Advances in computational protein design:

Development of more efficient search algorithms and their application to the full-sequence design of larger proteins

Thesis by

Geoffrey K. Hom

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2005

(Defended May 13, 2005)

This thesis is dedicated to my mother,

Christina Fong Hom,

and my late father,

Paul Francis Hom.

Acknowledgements

During my six years at Tech, I often felt I was leading multiple lives: the life my mom and relatives were familiar with, the life I had in Avery House, and the life of the Mayo lab. I did not mean for my stay to turn out this way, but I am grateful nonetheless. Each life has affected me differently, and in all cases I sensed a part of me growing up, helping to make me whole. Thank you to all the people below.

My mom, Christina Fong Hom, who always gave me encouragement, freedom, and benefit of the doubt. Mom would always sacrifice herself for my welfare. My sister, Mary Katherine Hom, who steadily sent me letters, postcards, and care packages even though I never returned the favor. Best sister a guy ever had.

My many relatives, who always remember more about me than I remember about them. Special thanks to the John Homs, Uncle Ron and Auntie Muriel, Uncle Al, Auntie Helen, the Paul Fongs, Auntie Beatrice, Lance Hom, Glenn Hom, Louis Hom, Candice Leigh, and, of course, Grandma Fong.

The many, many people associated with my life in Avery, especially: Abe Kuo, Alex Shim, Alice Coppock, Amos Anderson, Andreea Stuparu, Arjun Bansal, Arthur Tjia, Avery Council, Avery Social Team, Basit Khan, Brad Phillips, Brant Carlson, Caleb Lo, Carol Magun, Cathy Jurca, Cecile Lim, Chelsea Chang, Chris Wetzel, Claire Levaillant, Corinna Markenscoff-Zygourakis, Dave Rahmlow, Davin Maddox, Ekua Anane-Fenin, Elizabeth Reed, Erik Dreyer, Erik Measure, Francy Shu, Frank Lee, Grace Chuang, Habib Ahmad, Helen Claudio, Henry Shu, Jackie Peng, Jamal Rorie, Jason Minamora, Jason Yosinski, Jim Pugh, John Hatfield, John Sadowski, Jon Chang, Joseph Koo, Joy Rimchala, Kaisey Mandel, Katsu Tanabe, Kevin Bartz, Kevin Ko, Kevin Wang, Kewell Zheng, Kimberly Eilert, Kulsoom Hasan, the Libbrechts, Meaghan Paceley, Melisa Xin, Michael Cheung, Mona Sheikh, Moti Feingold, My-Ngoc Truong, Naboon Riddhiraksa, Neil Tiwari, Orion Crisafulli, Paul Lee, Peter Yi, Rafed Al-Huq, Ryan Witt, Sascha Schediwy, the Saris, Si-Ping Han, Stephanie Chan, Steve Pracko, the Stoltzes, Sue Friedman, Tom Mack, Tony Lee, Vidya Bhalodia, Wei Zhang, William Leblanc, Yun-hsueh Liu, and the Zmuidzinases.

Caltech staff, who were usually all smiles: Tom Mannion, Tim Chang, Margo Marshak, Maria De La Torre, Guille Robles, Jerry Houser, Jose Valles, Gilberto Rodriguez, Jeff Hechter, Linda Chappell, Alison Ross, and Mary Neary Morley.

Friends from even more double lives: Glenn Mondro, Ian McKenzie, Nicholas McGee, Redmond Urbino, Shantanu Sharma, Shan Ichiyanagi, Shay Chinn, Sander Granat, Jean Crowder, Michael Aono, Annabelle Young, Fred Serricchio, Stephen Tang, Richard Kekahuna, Bill Robinson, Nur Monson, Jessica Cheung, Robert Forster, Tyson Rogers, Xander Peterson, Eric Tang, Will Wright, and Tim Brengle.

The following people I am indebted to well beyond words: Amit Kenjale, Dan Zimmerman, Brian Schroeder, Jennifer Li, Kenneth Kuo, Qinzi Ji, Qing Liu, Lauren McHugh, Eduardo Garcia, Scott Wheeler, Bill and Molly Jones, Yuan Sun, Smaranda Marinescu, and Fiona Heung. Thank you for thinking well of me.

I thank God and my pastor, Ken Fong.

Mentors and friends who helped me get into graduate school: Alex Filippenko, Ed Buck, Isaac Pessah, and Marc Facciotti. Especially Peter Yau and Bob Glaeser, who seem to have the gift of foresight, both of research and character.

Friends from other labs, and others in the Mayo Lab: Josh Marcus, Alex Farley, Derek Hodgson, Ryan Austin, Terry Takahashi, Bill Ja, James Holton, Christine Ueda, Pavel Strop, Achilleas Frangakis, Daisuke Umeno, Edgardo Farinas, Chris Otey, Heidi Privett, Mary Devlin, Chris Balmaseda, Jennifer Keeffe, Julie Mayo, Andrea Crowfut, Takumi Koshiba, Barry Olafson, Sarah Hamilton, Cathy Sarisky, Ben Gordon, Dan Bolon, Caglar Tankirulu, Welison Floriano, Darryl Willick, Eun Jung Choi, Deepshikha Datta, Shira Jacobson-Rogers, Justin DuBois, Julia Shifman, Katie Miller, and Oscar Alvizo.

Professors Frances Arnold, Ray Deshaies, and Jackie Barton for letting me rotate in their labs. Professors Deshaies, Doug Rees, and Niles Pierce for being on my committee. I only regret not seeking your advice more often.

JJ Plecs, Len Thomas, Andy Yeh, James Holton, and Xavier Ambroggio, for teaching me at least some of the ropes of X-ray crystallography.

Ernie, for food. Gary Hathaway, for mass spectrometry help.

Funding sources for various projects I was on: HHMI, NIH, Ralph M. Parsons Foundation, DARPA, IBM, and the Gordon and Betty Moore Foundation. Thank you for your support, including your support of research in general.

My officemates, who were extremely accommodating: Tom Treynor, Peter Oelschlaeger, Joshua Klein, and Karin Crowhurst.

Scott Ross, who was always patient with my lack of NMR understanding. Marie Ary, who never lost her professionalism. Eric Zollars, who stayed with me when I went to the ER. Chris Voigt, who took me on *the* trip to Vegas. Christina Vizcarra, who plays a beautiful viola. Rhonda DiGiusto, who went out of her way to look for Pepsi iTunes caps for me. Ben Allen, whose knowledge of computers made my life more fulfilling. Robert Dirks, who taught me bridge, Civilization, and humility. Jessica Mao, who showed me strength of character. John Love, who taught me to be young.

Kyle J. Lassila, who is a better crystallographer than I will ever be. His discipline and courage will carry him forever.

Premal Shah, who was always willing to forgive the occasional heated argument, and who was a surprisingly good travel companion in Japan.

Cynthia Carlson, whose spirit will always inspire me. It would be wonderful if I could find a way to be even an iota as well-read, well-traveled, well-lived, hopeful, energetic, joyful, and faithful as she.

Possu Huang, who was always patient. Always Mac-faithful. Absorbs material like a sponge, but quick to admit mistakes. Hardworking yet helpful. An unsurprisingly good travel companion in Japan. Comes through in the clutch, when it really matters. I know it wasn't on purpose, Possu, but thanks for being here while I was here.

I apologize that I could not mention or remember all the kindnesses everyone has shown to me.

Finally, I thank my advisor, Steve Mayo. Steve gave me a chance when I thought all chances were gone. If he ever let success get to his head, he never really showed it. Never mean, even when stressed. I was constantly surprised by Steve's insight and his formidable breadth and depth of knowledge about *all* aspects of protein design, from computers to algorithms to molecular force fields to NMR to experimental techniques. Steve, I am emphatically grateful for the infinite opportunities you gave me, and for the tremendous variety and specificity of things I learned here. Thank you.

Abstract

Protein design is the art of choosing an amino acid sequence that will fold into a desired structure. Computational protein design aims to quantify and automate this process. In computational protein design, various metrics may be used to calculate an energy score for a sequence with respect to a desired protein structure. An ongoing challenge is to find the lowest-energy sequences from amongst the vast multitude of sequence possibilities. A variety of exact and approximate algorithms may be used in this search.

The work in this thesis focuses on the development and testing of four search algorithms. The first algorithm, HERO, is an exact algorithm, meaning that it will always find the lowest-energy sequence if the algorithm converges. We show that HERO is faster than other exact algorithms and converges on some previously intractable designs. The second algorithm, Vegas, is an approximate algorithm, meaning that it may not find the lowest-energy sequence. We show that, under certain conditions, Vegas finds the lowest-energy sequence in less time than HERO. The third algorithm, Monte Carlo, is an approximate algorithm that had been developed previously. We tested whether Monte Carlo was thorough enough to do a challenging computational design: the full-sequence design of a protein. Monte Carlo didn't find the lowest-energy sequence, although a similar sequence from Vegas folded into the desired structure. Several biophysical methods suggested that the Monte Carlo sequence should also fold into the desired structure. Nevertheless, the Monte Carlo structure as determined by X-ray crystallography was markedly different from the predicted structure. We attribute this discrepancy to the presence of a high concentration of dioxane in the crystallization conditions. The fourth algorithm, FC_FASTER, is an approximate algorithm for designs of fixed amino acid composition. Such designs may accelerate improvements to the physical model. We show that FC_FASTER finds lower-energy sequences and is faster than our current fixed-composition algorithm.

Table of Contents

List of Tables and Figures

Tables

Figures

Abbreviations

ANS	1-anilino-napthalene-8-sulfonate
CASP	Critical Assessment of Techniques for Protein Structure Prediction
CD	Circular Dichroism
DEE	Dead-End Elimination
DNA	Deoxyribonucleic Acid
ENH	Engrailed homeodomain
FASTER	Fast and Accurate Side-chain Topology and Energy Refinement
FC	Fixed amino acid composition
FC_FASTER	Fixed-composition FASTER
FC_MC	Fixed-composition Monte Carlo
FC_sPR	Fixed-composition, single-position perturbation/relaxation
ΔG_{unfold}	free energy of unfolding
Gβ1	β1 domain of protein G
GMEC	Global Minimum Energy Conformation
HERO	Hybrid Exact Rotamer Optimization
HETR	High-Energy Threshold Reduction
HEWL	Hen egg white lysozyme
HHMI	Howard Hughes Medical Institute
MC	Monte Carlo
NMR	Nuclear Magnetic Resonance
ORBIT	Optimization of Rotamers by Iterative Techniques
PDB	Protein Data Bank
r.m.s.d.	root mean square deviation
REM	Random Energy Model
SAD	Single Wavelength Anomalous Diffraction
SCMF	Self-Consistent Mean Field
SSRL	Stanford Synchrotron Radiation Laboratory
TFE	trifluoroethanol
T_m	melting temperature

Chapter I

Introduction

From proteins to computational protein design

Proteins are diverse, ubiquitous biological macromolecules. Hair, fingernails, skin, and even spider's silk are made up largely of fibrous proteins such as keratin, collagen, and silk fibroins. Many hormones, such as insulin and human growth hormone, are proteins. Antibodies, which help your immune system fight disease, are proteins. Hemoglobin, which transports oxygen in your blood, is a protein. Enzymes, which can speed up chemical reactions by more than a millionfold, are proteins. An example of an enzyme is DNA polymerase, which helps replicate your DNA.

