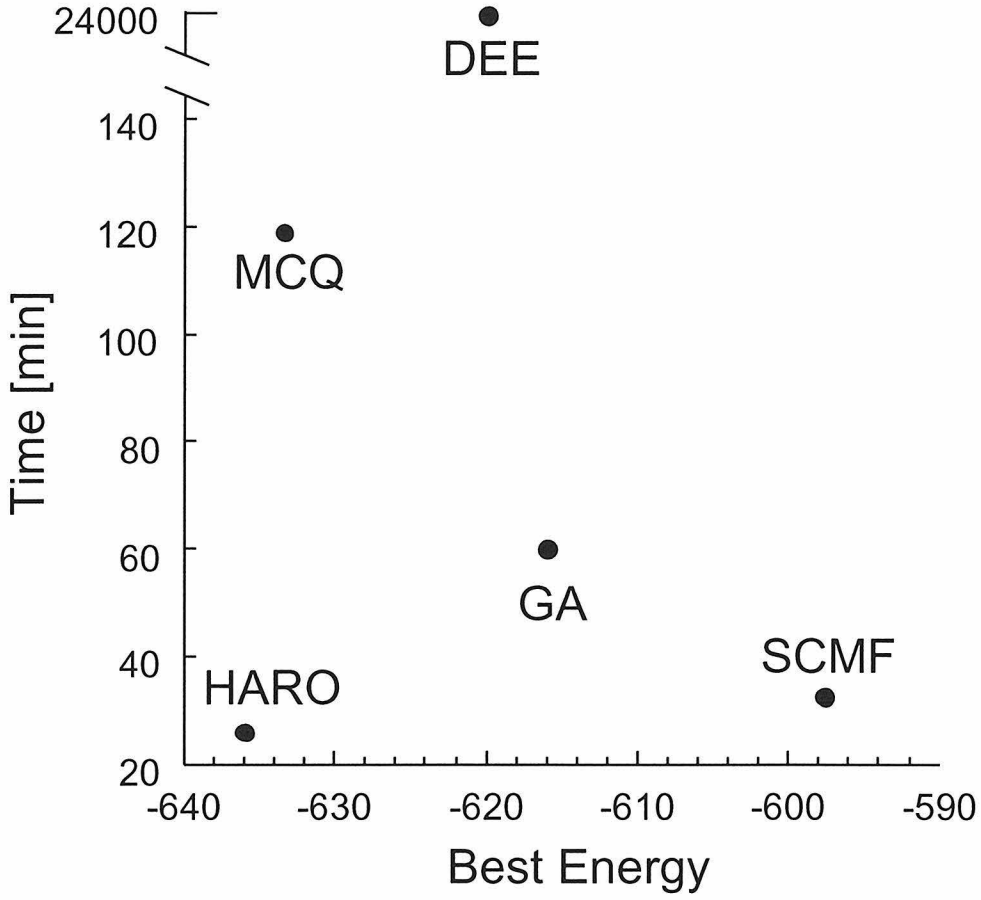

24000
140
120
100
80
60
40
20
Time [min]
DEE
MCQ
GA
SCMF
HARO
-640
-630
-620
-610
-600
-590
Best Energy

Figure VI-5: Dependence of HARO accuracy and time on n_p. The number of rotamers retained at each position, scaled according to super-rotamer construction as described in the text, is incremented by ten rotamers for each calculation. The energy produced by the calculation is denoted by each point. There was no observed improvement when n_p was set to more than 30 rotamers, which may be an indication that GMEC was found.