The field of protein design seeks to tap the infinite potential of proteins. Modified or completely novel proteins could be used to change the curliness of your hair, to help your body fight specific diseases, or even to alter your DNA. There is also great potential for designed proteins in environmental and industrial applications, such as cleaning up oil spills, creating new fabrics, or manufacturing chemicals. To design proteins that will carry out our every whim, we need "only" to understand how proteins work.

While different proteins serve many different functions, all proteins are made up of the same components: amino acids. A protein is a polypeptide of amino acids that folds into a well-defined structure. The composition and order of the amino acids, i.e., the amino acid sequence, determine the structure of a protein. This structure, which includes both the polypeptide backbone and the conformations of the side chains of the amino acids, is the source of a protein's functional abilities. So, to a first approximation, a protein's sequence determines its structure, which determines its function. The challenge of taking a sequence and predicting its structure (and ultimately its function) is the infamous protein-folding problem.

To design proteins, we have to solve basically the inverse of the protein-folding problem **(Fig. I-1)**. It is important to note that only a few structural elements may be necessary for a protein's function. So while one sequence may determine one structure and one function, that same function may be encoded by multiple, slightly different structures, and thus by multiple sequences. Protein design is the art of specifying the desired structural elements and then choosing an amino acid sequence that will fold into a structure consistent with those elements. Computational protein design aims to quantify and automate this process.

For computational protein design to succeed, we must achieve mastery over two distinct problems. The first problem is to accurately simulate the protein and its physical environment. A physical model is used to calculate an energy score for an amino acid sequence with respect to a desired protein structure. The second problem is to efficiently find the lowest-energy sequences from amongst the vast multitude of sequence possibilities. A variety of exact and approximate algorithms may be used in this search. The validity of both the physical model and the search algorithms is largely unestablished until computationally designed sequences have had their structures verified experimentally. Because the structures are unlikely to be perfect, useful information can often be gleaned from the experimental results and used to determine shortcomings in the physical model or search algorithm. The addressing of these shortcomings marks the end of one cycle of computational protein design. A new and improved cycle may then begin.

Search complexity

The combinatorial complexity faced by a protein design search algorithm is immense. Consider the task of finding the lowest-energy sequence that will fold into a backbone structure that is 50 amino acids in length. How many different sequences are there? If each of the 20 naturally occurring amino acids can be at each of the 50 amino acid positions, then there are 20^{50}, or $\sim 10^{65}$, different sequences. Even if the energies of one trillion sequences could be calculated per second, it would take $\sim 10^{45}$ years (i.e., 1,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000 years) to calculate the energies of all the sequences. An exhaustive search is thus prohibitive, except for the simplest designs. For protein design to be effective, we need more efficient search algorithms.

Algorithms, chapter by chapter

The work in this thesis focuses on the development and testing of four search algorithms.

Chapter II describes the first algorithm, HERO. HERO is an exact algorithm, meaning that it will always find the lowest-energy sequence if the algorithm converges. We show that HERO is faster than other exact algorithms and converges on some previously intractable designs. Larger designs thus become more feasible.

Chapter III describes the second algorithm, Vegas. Vegas is an approximate, or inexact, algorithm. Inexact algorithms tend to be faster, but they may not find the lowest-energy sequence. We show that, under certain conditions, Vegas finds the lowest-energy sequence in less time than HERO.

Chapter IV examines the utility of the third algorithm, Monte Carlo. Monte Carlo is an approximate algorithm that was developed previously.[1,2] We tested whether Monte Carlo was thorough enough to do a challenging computational design: the full-sequence design of a small protein. This design specified a backbone structure 51 amino acids in length, and multiple amino acids were allowed at each position. Monte Carlo didn't find the lowest-energy sequence. However, Vegas found a similar, lower-energy sequence that was shown by NMR to be folded to the desired structure. Furthermore, several biophysical methods indicated that the Monte Carlo and Vegas molecules are nearly identical, suggesting that the Monte Carlo sequence should also fold into the desired structure.

Chapter V reveals the structure of the Monte Carlo sequence, as determined by X-ray crystallography. The crystal structure was markedly different from the predicted structure. We attribute this discrepancy to the high concentration of dioxane present in the crystallization conditions, as biophysical experiments showed that dioxane increases both the helicity and the oligomerization state of the designed protein.

Chapter VI describes the fourth algorithm, FC_FASTER. FC_FASTER is an inexact algorithm for designs of fixed amino acid composition. Fixed-composition designs may be useful for circumventing defects in the modeling of the denatured state of proteins, and thus FC_FASTER could accelerate improvements to the physical model. We show that FC_FASTER finds lower-energy sequences and is faster than our current fixed-composition algorithm.

Compared to its vast potential, computational protein design is still in its infancy. Typical designs do not address complex functions, large proteins, or protein complexes. The future of computational protein design will be in these areas, but we will need powerful new search algorithms to get us there.

References

1. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., and Teller, A.H. 1953. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics* **21:** 1087-1092.
2. Voigt, C.A., Gordon, D.B., and Mayo, S.L. 2000. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *J. Mol. Biol.* **299:** 789-803.

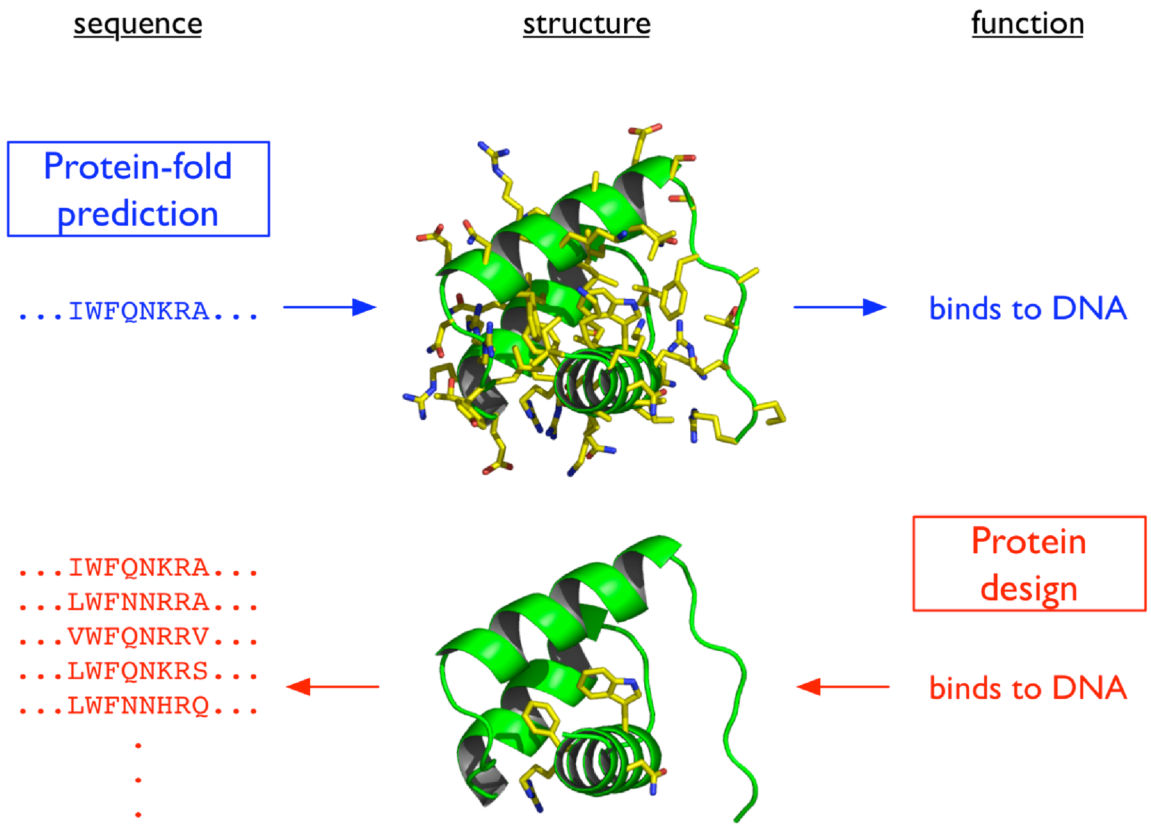

Figure I-1. Protein design is the inverse of protein-fold prediction. Protein-fold prediction is the art of predicting an amino acid sequence's structure (and ultimately its function). Protein design is the art of specifying the desired structural elements and then choosing a sequence that will fold into a structure consistent with those elements. The same function may be encoded by multiple sequences.

Chapter II

Exact rotamer optimization for protein design

The text of this chapter has been adapted from a published manuscript that was coauthored with D. Benjamin Gordon and Professors Stephen L. Mayo and Niles A. Pierce.

D. B. Gordon, G. K. Hom, S. L. Mayo and N. A. Pierce. 2003. *J. Comput. Chem.*, **24:** 232–243.

Abstract

Computational methods play a central role in the rational design of novel proteins. The present work describes a new hybrid exact rotamer optimization (HERO) method that builds on previous dead-end elimination algorithms to yield dramatic performance enhancements. Measured on experimentally validated physical models, these improvements make it possible to perform previously intractable designs of entire protein core, surface, or boundary regions. Computational demonstrations include a full core design of the variable domains of the light and heavy chains of catalytic antibody 48G7 FAB with 74 residues and 10^{128} conformations, a full core/boundary design of the β1 domain of protein G with 25 residues and 10^{53} conformations, and a full surface design of the β1 domain of protein G with 27 residues and 10^{60} conformations. In addition, a full sequence design of the β1 domain of protein G is used to demonstrate the strong dependence of algorithm performance on the exact form of the potential function and the

fidelity of the rotamer library. These results emphasize that search algorithm performance for protein design can only be meaningfully evaluated on physical models that have been subjected to experimental scrutiny. The new algorithm greatly facilitates ongoing efforts to engineer increasingly complex protein features.

Introduction

Advances in computational protein design have largely been paced by two factors: the development of biologically meaningful physical models for describing the design space, and the development of combinatorial optimization algorithms for searching this space over all allowed sequences and conformations. High-performance search algorithms make it possible to perform atomic-resolution side-chain placement calculations for the selection of novel amino acid sequences. The sequences can then be evaluated in the laboratory to validate and/or improve the physical model. Both the discrete rotamer libraries used to represent the possible side-chain conformations and the empirical potential function used to assess the quality of the possible design sequences are critical to the biological validity of the approach.

Several computational models have been experimentally validated for the design of protein cores,[1-5] and for the design of boundary and surface residues with varying degrees of solvent exposure.[6-8] However, there are still few examples of experimentally validated computational designs for complete protein domains.[9] In the context of protein design, physical model validation[2] is a challenging endeavor in which experimental assays of designed molecules are used to parameterize or enhance an existing model. A physical model becomes useful if it is able to identify sequences that fulfill the design requirements from amidst the astronomically large number of possible sequences. To improve the prospects for performing more ambitious designs including protein function, it is necessary to increase the efficiency of the computer algorithms while maintaining or even improving the physical models that form the basis for sequence selection.

Significant effort has been expended in developing both exact and approximate search algorithms for protein design.[10] Protein design has recently been shown to be *NP*-hard,[11] meaning that it joins a class of challenging combinatorial optimization problems for which no exact polynomial-time algorithms are known. Approximate algorithms that have been applied to protein design include Monte Carlo methods,[12,13] genetic algorithms,[1] and self-consistent mean field approaches.[14,15] These methods are computationally inexpensive, but their accuracy in identifying the global minimum energy conformation (GMEC) is known to degrade as problem size increases.[16] To avoid corrupting the potential function with experimental feedback based on incomplete searches, it is highly desirable to rely on exact search algorithms if effective exponential-time algorithms are available. Exact tree-based algorithms have been successfully applied to protein design.[17,18] For large design problems, methods based on the dead-end elimination (DEE)[19-26] theorem have emerged as the most successful.

It is important to note that search algorithm performance is strongly affected by the physical model on which the optimization is based. For example, DEE has been found to perform best when the number of rotamers per position is small, as evidenced by the relative ease in performing side-chain placement calculations for homology modeling studies on large proteins.[23,26] For design calculations, the number of rotamers at each position increases dramatically since multiple amino acid identities are represented at each position. This change increases both the cost of each iteration and the difficulty in reducing the combinatorial size of the problem. These effects are only exacerbated as the fidelity of the rotamer library is improved or as the number of design positions is increased. The physical context of the design positions influences algorithm performance significantly; confined core residues are generally much faster to design than less-constrained residues on the protein surface. This is because side chains packed into protein cores experience more physical restrictions, facilitating the identification of rotamers that do not belong to the GMEC. We have also observed that the precise implementation of the energy expression can have dramatic effects on the search speed. As is demonstrated later, alterations of the potential function can make seemingly intractable optimization problems trivial.

Because the performance of search algorithms depends strongly on factors in addition to sheer combinatorial complexity, it is critical to evaluate new improvements using potential functions and rotamer libraries that are meaningful in the context of protein design. Thus, optimization benchmarks are best performed on potential functions and rotamer libraries that have been subjected to experimental scrutiny, or, alternatively, benchmarks should be closely followed by experimental validation. At this time, there still remains a need to develop search algorithms that can perform large-scale optimizations on experimentally validated physical models.

The development of increasingly powerful dead-end elimination algorithms is specifically targeted at addressing this challenge. The basic idea of DEE is to eliminate rotamers from consideration that can be proven to be incompatible with the GMEC. In the context of side-chain placement for homology modeling, the original algorithm introduced criteria for eliminating individual rotamers and “flagging” dead-ending pairs of rotamers to facilitate the elimination of single rotamers during subsequent iterations.[19,20] The “unification” of rotamers at two or more positions into super-rotamers was subsequently introduced as an effective method for starting new cascades of eliminations.[21,22] Using Goldstein’s more powerful elimination criteria,[22] DEE methods were extended to protein design applications.[2] Metrics were subsequently developed to mitigate the added expense of these criteria, particularly for the flagging of dead-ending rotamer pairs.[24] Recently, more sophisticated elimination criteria were introduced based on the concept of conformational splitting.[25] An adaptive implementation of split DEE has since been described that reduces the cost of each iteration in the case of multiple splitting positions.[26] A further approach termed “generalized” DEE has been introduced,[26] although in our hands it does not yield a performance enhancement over existing methods.

The present work reports new ideas that extend the application of DEE algorithms to larger design regimes for all structural contexts: core, boundary, and surface. We have previously obtained large speed enhancements from optimizing dead-ending pairs calculations.[24] These improvements are effective because the majority of the overall calculation time is spent attempting to flag dead-ending pairs. To further reduce this time, we have focussed on exploring additional flagging methods. Two complementary approaches have resulted, each providing new and inexpensive ways to find dead-ending pairs. Taken together, and often independently of each other, these methods make previously intractable optimization problems solvable.

The first approach employs bounding criteria that were originally developed for use in tree-based optimization methods.[17,27] These bounding criteria are fundamentally different in nature from the dominance criteria that typify dead-end elimination. A bounding criterion eliminates a rotamer by comparing the lower-energy bound of possible sequences containing that rotamer to the total energy of a known reference sequence. On the other hand, DEE criteria are examples of dominance relations[28] that attempt to show that one rotamer is preferred over another in all circumstances. As with the DEE dominance criteria, bounding criteria may be used both to eliminate individual rotamers and to flag pairs of rotamers. Moreover, because "bound flags" are obtained by measures other than dominance, they have the potential to augment the DEE reductions and to enhance the performance of the algorithm. Bounding criteria require the energy of a reference sequence to which bounding energies may be compared. We therefore employ a stochastic Monte Carlo search to rapidly determine a valid reference energy. Interestingly, the algorithm remains exact, but it is no longer deterministic.

The second approach makes it possible to flag many dead-ending pairs at essentially no additional cost. These "split flags" are generated as a by-product of applying the conformational splitting criteria to eliminate single rotamers. By promoting further reduction in the combinatorial size of the problem prior to the application of expensive doubles criteria, these dead-ending pairs provide substantial computational savings.

The algorithm described in the present work combines three completely different search paradigms (dominance, bounding, and stochastic) into a single compatible approach. For ease of description, we term the new method hybrid exact rotamer optimization (HERO). Taken together, the two new strategies for flagging dead-ending pairs have dramatically increased the size of the design problems that can be attempted on a daily basis in the laboratory of one of the authors (S. L. Mayo). Results in the present work will demonstrate that exact search algorithms based on experimentally validated physical models are now able to tackle design problems that could previously be attempted only with approximate methods. In particular, it is frequently possible to perform full protein core, boundary, or surface designs with surprising efficiency.

Theory

Energy expression

Using a potential function described in terms of pairwise interactions, the total energy of the protein can be expressed as

$$E_{\text{total}} = E_{\text{template}} + \sum_i E(i_r) + \sum_i \sum_{j,j<i} E(i_r, j_u)\,, \qquad (1)$$

where E_{template} represents the self-energy of the backbone, $E(i_r)$ represents the energy of rotamer r at position i interacting with the backbone, and $E(i_r, j_u)$ represents the interaction energy between rotamers r and u at positions i and j, respectively. The objective of dead-end elimination criteria is to eliminate single rotamers that are dominated by other competing rotamers, and to flag dead-ending pairs of rotamers that are dominated by other competing rotamer pairs. Either of the rotamers in a dead-ending pair could still belong to the GMEC conformation, but they cannot appear together; this strengthens the possibility of eliminating rotamers during subsequent iterations. For notational convenience, a flagged dead-ending pair is said to belong to the set F.

Goldstein DEE

The Goldstein DEE criterion for single rotamers states that a rotamer i_r can be eliminated if there exists a competing rotamer i_t that satisfies

$$E(i_r) - E(i_t) + \sum_{j,j \neq i} \min_{\substack{u \\ (i_r, j_u) \notin F}} \left[E(i_r, j_u) - E(i_t, j_u) \right] > 0. \qquad (2)$$

In other words, i_r can be eliminated if the contribution to the total energy is always reduced by using an alternative rotamer i_t. Note that the minimum specifically excludes contributions from flagged (i_r, j_u) pairs, as these rotamers cannot coexist in the GMEC. If there are p residue positions and an average of n rotamers per position, the computational complexity of attempting to eliminate each rotamer during a round of Goldstein DEE is $O(n^3p^2)$, corresponding to loops of cost n over r, t, and u as well as loops of cost p over i and j.

The doubles version of this criterion[22] flags a rotamer pair (i_r, k_s) if there exists a competing pair (i_v, k_w) that satisfies

$$\begin{aligned} &[E(i_r) + E(k_s) + E(i_r,k_s)] - [E(i_v) + E(k_w) + E(i_v,k_w)] \\ &+ \sum_{j,j \neq i \neq k} \min_{\substack{u \\ (i_r,j_u) \notin F \\ (k_s,j_u) \notin F}} \left\{ \left[E(i_r,j_u) + E(k_s,j_u) \right] - \left[E(i_v,j_u) + E(k_w,j_u) \right] \right\} > 0. \qquad (3) \end{aligned}$$

For each of the rotamer pairs between two given positions, $O(n^2)$ comparisons are made with the other rotamer pairs at these positions. This criterion therefore makes $O(n^2)$ dominance checks in attempting to flag each rotamer pair. The computational complexity of Goldstein doubles is $O(n^5p^3)$, representing the most expensive component in most DEE implementations.

To obtain a subset of these flags at a lower cost, a “magic bullet” version of Goldstein doubles[24] was introduced that uses only one competing (i_v, k_w) pair to attempt to flag all other pairs of rotamers between positions *i* and *k*. The computational complexity is thus reduced to $O(n^3p^3)$, and only a single dominance check is made in attempting to flag each rotamer pair.