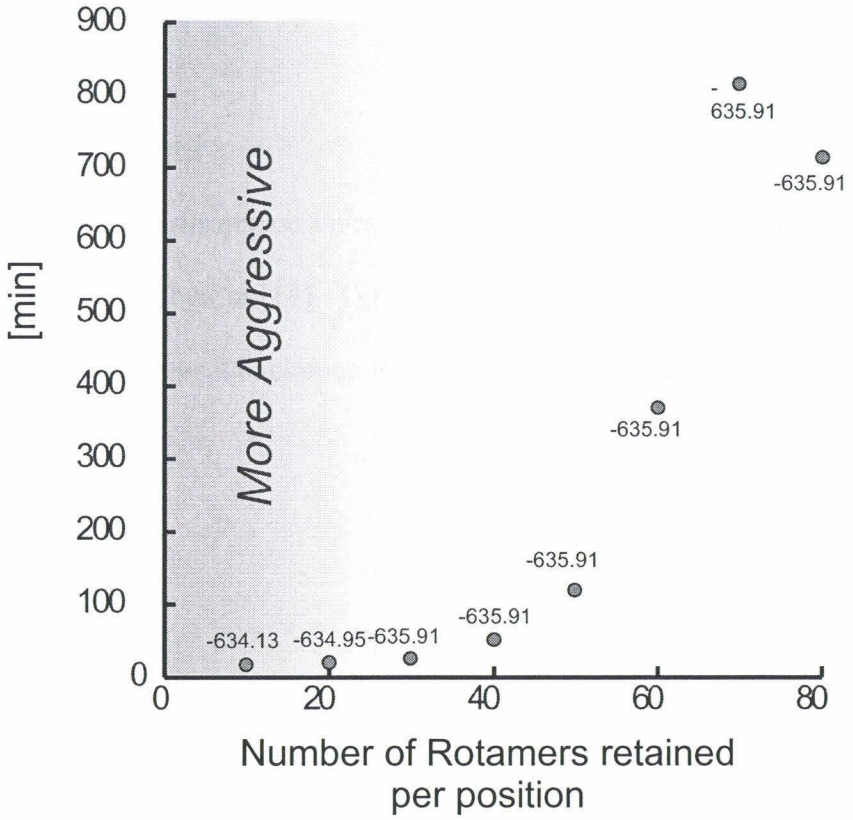
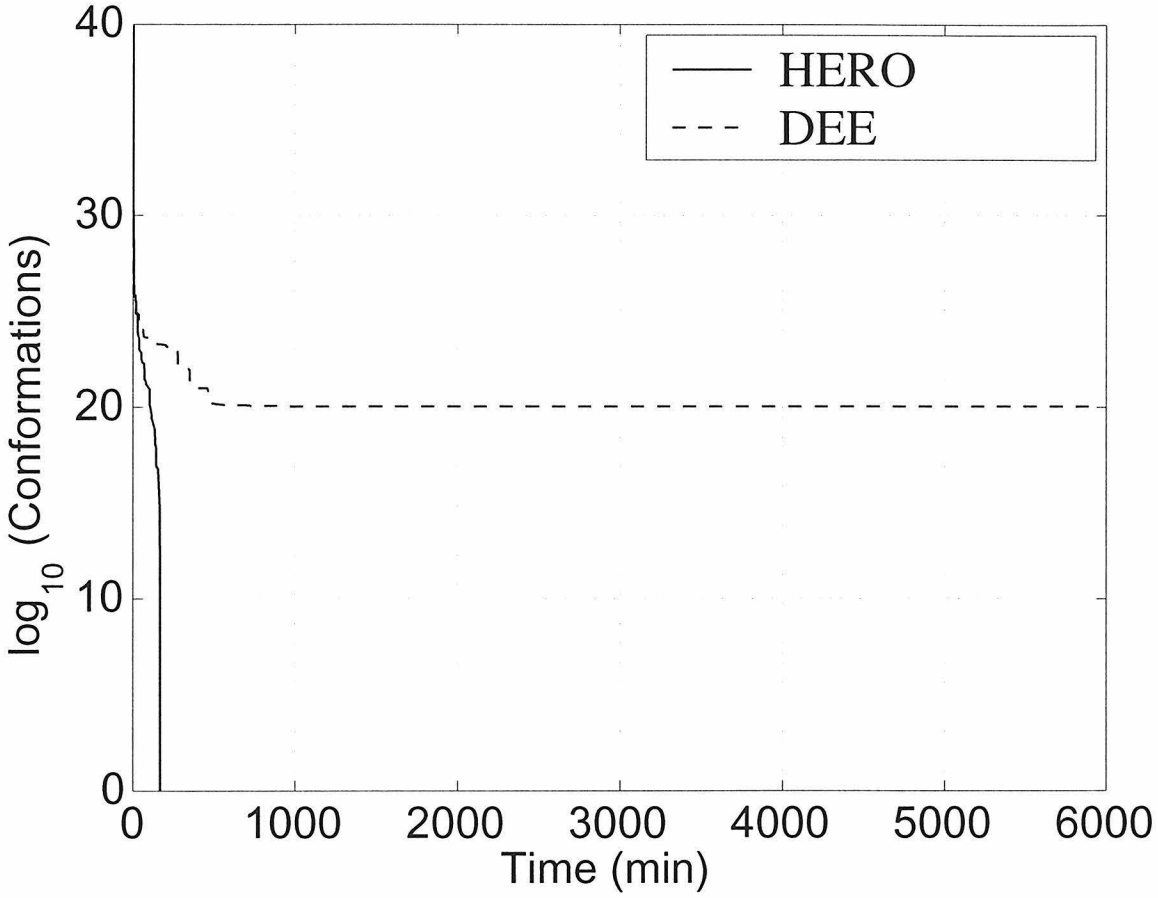