Split DEE

If no i_t rotamer dominates i_r for all possible conformations, then the Goldstein criterion will fail to make an elimination. Conceptually, however, i_r may still be eliminated if at least one (possibly varying) i_t rotamer dominates i_r for each conformation. Split DEE[25] embodies this idea by splitting the conformational space into partitions and checking to see if i_r is dominated by some i_t rotamer within each partition. In the simplest case (called "s = 1"), $O(n)$ partitions are created using the rotamers at a single splitting position. The rotamer i_r can then be eliminated if, for each splitting rotamer *v* at some splitting position *k*, there exists an i_t rotamer that dominates i_r within that partition:

$$E(i_r) - E(i_t) + \sum_{j,\, j\neq k\neq i} \left\{ \min_{\substack{u \\ (i_r, j_u)\notin F}} \left[E(i_r, j_u) - E(i_t, j_u)\right] \right\} + \left[E(i_r, k_v) - E(i_t, k_v)\right] > 0. \quad (4)$$

Domination in partition k_v is automatic if (i_r, k_v) is a flagged pair. The split DEE (s = 1) criterion is illustrated in **Figure II-1(a)**. The computational complexity of this approach remains $O(n^3p^2)$ despite the increase in elimination power.[25]

Increasing the number of splitting positions increases both the elimination power and the computational complexity. For two splitting positions (s = 2), there are $O(n^2)$ partitions and i_r may be eliminated if, for each pair of splitting rotamers k_v and h_w at splitting positions $k \neq h \neq i$, there exists an i_t rotamer that dominates i_r in that partition:

$$E(i_r) - E(i_t) + \sum_{j,\, j\neq i\neq h\neq k} \left\{ \min_{\substack{u \\ (i_r,\, j_u)\notin F}} \left[E(i_r,j_u) - E(i_t,j_u)\right] \right\} \tag{5}$$
$$+ \left[E(i_r,k_v) - E(i_t,k_v)\right] + \left[E(i_r,h_w) - E(i_t,h_w)\right] > 0.$$

Here, domination follows automatically if either (i_r, k_v) or (i_r, h_w) is a flagged pair. The application of this criterion is illustrated in **Figure II-1(b)**, where the rotamers at the second splitting position (h_w) effectively create sub-partitions of those created by the first splitting position (k_v). The computational complexity for (s = 2) split DEE is $O(n^4p^3)$.[25] Looger and Hellinga[26] present the same approach and provide the same complexity estimates in a later publication. With regard to implementation, Looger and Hellinga make the useful observation that conformational splitting may be coded adaptively so that sub-partitions at a new splitting level are explored only within those existing partitions that have failed to achieve dominance of i_r at the current level. This decreases the computational cost of an iteration relative to the worst-case complexity estimates.

Expressions for split DEE criteria and cost bounds for arbitrary numbers of splitting positions have been reported previously.[25] In practice, we rarely find it beneficial to use splitting criteria beyond s = 2. Split DEE criteria may be extended to flag pairs of rotamers exactly as for the Goldstein doubles criterion (3), with a corresponding increase in computational overhead relative to the singles implementation. However, we now pursue the following more interesting observation that flags can be generated during split singles calculations with no increase in computational complexity.

Split flags

Consider the scenario where i_r cannot be eliminated by split DEE (s = 1) because there are some partitions in which no i_t rotamer dominates i_r. It may still be possible to identify dead-ending pairs during the process of discovering this negative result. In those partitions k_v where i_r is dominated by some i_t, then the rotamer pairs (i_r, k_v) may be flagged as dead-ending. This concept is illustrated in **Figure II-1(c)**. The comparisons that are made in an effort to identify flags remain a subset of those made for a full Goldstein doubles calculation. $O(n)$ dominance comparisons are made in attempting to flag each rotamer pair. The complexity of split DEE is unaffected by this modification, remaining $O(n^3p^2)$ for ($s = 1$), so this approach compares very favorably with both full Goldstein doubles and magic bullet Goldstein doubles, as summarized in **Table II-1**.

Split flags may be generated with arbitrary numbers of splitting positions. The concept is illustrated for split DEE ($s = 2$) in **Figure II-1(d)**. In this case, the number of flagging comparisons remains $O(n)$ per rotamer pair but the iteration complexity increases to $O(n^4p^3)$. For a given maximum number of splitting positions, the attempt to eliminate a rotamer i_r has failed as soon as a sub-partition at the lowest level is encountered in which i_r is not dominated by some competitor. It is possible to continue checking dominance for other partitions to attempt to identify more flags, but for ($s \geq 2$), the mounting cost motivates our decision to branch out of an elimination attempt as soon as failure is assured. It appears that Looger and Hellinga[26] allude to a special case of this approach corresponding to ($s = 1$) split flags.

Bounding expressions

Bounding expressions provide an alternative means of determining whether a particular arrangement of rotamers at a subset of the residue positions can exist as part of the GMEC. Rather than eliminating rotamers by comparing them to other competing rotamers at the same positions, bounding expressions seek to produce a sharp lower bound on the total conformational energy given a certain subset of specified rotamers. If this bound is higher than the energy of some known complete reference sequence,

$$E_{\text{bound}}(\text{subset}) > E_{\text{total}}(\text{reference}), \qquad (6)$$

then the specified rotamers cannot coexist in the GMEC. The reference energy should be as low as possible, and may be obtained by a computationally inexpensive approximate search of the same rotamer conformation space.

There are many possible ways of constructing an expression to compute the lower energy bound for an arrangement of rotamers. The expression that yields the best performance in the branch-and-terminate algorithm[17] folds the one-body terms into the two-body terms:

$$E'(i_r, j_u) \equiv \frac{E(i_r) + E(j_u)}{2(p-1)} + \frac{E(i_r, j_u)}{2} \qquad (7)$$

and computes the lower bound on the total energy as

$$E_{\text{bound}} = \sum_{i \in C} \sum_{\substack{j \in C \\ j \neq i}} E'(i_r, j_u) + \sum_{i \in V} \min_r \left\{ 2 \sum_{j \in C} E'(i_r, j_u) + \sum_{j \in V} \min_u \left[E'(i_r, j_u) \right] \right\}. \ (8)$$

The set of residue positions C is the subset of "constrained" positions that are occupied by the rotamers under scrutiny, and the set V encompasses all the remaining "variable" residue positions. The more positions that are constrained, the sharper the bound becomes.

To use the bounding expression efficiently in the context of dead-end elimination, the set C may be considered to consist of a single rotamer, so that the lower bound on the energy of all conformations containing rotamer i_r is

$$E_{\text{bound}}(i_r) = \sum_{m,m\neq i} \min_{\substack{t \\ (i_r,m_t)\notin F}} \left\{ 2E'(i_r,m_t) + \sum_{j,j\neq m\neq i} \min_{\substack{u \\ (j_u,m_t)\notin F}} \left[E'(j_u,m_t) \right] \right\}. \qquad (9)$$

Using the implementation described previously,[17] where the innermost summation is precomputed, the complexity of computing the energy bound for each single rotamer is $O(n^2p^3)$. The more positions that are constrained, the sharper the bound becomes. Note that flagged dead-ending pairs can be excluded during the "min" operations.

Bounding flags

Increasing the constrained set C to encompass a pair of rotamers produces the bounding expression

$$\begin{aligned} E_{\text{bound}}(i_r,k_s) &= 2E'(i_r,k_s) \\ &+ \sum_{m,m\neq i\neq k} \min_{\substack{t \\ (i_r,m_t)\notin F \\ (k_s,m_t)\notin F}} \left\{ 2E'(i_r,m_t) + 2E'(k_s,m_t) + \sum_{j,j\neq m\neq i\neq k} \min_{\substack{u \\ (j_u,m_t)\notin F}} \left[E'(j_u,m_t) \right] \right\}. \end{aligned} \qquad (10)$$

The pair (i_r, k_s) can be flagged if $E_{\text{bound}}(i_r, k_s) > E_{\text{total}}(\text{reference})$ even if the pair is not dead-ending according to any known DEE criterion. The innermost summation is invariant with the rotamer indices r and s for a choice of positions i and k. By pre-computing this term independent of r and s, the computational complexity of bounding the total energy for each rotamer pair is $O(n^2p^4 + n^3p^3)$. Again, it is possible to take advantage of previously flagged pairs in computing the energy bounds.

The potential benefit of using this bounding expression is illustrated in **Figure II–2**, where E_{bound} is compared to E' for all the remaining unflagged rotamer pairs at one point during the convergence process for the core design of plastocyanin (described as

Case 1 in Methods). The performance of the bounds improves as residues are unified together to create super-rotamers representing larger fractions of conformational space.

Monte Carlo search

The efficacy of using bounding expressions to eliminate candidate rotamers and to flag rotamer pairs depends critically on the availability of a reference energy of a rotameric arrangement close in energy to the GMEC. This reference energy is obtained during the calculation using parallel Monte Carlo[29] searches from the current state of the conformational ensemble. The overall approach is therefore stochastic but exact, in the standard sense that if it converges, it converges to the GMEC.

Because Monte Carlo is repeated periodically as rotamers are eliminated, the searches are performed on a shrinking conformational space and the reference energy typically decreases as the calculation proceeds. By monitoring the top-ranked Monte Carlo sequences, it is possible to gain some insight into the convergence of the algorithm in sequence space prior to reaching full convergence. This can be particularly valuable for very large calculations that converge slowly or do not converge at all.

Unification

Dominance and bounding criteria can often benefit from residue unification, in which a "super-residue" is constructed from the rotamer pairs at two residue positions. The super-residue is treated as a single residue for the remainder of the calculation, and may be unified with other residues at a later iteration. Because flagged pairs that are unified can be eliminated, unification is performed on the pair of residues that have the largest fraction of dead-ending rotamer pairs, provided that the resulting super-residue has fewer than some maximum number of super-rotamers [typically $(np)_{\max} = 10^4$].

Algorithm schedule

The criteria described above may be coupled in many different ways. Our preferred strategy is to develop a standard schedule that performs well for a variety of design problems to minimize the need for user intervention. The entire iterative process is guaranteed to converge given sufficient time and computer memory. In practice, convergence is only possible if the elimination and flagging criteria prune the size of the combinatorial problem sufficiently rapidly to remain within the bounds of a human attention span and available computer memory.

Our preferred HERO implementation is described in **Figure II-3**. The Goldstein singles criterion is applied iteratively until no further rotamers are eliminated. The split (s = 1) criterion is then applied iteratively until no further eliminations are found. Split flags are generated during this process with no increase in the computational complexity of the original split implementation. Split criteria are then applied with multiple splitting positions [to the desired partition depth ($s \geq 2$)] once for each rotamer. A magic bullet metric may be employed to select the splitting partitions that are deemed most likely to produce flags or an elimination.[25] Magic bullet Goldstein doubles is then applied once to each rotamer pair to generate flags. The singles-elimination and split-flagging process is then repeated taking advantage of these new flags. On the second time through the cycle, a Monte Carlo search is performed to attempt to reduce the reference energy used to inform the bounding criteria. (Initially, the reference energy is set to be an arbitrarily large number.) The doubles bounding criterion is then applied once to each rotamer pair to identify more flags. After another round of singles eliminations and split flagging, a full round of Goldstein doubles flagging is performed using "q_{rs}" and "q_{uv}" metrics[24] to enhance performance. Following a fourth and final singles-elimination and split-flagging phase, unification is performed in lieu of a doubles calculation and the entire process is repeated.

For purposes of this study, we perform split DEE only up to two splitting positions ($s = 2$). For historical purposes, we include results using a previously published[25] magic bullet ranking metric (DEE $s2_{mb}$) that selects the two splitting positions that appear most likely to facilitate the elimination of rotamer i_r. The current baseline scheme for demonstrating the advancements of the present work is (DEE s2) without split flagging or bound flagging. To demonstrate the role that bound flagging and split flagging play for protein design calculations, these components are introduced separately to produce the schemes (DEE s2 bound flags) and (DEE s2 split flags). The complete hybrid exact rotamer optimization method described above is then termed HERO, which in longhand would be the less wieldy (DEE s2 bound & split flags).

Results and discussion

Benchmark design calculations

The protein design benchmarks described in this work are performed using a potential function and rotamer libraries that have been subjected to extensive laboratory testing.[2,5-7,9,30-37] This is an important consideration when assessing the significance of computational demonstrations. In particular, it is trivial to dramatically improve apparent search algorithm performance either by reducing the size of the rotamer library or by modifying the potential function. Such modifications would require laboratory validation before the resulting increase in algorithm efficiency could be considered to have significance to the field of protein design.

The performance enhancements provided by bound flags and split flags in the context of an experimentally validated physical model are demonstrated by the five problems described in **Table II-2**. These design cases arose during computational and experimental studies in the lab of one of the authors (S. L. Mayo). The conformational sizes in **Table II-2** are based on the rotamers that remain after high-energy threshold reduction (HETR)[23] is used to eliminate rotamers that clash with the backbone [for these tests, we removed rotamers with $E(i_r) > 20$ kcal/mol]. This practice reduces the risk of inflating the apparent conformational size of the problem using a large number of rotamers that are incompatible with the protein fold.

Case 1 represents a full core design of plastocyanin.[38] Case 2 is an unusual design problem involving all core positions on a novel repeating backbone based on the leucine-rich-repeat motif;[39] the residues in each of two repeats are restricted to have linked (but unspecified) amino acid identities. Case 3 represents the full core design of the variable domains of the light and heavy chains of catalytic antibody 48G7 FAB.[40] Case 4 is a full core and boundary design of the β1 domain of protein G,[41] and Case 5 is a full surface design of the same domain.

Timing results for the five benchmark design cases are described in **Table II-3** and displayed graphically in **Figure II-4**. Failure to converge implies that the unification process cannot continue without exceeding the specified maximum number of rotamers [we use $(np)_{\max} = 10^4$ for Cases 1, 2, 4, and 5; we use $(np)_{\max} = 2 \times 10^4$ for the larger conformational space of Case 3]. For the plastocyanin core design of Case 1, the previously published method (DEE $s2_{mb}$) fails to converge, leaving over 10^{14} conformations after 334 min. The current baseline scheme (DEE s2) also fails to converge, requiring 150 min to narrow the search space to 10^{11} conformations. This improvement is due both to the additional eliminations produced by full (s = 2) split DEE (as compared to the magic bullet version), and to the time savings yielded by the adaptive implementation of this approach.[26] Introducing bound flags gives full convergence to the GMEC in 22 min, while split flags give full convergence in 46 min. The combined approach (HERO) reaches convergence in 13 min.

Case 2 is unusual because the number of rotamers is not large and yet the case is challenging. This is evidently a product of the linking of amino acid identities across the repeating sub-units of the design. The algorithm converges only when using bound flags, requiring 23 min for (DEE s2 bound flags) and 7 min for HERO.

Case 3 is a large core design that converges with all schemes except the previously published method (DEE $s2_{mb}$), requiring 299 min for (DEE s2 split flags) and 359 min for HERO. Evidently, the bound flags do not play a substantial role for this problem and their calculation is effectively a computational overhead that accounts for the increase in time.

Case 4 is a full core/boundary design that fails to converge with any algorithm except HERO, which converges in 476 min. Case 5 is a full surface design of the same protein; it converges with all but (DEE $s2_{mb}$), with both bound flags and split flags yielding improvements, and HERO converging fastest in 35 min.

Performance of "Generalized" DEE

"Generalized DEE" was introduced[26] as another method for eliminating rotamers that cannot be eliminated by Goldstein DEE. The idea is to reoptimize a portion of the conformational background, taking advantage of flags between the reoptimized positions to increase the disparity in the net energy contributions of the i_r and i_t rotamers with these positions. The method is dominated by conformational splitting in the sense that for the same number of generalized positions g or splitting positions s, the eliminations obtained by generalized DEE are a subset of those obtained by split DEE. However, generalized DEE is more amenable to less costly implementations than split DEE, so it is possible that performance enhancements might still be achieved. Unfortunately, in our hands, this has not been observed, as illustrated in **Figure II-5** for a subset of 14 surface positions from benchmark Case 5. This smaller case was chosen to allow all of the generalized variants to run to completion. Generalized DEE was performed starting from the baseline scheme (DEE s2) with the maximum number of reoptimized positions corresponding to (g = 2, 3, 4, 5). For this example, the algorithm performance decreases monotonically with increasing g.

Physical model dependence

As is apparent from eqs. (1) and (2), the performance of any DEE algorithm will depend heavily on the nature of the physical model used to compute the one- and two-body terms [$E(i_\mathrm{r})$ and $E(i_r, j_u)$, respectively in eq. (1)]. Potential functions that emphasize energy terms that contribute to $E(i_r)$ relative to $E(i_r, j_u)$ will result in less coupling and easier optimization. In the limit of $\left|E(i_r)\right| >> \left|E(i_r, j_u)\right|$, the optimization reduces to the selection of the rotamer with the best one-body energy at each residue position. This observation emphasizes the importance of developing (and comparing) optimization schemes that are based on validated physical models—construction of inappropriate physical models can easily lead to impressive optimization performance.

A demonstration of the dependence of optimization performance on the underlying physical model is shown in **Figure II-6**. This case is a full sequence design of the 56 positions in the β1 domain of protein G. Three of these positions are preset to glycine (position 38 has a positive phi angle and functions as a C-cap for the alpha helix; positions 9 and 41 are sterically constrained core positions). The remaining positions are divided into core, boundary, and surface regions with the allowed amino acid identities at each of the 53 positions constrained to preserve the binary pattern of the wild-type sequence.[35] The resulting combinatorial complexity is 10^{112} conformations with 7775 initial rotamers after applying HETR[23] to eliminate rotamers that clash with the backbone. A HERO run with our "standard" potential function and rotamer library fails to converge after more than 1000 min. Optimization with a potential function modified to emphasize one-body terms reaches the GMEC in 20 min. The potential function modifications include (in order of decreasing importance): use of a one-body atomic solvation potential;[42] use of a Coulombic potential with a non-distance-dependent dielectric constant for rotamer/backbone interactions and a distance-dependent dielectric constant for rotamer/rotamer interactions;[43] use of rotamer internal strain energy; use of secondary structure propensities for helical and β-strand positions;[6] and, use of normalized van der Waals energies to remove the bias for selection of large amino acids. The validity of these modifications remains to be determined.

In addition to the potential function component of the physical model, great care must be taken with respect to the rotamer library. Previous computational work using surprisingly small rotamer libraries (approximately 67 rotamers per residue position) showed large, full-sequence design problems to be tractable.[26] For the full-sequence design of protein G described above, the average number of rotamers per residue position is 147. Using an unexpanded rotamer library and aggressive HETR, the average number of rotamers per position can be reduced to 70 (10^{80} conformations for 3705 rotamers). Obtaining the GMEC for the resulting problem using the standard potential function requires 28 min.

These results strikingly illustrate the dependence of algorithm performance on both the potential function and the rotamer library. Clearly, search algorithm performance cannot be meaningfully ascertained on models of uncertain biological validity. On the other hand, the development of biologically valid, one-body-weighted physical models provides an opportunity to tame the combinatorial beast that is at the root of computational protein design.[11]

Approximate alternatives

It is apparent from **Figure II-2** that bounding energies are a better indicator than self-energies of the likelihood that certain rotamers are not members of the GMEC. Based on this observation, we have observed that it is sometimes possible to find the GMEC in a few minutes using an approximate version of HERO in which bounding energies are used as a substitute for self-energies when applying HETR[23] to eliminate rotamers (data not shown).

Conclusion

Existing DEE algorithms spend most of their time attempting to flag dead-ending pairs of rotamers to facilitate future eliminations of dead-ending single rotamers. Two new methods have been formulated for efficiently identifying pairs of rotamers that are incompatible with the GMEC. One approach builds on split DEE methods to flag dead-ending pairs during the singles elimination process at essentially no additional expense. The other approach uses bounding criteria to flag pairs of rotamers for which a lower bound on the total conformational energy exceeds the energy of a reference conformation that has been identified by a computationally inexpensive Monte Carlo search. These bound flags would not necessarily be identified as dead-ending by any known DEE criterion. The new hybrid algorithm thus combines dominance criteria, bounding criteria and a stochastic search into a single compatible framework that is exact but no longer deterministic.

The present benchmark calculations and our ongoing experience with these algorithms suggest that the most reliable performance is achieved using the HERO algorithm that combines previous work on dead-end elimination with both new strategies for flagging pairs. This unified approach facilitates the daily optimization of protein design cases that were previously intractable using available computational resources.

As illustrated by our full-sequence design example, care must be taken to ensure that algorithmic performance benchmarks are biologically meaningful. An unbiased evaluation process that mimics the invaluable role that CASP[44] has played for the protein structure–prediction community could similarly aid the development and evaluation of computational protein design algorithms. Comparisons should evaluate two features of protein design methods: search efficiency on test cases based on a validated physical model, and design quality based on new physical models submitted by the contributors.