900
800
700
600
500
400
300
200
100
0
[min]
More Aggressive
-
635.91
-635.91
-635.91
-635.91
-635.91
-634.13
-634.95
-635.91
0
20
40
60
80
Number of Rotamers retained
per position

Figure VI-6: HERO performance on a problem insoluble by a highly optimized DEE implementation as described in [18]. Optimization was performed on 37 core residues of a designed leucine-rich repeat backbone (Courtesy Niles Pierce and Possu Huang).


HERO
DEE
40
30
20
10
0
0
1000
2000
3000
4000
5000
6000
log10 (Conformations)
Time (min)

Chapter VII

Combinatorial Optimization in Computational Protein Design

This chapter briefly summarizes general conclusions regarding the development and refinement of combinatorial algorithms for protein design.

The combinatorial optimization work presented in the previous chapters of this dissertation provides the basis for some general conclusions about the application of combinatorial algorithms to protein design problems. Such observations may be useful for the continuing development of search algorithms for protein design, and they may have extensions to other unsolved combinatorial search problems in biology, such as the protein structure prediction problem.

The first observation is that the performance of standard "off the shelf" algorithms may be enhanced by several orders of magnitude by tuning them to the class of optimization problems at hand. The fact that algorithms can be tuned for specific tasks is not a new finding, but refinement methods are difficult to generalize because they are heavily case-dependent. However, the techniques described here may be useful in inspiring insight into the refinement of future algorithms. It has been demonstrated in the previous chapters that refinement is accomplished in large part by imparting knowledge about the protein system to the algorithm. For example, the knowledge can take the form of heuristics for algorithmic decision-making, such as the metrics described in Chapter IV or the sorting factor described in Chapter V. The construction of these heuristics is based on trends in energetic distributions or problem structure observed in other protein design problems.

The second, almost self-evident observation is that problems of the same complexity may be very different in terms of optimization difficulty, as indicated by the time required for search. We believe that the disparities are due to differences in both the features of the landscapes and the organizations of the problems. However, there is a suggestion based on this work that the perceived difficulty of a problem instance is heavily algorithm dependent. While it is feasible that there exist instances of problems that are universally hard for all algorithms, for many problems, the choice of algorithm overshadows the intrinsic difficulty of the problem. For example, we have encountered cases in which a problem than cannot be solved by DEE can be solved by B&T. We have also found cases in which the situation is reversed.

The important corollary to this observation of algorithmic dependence is that different types of problems are best addressed by different types of algorithms. While it may be obvious that the corollary applies to optimization problems that are very different from one another, it is important to emphasize that it also applies to similar instances of problems within the same class. In the case of the protein design class of problems, instances that differ by as little as the protein secondary structure they represent are often best optimized by completely different algorithms.

As a consequence of the fact that closely related problems may require different algorithms, it is sometimes the case that over the course of its optimization, a single problem instance may go through phases that are best addressed by different algorithms. This phenomenon is best illustrated in the latter part of Chapter V, which describes how DEE and BAT algorithms may be applied to a problem in series. It remains an outstanding problem to find a way to determine *a priori* when an optimization problem is

best transplanted from one algorithm to another. Nevertheless, it has been repeatedly observed in our laboratory that a multifaceted algorithmic approach is the best way to address problems that are very difficult.

One way the ambiguity regarding the most effective application of different algorithms may be addressed is by merging optimization techniques at a deeper algorithmic level. The ideal would be to construct a "master" algorithm that would dispatch different optimization techniques as necessary to treat the problem appropriately at each stage of optimization. The successes of Hybrid Rotamer Optimization methods described in Chapter 6 validate the growing potential of such an approach.

Looking forward, the future prospects of combinatorial optimization in computational protein design will likely be based largely on the refinement methods and hybrid and serial approaches described here. In the short term, there may be significant opportunity to impart greater intelligence to the HERO algorithm in order to try to attain the "master" algorithm ideal. Regardless of the particular form of future algorithms, it will be interesting to see what new strategies for algorithmic improvement emerge, as will be discovering how the current strategies may be applied to other combinatorial problems in biology.

Appendix A

Experimental Analysis of Residues Interacting with β-turns

In the course of analyzing computed amino acid sequences for β-sheet surfaces, it was necessary to assess the importance of certain individual side-chain interactions. In particular, we suspected that amino acids involved in interactions with turns might have significant impact on the stability.

The four strands that comprise the β-sheet of protein G may be grouped into two pairs of β-strands, each connected by a β-turn. In the crystal structure, side-chains on the β-sheet surface interact with backbone atoms of the turn. In particular, Lys 13 forms a hydrogen bond with the carbonyl group of residue 10 in the first turn, and in the second turn, Asp 46 forms a hydrogen bond with the amide hydrogen of residue 48, and Thr 51 acts as both as a hydrogen bond donor and acceptor for the side chains at positions 49 and 50.

Side chain placement calculations did not always reproduce these native interactions. In particular, other acidic residues were selected for position 46, and the Lys at position 13 was often replaced by a non-similar amino acid, such as Glu. To assess the impact of these substitutions on the stability of the molecule, we performed mutagenesis at positions 8, 13, and 46, and measured the melting temperatures of the resulting mutants.

To examine the interaction involving Lys 13, two mutants were made, K13E and K13E-N8K. The double mutant was selected because it was produced by number of computations due to the predicted 8-13 salt-bridge interaction. The thermal stabilities of

the two mutants were 75 °C and 83 °C, respectively, demonstrating some recovery of thermal stability with the introduction of the residue with a compensating charge.

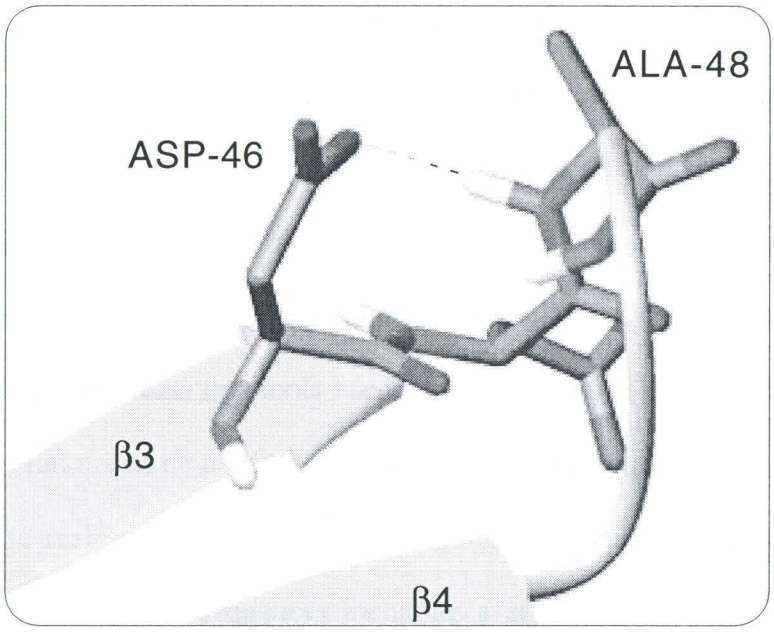
The native side chain-to-backbone interaction of Asp46 is illustrated in figure A-1. Three mutants were studied at this position: D46E, D46N, and D46Q. One would expect that any difference in stability between Asp and Asn would be evidence of the importance of charge on the side-chain, since the two amino acids may occupy the same conformations and form the same hydrogen bonds. Analogously, the D46E mutation probes the importance of the geometry of the interaction, since the overall charge and functionality are preserved, but the conformation must be altered to accommodate the extra methylene unit. Finally, the D46Q mutant provided a means to assess the additivity of the effects of charge and geometry.

The melting temperatures of the mutants at position 46 are shown in Table A-1. As anticipated, all mutations are destabilizing, but to different extents. It appears that while both the charge and geometry are both important, the difficulties associated with accommodating an extra methylene unit are slightly more significant. Also, the destabilizations appear to be roughly additive.

Table A-1. Thermal stabilities of mutations at position 46 of protein G (1pga).

Mutant	Tm [°C]
D (wt)	83
D46E	81
D46N	79
D46Q	71

Figure A-1: Wild-type turn interaction of Asp-46 with neighboring β-turn. Structure coordinates are taken from PDB entry 1pga.