Methods

Physical model

The potential function has been previously described,[2,9,45-47] and incorporates terms for van der Waals interactions, hydrogen bonds, electrostatic interactions, and solvation. The van der Waals term is based on a Lennard-Jones 12-6 form with scaled atomic radii to promote overpacking in the protein core;[30] the hydrogen bond potential is a distance-dependent term based on a similar 12-10 form but attenuated by an angle-dependent term to enforce reasonable geometry;[6] electrostatic interactions are modeled using Coulomb's law with a distance-dependent dielectric;[46] solvation effects are modeled using approximate pairwise surface area decompositions to reward and penalize buried and exposed nonpolar surface areas, respectively[45] (this term is not computed for surface positions due to a lack of appropriate experimental data with which to parameterize the scaling factor); an additional solvation term penalizes polar hydrogen burial.[6]

The backbone-dependent rotamer libraries are based on the mean values from the Dunbrack and Karplus library[48] with expansion of the χ_1 and χ_2 angles for the aromatic residues, the χ_1 angle for hydrophobic residues, and no expansion for polar residues. Canonical values of the χ_3 and χ_4 angles are used for amino acids E, Q, K, and R. Residues are classified into core, boundary, or surface positions by an automated algorithm.[47] Core residue identities are selected from among the amino acids A, V, L, I, F, Y, and W, while surface residue identities are selected from among A, S, T, D, N, H, E, Q, K, and R. Boundary residue identities are chosen from the union of these sets.

Benchmark design cases

Case 1 represents the design of all 25 nonglycine residues (5, 14, 21, 27, 29, 31, 35, 37, 38, 39, 41, 46, 50, 55, 56, 63, 70, 72, 74, 80, 82, 84, 92, 96, 98) in the core of plastocyanin (PDB code 2pcy).[38] Case 2 involves all 34 core positions on a novel repeating backbone based on the leucine-rich-repeat motif;[39] the 17 residues in each of two repeats have linked (but unspecified) amino acid identities. Case 3 represents the full core design of the variable domains of the light and heavy chains of catalytic antibody 48G7 FAB (PDB code 1gaf).[40] This corresponds to residues (2, 4, 6, 19, 21, 25, 29, 33, 36-38, 44, 46-48, 55, 58, 62, 71, 73, 75, 78, 82, 84-87, 89, 90, 95-98, 102, 104) of chain L and residues (4, 6, 18, 20, 24, 32, 34-39, 45, 47, 48, 50, 51, 53, 61, 64, 68, 70, 72, 77, 79, 81, 83, 86, 90, 92-95, 97, 98, 103, 104, 108, 110) of chain H. Case 4 involves the design of all 10 nonglycine core residues (3, 5, 7, 20, 26, 30, 34, 39, 52, 54) and all 15 boundary residues (1, 11, 12, 16, 18, 23, 25, 27, 29, 33, 37, 43, 45, 50, 56) of the β1 domain of protein G (PDB code 1pga).[41] Case 5 represents the design of all 27 nonglycine surface residues (2, 4, 6, 8, 10, 13, 15, 17, 19, 21, 22, 24, 28, 31, 32, 35, 36, 40, 42, 44, 46, 47, 48, 49, 51, 53, 55) of the β1 domain of protein G. The benchmark calculations were performed on 16 Power3 processors of an IBM SP3 running at 375 MHz.

Acknowledgments

This work was supported by the Helen G. and Arthur McCallum Foundation (D. B. Gordon); an NRSA predoctoral training grant 5T32-GM07616 (G. K. Hom); the Howard Hughes Medical Institute (S. L. Mayo); the Burroughs-Wellcome Foundation and the Caltech Initiative in Computational Molecular Biology (N. A. Pierce).

References

1. Desjarlais, J.R., and Handel, T.M. 1995. De novo design of the hydrophobic cores of proteins. *Protein Science* **4:** 2006-2018.
2. Dahiyat, B.I., and Mayo, S.L. 1996. Protein design automation. *Protein Science* **5:** 895-903.
3. Lazar, G.A., Desjarlais, J.R., and Handel, T.M. 1997. *De novo* design of the hydrophobic core of ubiquitin. *Protein Science* **6:** 1167-1178.
4. Harbury, P.B., Plecs, J.J., Tidor, B., Alber, T., and Kim, P.S. 1998. High-resolution protein design with backbone freedom. *Science* **282:** 1462-1467.
5. Shimaoka, M., Shifman, J.M., Jing, H., Takagi, J., Mayo, S.L., and Springer, T.A. 2000. Computational design of an integrin I domain stabilized in the open high affinity conformation. *Nature Structural Biology* **7:** 674-678.
6. Dahiyat, B.I., Gordon, D.B., and Mayo, S.L. 1997. Automated design of the surface positions of protein helices. *Protein Science* **6:** 1333-1337.
7. Malakauskas, S.M., and Mayo, S.L. 1998. Design, structure and stability of a hyperthermophilic protein variant. *Nature Structural Biology* **5:** 470-475.

8. Street, A.G., Datta, D., Gordon, D.B., and Mayo, S.L. 2000. Designing protein beta-sheet surfaces by z-score optimization. *Physical Review Letters* **84:** 5010-5013.

9. Dahiyat, B.I., and Mayo, S.L. 1997. De novo protein design: fully automated sequence selection. *Science* **278:** 82-87.

10. Desjarlais, J.R., and Clarke, N.D. 1998. Computer search algorithms in protein modification and design. *Current Opinion in Structural Biology* **8:** 471-475.

11. Pierce, N.A., and Winfree, E. 2002. Protein design is NP-hard. *Protein Eng* **15:** 779-782.

12. 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.

13. Hellinga, H.W., and Richards, F.M. 1994. Optimal sequence selection in proteins of known structure by simulated evolution. *Proceedings of the National Academy of Sciences USA* **91:** 5803-5807.

14. Koehl, P., and Delarue, M. 1994. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *Journal of Molecular Biology* **239:** 249-275.

15. Lee, C. 1994. Predicting protein mutant energetics by self-consistent ensemble optimization. *Journal of Molecular Biology* **236:** 918-939.

16. Voigt, C.A., Gordon, D.B., and Mayo, S.L. 2000. Trading accuracy for speed: a quantitative comparison of search algorithms in protein sequence design. *Journal of Molecular Biology* **299:** 789-803.

17. Gordon, D.B., and Mayo, S.L. 1999. Branch-and-terminate: a combinatorial optimization algorithm for protein design. *Structure* **7:** 1089-1098.

18. Wernisch, L., Hery, S., and Wodak, S.J. 2000. Automatic protein design with all atom force-fields by exact and heuristic optimization. *Journal of Molecular Biology* **301:** 713-736.

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

20. 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. *Protein Engineering* **6:** 717-722.

21. Desmet, J., De Maeyer, M., and Lasters, I. 1994. In *The protein folding problem and tertiary structure prediction*. (ed. K.a.L.G. Merz Jr., S. eds), pp. 307. Birkhauser, Boston.

22. Goldstein, R.F. 1994. Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophysical Journal* **66:** 1335-1340.

23. DeMaeyer, M., Desmet, J., and Lasters, I. 1997. All in one: A highly detailed rotamer library improves both accuracy and speed in the modelling of sidechains by dead-end elimination. *Folding and Design* **2:** 53-66.

24. Gordon, D.B., and Mayo, S.L. 1998. Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *Journal of Computational Chemistry* **19:** 1505-1514.

25. Pierce, N.A., Spriet, J.A., Desmet, J., and Mayo, S.L. 2000. Conformational splitting: a more powerful criterion for dead-end elimination. *Journal of Computational Chemistry* **21:** 999-1009.

26. Looger, L.L., and Hellinga, H.W. 2001. Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: implications for protein design and structural genomics. *Journal of Molecular Biology* **307:** 429-445.

27. Leach, A.R., and Lemon, A.P. 1998. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. *Proteins* **33:** 227-239.

28. Papadimitriou, C.H., and Steiglitz, K. 1982. *Combinatorial optimization: algorithms and complexity*. Prentice Hall, New Jersey.

29. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., and Teller, A.H. 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics* **21:** 1087-1092.

30. Dahiyat, B.I., and Mayo, S.L. 1997. Probing the role of packing specificity in protein design. *Proceedings of the National Academy of Sciences USA* **94:** 10172-10177.

31. Bolon, D.N., and Mayo, S.L. 2001. Enzyme-like proteins by computational design. *Proc Natl Acad Sci U S A* **98:** 14274-14279.

32. Strop, P., and Mayo, S.L. 1999. Rubredoxin variant folds without iron. *Journal of the American Chemical Society* **121:** 2341-2345.

33. Sarisky, C.A., and Mayo, S.L. 2001. The ββα fold: explorations in sequence space. *Journal of Molecular Biology* **307:** 1411-1418.

34. Strop, P., Marinescu, A.M., and Mayo, S.L. 2000. Structure of a protein G helix variant suggests the importance of helix propensity and helix dipole interactions in protein design. *Protein Science* **9:** 1391-1394.

35. Marshall, S.A., and Mayo, S.L. 2001. Achieving stability and conformational specificity in designed proteins via binary patterning. *Journal of Molecular Biology* **305:** 619-631.

36. Ross, S.A., Sarisky, C.A., Su, A., and Mayo, S.L. 2001. Designed protein G core variants fold to native-like structures: sequence selection by ORBIT tolerates variation in backbone specification. *Protein Science* **10:** 450-454.

37. Su, A., and Mayo, S.L. 1997. Coupling backbone flexibility and amino acid sequence selection in protein design. *Protein Science* **6:** 1701-1707.

38. Garrett, T.P., Clingeleffer, D.J., Guss, J.M., Rogers, S.J., and Freeman, H.C. 1984. The crystal structure of poplar apoplastocyanin at 1.8-A resolution. The geometry of the copper-binding site is created by the polypeptide. *Journal of Biological Chemistry* **259:** 2822-2825.

39. Kobe, B., and Deisenhofer, J. 1996. Mechanism of ribonuclease inhibition by ribonuclease inhibitor protein based on the crystal structure of its complex with ribonuclease A. *Journal of Molecular Biology* **264:** 1028-1043.