ALA-48
ASP-46
β3
β4

Appendix B

Conversion of Computational Protein Design Tools for Z-score Optimization

Note: This appendix summarizes unpublished details of the implementation of Z-score optimization tools used to optimize sequences for solvent-exposed b-sheet residues. The results of this work are currently in press in Phys Rev. Lett.

It has been demonstrated in lattice models that it is possible to optimize an energy expression for protein design by tuning its parameters to maximize its disposition toward sequences known to be stable over random sequences. The preference of an energy expression toward a known stable sequence is referred to as the Z-score, and is measured in units of standard deviation from the average energy of all sequences. The wild-type sequence is used as the known stable sequence, and the average and standard deviation are computed by sampling a test set of random sequences.

Street et al. proposed that tools used for computational protein design with full-atom representation might be used to test if the Z-score technique may be extended from lattice models to real proteins. The most straightforward implementation would entail generating a list of random sequences including a sequence known to be stable, and performing a side chain placement calculation on each one. The calculation would entail computing the interaction matrix and using a combinatorial search algorithm to determine the global minimum configuration and the associated energy of each sequence. All the necessary statistical information could be readily assembled for tuning force field parameters once all the sequences have been evaluated in this way.

Unfortunately, the straightforward approach is prohibitively slow in practice, due mainly to two effects. First is the significant comprehensive time spent reading and

writing information to the computer disk. For each calculation, necessary input/output operations include loading a large rotamer library into memory, writing the energy matrix to disk, and loading the energy matrix back into memory for combinatorial optimization. The second problem is that the computation of the energy matrix itself can be slow, particularly when including a surface-area based solvation term. The total calculation time for a single eight-residue sequence, when performed this way, takes on the order of 10 minutes. Therefore, canvassing a thousand sequences requires approximately a week of computer time.

We recognized that a large part of this time is spent performing redundant operations. By making two types of changes to the computational methodology, we were able to reorganize the flow of information to reduce the redundancy. The resulting reduction in calculation time is dramatic.

The first change was to compute a single energy matrix that would contain all the necessary information for all the sequences being studied, thereby eliminating redundancy in two aspects of the calculation: Repeatedly loading the same rotamer library into memory, and computing the same interactions between side chains that are the same on different random sequences. The implementation consisted of two elements. The first element consisted of computing the energy matrix as would be done for a protein design problem, rather than a homology modeling problem, by allowing all amino acid types represented in the set of random sequences at every residue position. The second step was to modify the combinatorial optimization software to load the energy matrix once, and to then allow the repeated specification of sequences, for which the program would temporarily discard all the unnecessary elements of energy matrix.

Optimization of the Z-score of the wild-type sequence was performed by iterative modification of the energy expression. The second major change was directed at speeding up the time required to compute the interaction matrix for any energy expression by eliminating the need to recompute all interactions after every modification. The critical observation was that differences between energy expressions amount only to altered weightings of the potential energy terms; the unscaled energy contributions from different energy terms remain unchanged from iteration to iteration. It is therefore only necessary to compute the energy contributions of each the individual terms once. To make use of this observation, it was necessary to construct mechanisms to store the unscaled energy terms and assemble them back into a complete energy expression with an arbitrary set of weightings. This was implemented by storing a file for each term of the energy expression structured much like the overall interaction matrix, and by adding the ability to the optimization program to read a set of these files and assemble them according to parameters supplied at search time.

The unscaled energy terms were assembled into total interaction energies at the time of optimization using the following equation:

$$\begin{aligned} E(i) = {} & E_{vdw+HB}(i) && + k \times A_{np}^{exp}(i) \times s_{np} \\ & + A_{np}^{bur}(i) \times s_{np} && - A_{pol}^{bur}(i) \times s_{pol} \\ & + \frac{E_Q(i)}{e} && + 10.0^{\eta_{ss} S(i)} - 1.0 \\ & + \left(\frac{D_{HB}}{D_{8.0}} - 1 \right) E_{HB}(i) + \left(\frac{P_H}{P_{2.0}} - 1 \right) E_{PolHb}(i) \end{aligned} \qquad (1)$$

One noteworthy caveat concerns the hydrogen bond potential. To work around interdependence of the hydrogen-bond potential and the polar hydrogen burial term, the

hydrogen-bonding energy is explicitly included with the Van der Waals energy. It is scaled at the time of optimization by subtracting a correction term normalized by the default hydrogen bond well depth. The same treatment is applied to the polar hydrogen burial energy as well.

The form of the reassembly is the same for rotamer-rotamer energies.