40. Patten, P.A., Gray, N.S., Yang, P.L., Marks, C.B., Wedemayer, G.J., Boniface, J.J., Stevens, R.C., and Schultz, P.G. 1996. The immunological evolution of catalysis. *Science* **271:** 1086-1091.

41. Gallagher, T., Alexander, P., Bryan, P., and Gilliland, G.L. 1994. Two crystal structures of the B1 immunoglobulin-binding domain of streptococcal protein G and comparison with NMR. *Biochemistry* **33:** 4721-4729.

42. Shah, P., and Mayo, S.L. 2001. Personal communication.

43. Marshall, S.A., and Mayo, S.L. 2001. Personal communication.

44. Venclovas, C., Zemla, A., Fidelis, K., and Moult, J. 1999. Some measures of comparative performance in the three CASPs. *Proteins* **Suppl:** 231-237.

45. Street, A.G., and Mayo, S.L. 1998. Pairwise calculation of protein solvent-accessible surface areas. *Folding and Design* **3:** 253-258.

46. Gordon, D.B., Marshall, S.A., and Mayo, S.L. 1999. Energy functions for protein design. *Current Opinion in Structural Biology* **9:** 509-513.

47. Street, A.G., and Mayo, S.L. 1999. Computational protein design. *Structure with Folding and Design* **7:** R105-R109.

48. Dunbrack, R.L., Jr., and Karplus, M. 1993. Backbone-dependent rotamer library for proteins. Application to side-chain prediction. *Journal of Molecular Biology* **230:** 543-574.

Table II-1. Cost comparison of different flagging approaches.

Method	Iteration complexity	Flag attempts per rotamer pair
Full Goldstein doubles	$O(n^5p^3)$	$O(n^2)$
Magic bullet Goldstein doubles	$O(n^3p^3)$	1
Split flags (s = 1)	$O(n^3p^2)$	$O(n)$
Split flags (s = 2)	$O(n^4p^3)$	$O(n)$

Table II-2. Benchmark design cases.

Case	Description	Type	Residues	Rotamers	Conformations
1	Plastocyanin	Core	25	1716	1.7×10^{38}
2	Novel backbone	Linked core	34	674	8.4×10^{39}
3	Catalytic antibody	Core	75	4919	4.7×10^{128}
4	β1 of Protein G	Core/boundary	25	4295	4.0×10^{53}
5	β1 of Protein G	Surface	27	4842	4.9×10^{60}

Table II-3. CPU times for benchmark design cases running on 16 processors of an IBM SP3.

Case	Method	Time (min)	Remaining conformations
1	DEE s2mb	334	7 x 10^{14}
	DEE s2	150	2 x 10^{11}
	DEE s2 bound flags	22	1
	DEE s2 split flags	46	1
	HERO	13	1
2	DEE s2mb	250	1 x 10^{18}
	DEE s2	210	1 x 10^{18}
	DEE s2 bound flags	23	1
	DEE s2 split flags	167	3 x 10^{16}
	HERO	7	1
3	DEE s2mb	984	3 x 10^{8}
	DEE s2	687	1
	DEE s2 bound flags	663	1
	DEE s2 split flags	299	1
	HERO	359	1
4	DEE s2mb	1449	2 x 10^{35}
	DEE s2	1333	1 x 10^{35}
	DEE s2 bound flags	1688	1 x 10^{35}
	DEE s2 split flags	875	9 x 10^{19}
	HERO	476	1
5	DEE s2mb	292	3 x 10^{16}
	DEE s2	129	1
	DEE s2 bound flags	72	1
	DEE s2 split flags	46	1
	HERO	35	1

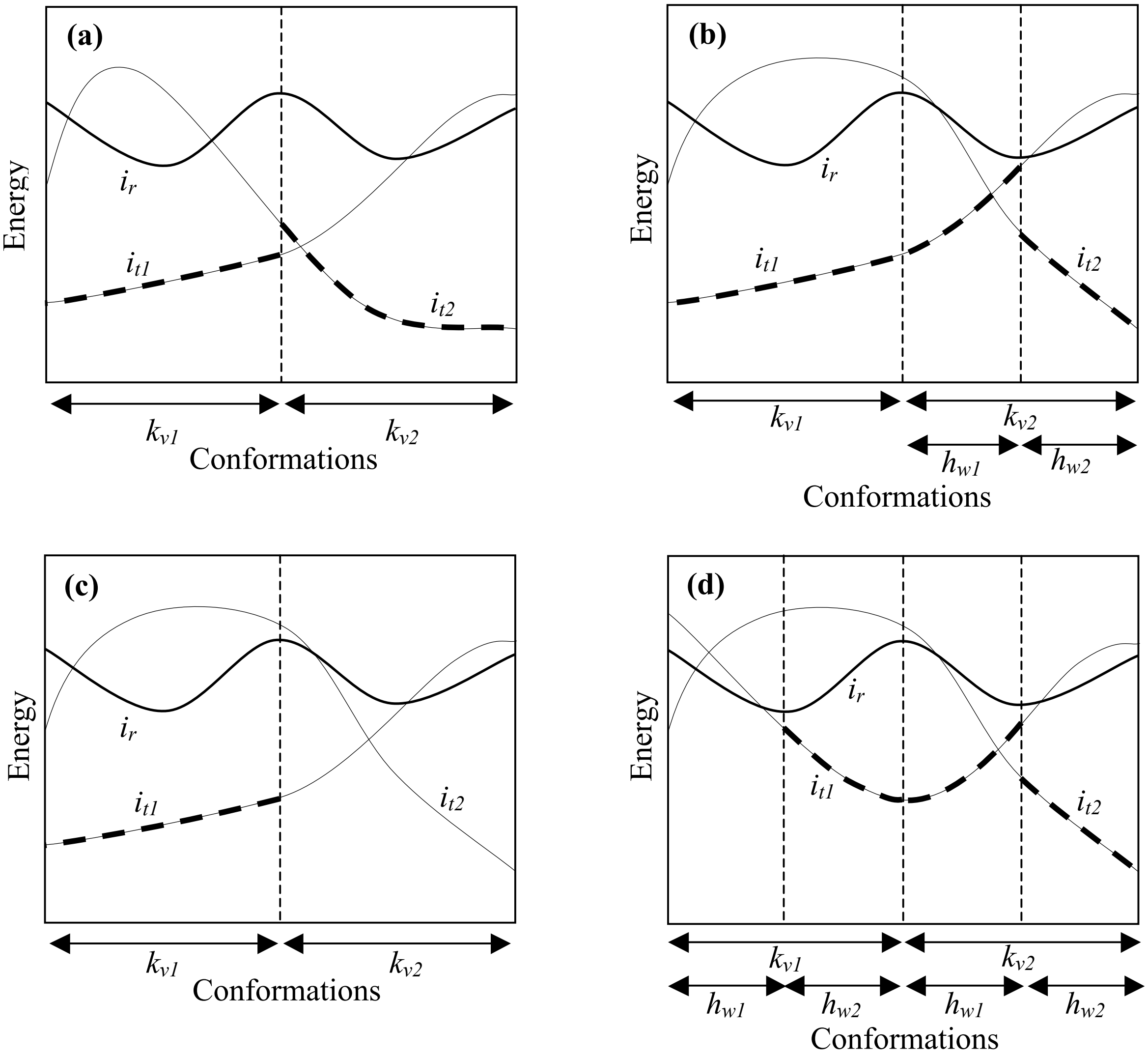

Figure II-1. Application of split DEE to sample energy profiles. The abscissa represents all possible conformations of the protein and the ordinate represents the net energy contributions produced by interactions with specific rotamers at position *i*. (a) Split elimination ($s = 1$): i_r is dominated by i_{t1} and i_{t2} in the partitions corresponding to splitting rotamers k_{v1} and k_{v2}, respectively. Hence, i_r may be eliminated even though it is not dominated by any single rotamer for all of conformational space. (b) Split elimination ($s = 2$): because neither i_{t1} nor i_{t2} dominates i_r in partition k_{v2}, a second splitting position is used to create subpartitions h_{w1} and h_{w2}, where i_r is dominated by i_{t1} and i_{t2}, respectively. Hence, i_r may be eliminated using two splitting positions. (c) Split flagging ($s = 1$): i_r is not dominated in partition k_{v2} so elimination is not possible with only one splitting position. However, i_r is dominated by i_{t1} in partition k_{v1}, so that pair (i_r, k_{v1}) may be flagged. (d) Split flagging ($s = 2$): i_r is no longer dominated for all of conformational space so it cannot be eliminated with only two splitting positions. However, i_r is dominated for partition k_{v2} by i_{t1} and i_{t2} in subpartitions h_{w1} and h_{w2}, respectively. Hence, the pair (i_r, k_{v2}) may be flagged. Likewise, the pair (i_r, h_{w2}) may be flagged, as becomes more readily apparent if the hierarchy of the splitting positions *k* and *h* is reversed.

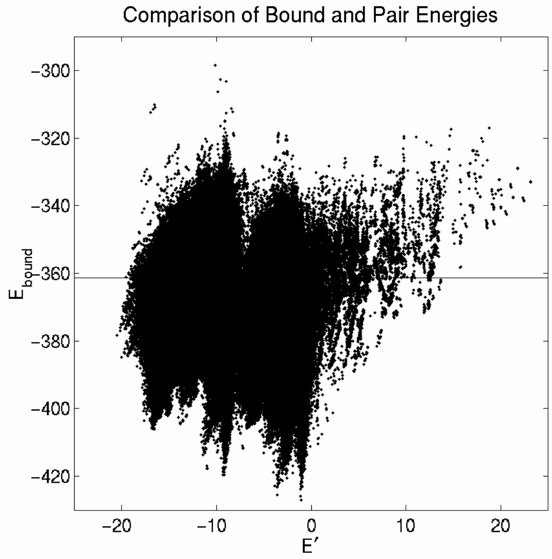

Figure II-2. Comparison of bounding and pairs energies during a bound flags iteration of the plastocyanin core calculation of Case 1. The reference energy obtained by a Monte Carlo calculation is shown as a horizontal line. All pairs with a bounding energy above the line may be flagged. In this instance, 400,822 out of 966,656, or 41%, of the remaining unflagged pairs can now be flagged as dead-ending.

1. Goldstein singles DEE until no further eliminations
2. Split singles DEE with split flags ($s = 1$) until no further eliminations
3. Split singles DEE with split flags ($s \geq 2$) once for each rotamer (with or without magic bullet metric)
4. Singles bounding criterion once for each rotamer
5. Alternate sequentially between the following, applying one during each cycle:
    - Magic bullet Goldstein doubles once for each rotamer pair
    - Monte Carlo search to find $E_{\text{reference}}$ from a valid conformation followed by doubles bounding criterion once for each rotamer pair
    - Full Goldstein doubles once for each rotamer pair using q_{rs} and q_{uv} metrics
    - Unification of residues with the highest fraction of dead-ending pairs
6. Return to 1

Figure II-3. The schedule of dominance and bounding criteria used for hybrid exact rotamer optimization (HERO).

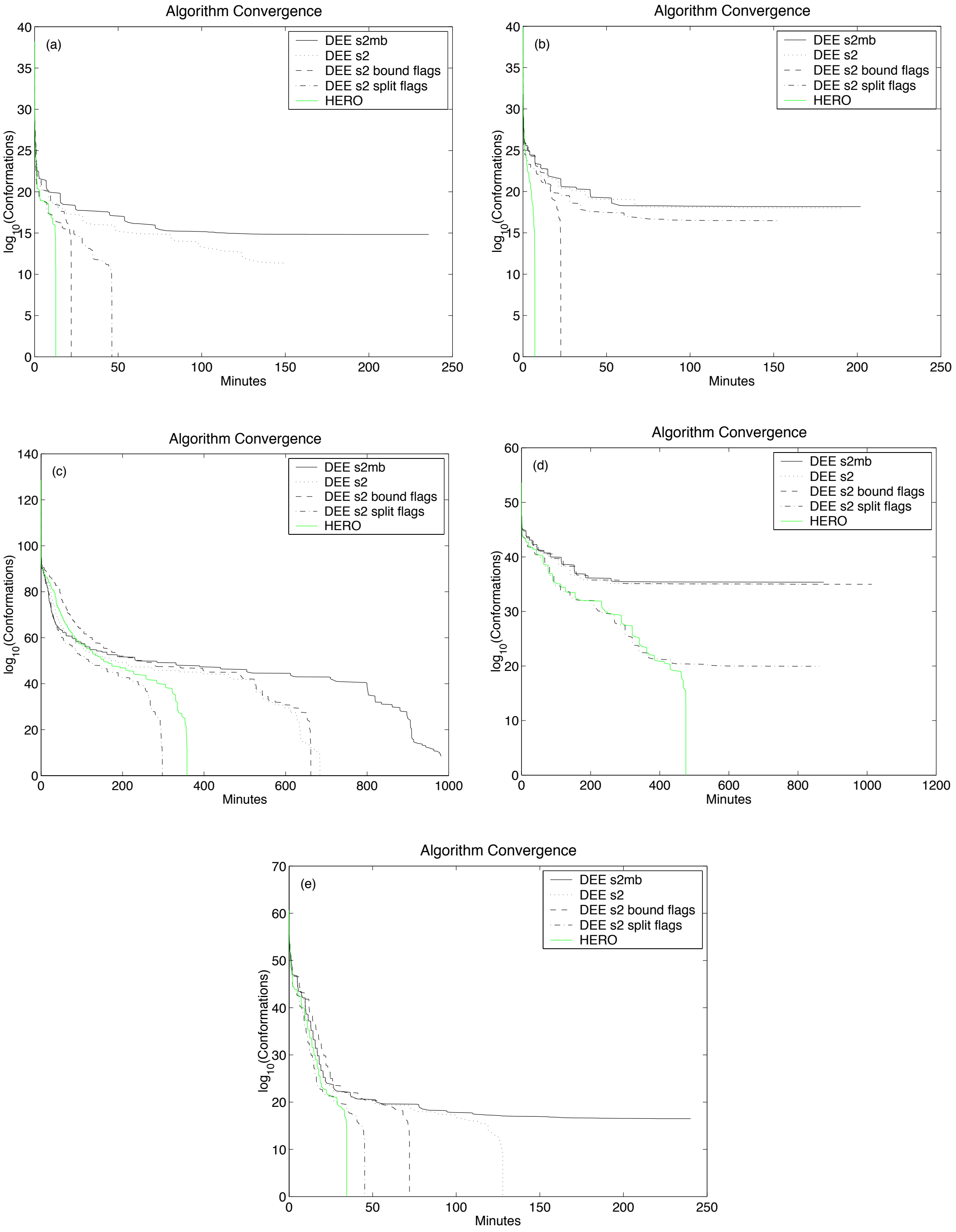

Figure II-4. DEE convergence results. (a) Case 1: full core design of plastocyanin, (b) Case 2: full core design of a novel repeating backbone, (c) Case 3: full core design of the variable domains of the light and heavy chains of a catalytic antibody, (d) Case 4: full core and boundary design of the β1 domain of protein G, (e) Case 5: full surface design of the β1 domain of protein G.

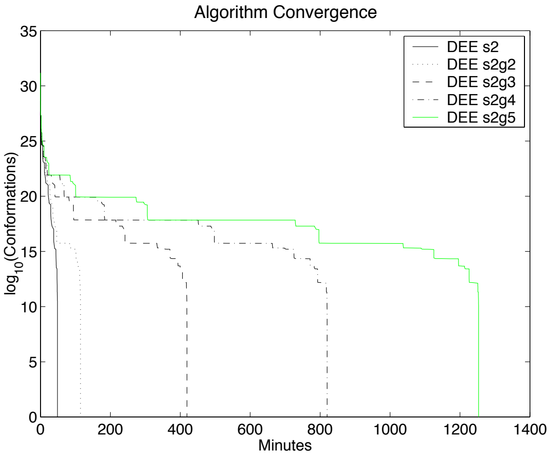

Figure II-5. Performance assessment of "generalized DEE" for a partial surface design of the β1 domain of protein G. Comparisons are made relative to the baseline scheme (DEE s2) using reoptimizations at a maximum of (*g* = 2, 3, 4, 5) positions.

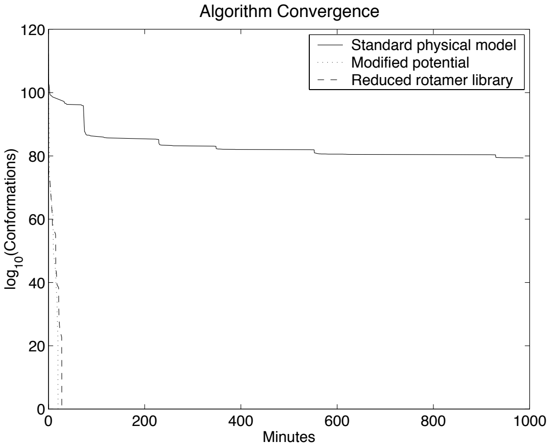

Figure II-6. Convergence comparison for HERO on a full sequence design of the β1 domain of protein G using the experimentally validated "standard" potential function and rotamer library, a modified potential function with the standard rotamer library, and the standard potential function with a reduced rotamer library.

Chapter III

Preprocessing of rotamers for protein design calculations

The text of this chapter has been adapted from a published manuscript that was coauthored with Premal S. Shah and Professor Stephen L. Mayo.

P. S. Shah, G. K. Hom, and S. L. Mayo. 2004. *J. Comput. Chem.*, **25:** 1797–1800.

Abstract

We have developed a process that significantly reduces the number of rotamers in computational protein design calculations. This process, which we call Vegas, results in dramatic computational performance increases when used with algorithms based on the dead-end elimination (DEE) theorem. Vegas estimates the energy of each rotamer at each position by fixing each rotamer in turn and utilizing various search algorithms to optimize the remaining positions. Algorithms used for this context-specific optimization can include Monte Carlo, self-consistent mean field, and the evaluation of an expression that generates a lower bound energy for the fixed rotamer. Rotamers with energies above a user-defined cutoff value are eliminated. We found that using Vegas to preprocess rotamers significantly reduced the calculation time of subsequent DEE-based algorithms while retaining the global minimum energy conformation. For a full boundary design of a 51 amino acid fragment of engrailed homeodomain, the total calculation time was reduced by 12-fold.

Introduction

An important goal of computational protein design is to identify the amino acid sequence and side-chain orientations that correspond to the global minimum energy conformation (GMEC). However, searching for the GMEC is challenging due to the enormity of sequence space; even a small protein of 100 amino acids has 20^{100} ($\sim10^{130}$) possible sequences. Accounting for side-chain flexibility by including different side-chain conformations called rotamers[1-3] further increases the combinatorial complexity. Consequently, exhaustive searches for the GMEC are almost always intractable.

Algorithms based on the dead-end elimination (DEE) theorem[4] have been developed to address combinatorial optimization problems in side-chain placement and protein design. If DEE-based algorithms converge, the solution is guaranteed to be the GMEC. As a result, not only are these algorithms useful when performing force field improvements or parameter optimization,[5,6] their use has proven to be successful for many challenging design problems.[7-11] Although recent enhancements to DEE have allowed difficult designs to be performed,[12-15] more ambitious design problems can cause even the most effective DEE-based algorithms to stall. In addition, some calculations take an impractical amount of time to converge to the GMEC. In such cases, other algorithms may be employed. These include Monte Carlo (MC) methods,[16,17] genetic algorithms,[18,19] self-consistent mean field (SCMF) techniques,[20,21] and branch-and-bound methods.[22] Although these approaches can provide solutions when DEE-based algorithms stall, they typically have the drawback of not being able to guarantee that their solutions are the GMEC even when starting from a DEE-reduced rotamer space. As a result, there is still ample motivation to develop techniques to improve or assist current DEE-based algorithms.

One approach is to reduce the number of rotamers in a calculation by eliminating a subset of rotamers prior to use of DEE-based algorithms. An example of this strategy can be found in the high-energy threshold reduction method.[23] In most cases, by eliminating rotamers possessing energies above a user-defined threshold, De Maeyer et al. were able to eliminate over one-third of rotamers without sacrificing the GMEC in side-chain placement calculations. Remaining rotamers were then evaluated with DEE. Here, we present a similar approach for protein design calculations; we prune rotamer space by judiciously eliminating rotamers, thus allowing DEE-based algorithms to proceed more efficiently. Our method, which we call Vegas, scores each rotamer at each position by fixing it in turn and using MC or SCMF to optimize the rest of the positions. The rotamer's score is the energy of the resulting solution. In addition, a rotamer's score can be calculated by evaluating an expression that generates a lower bound energy.[22] Rotamers remaining after the elimination step are passed on to a DEE-based algorithm. We can safely eliminate a large subset of rotamers without compromising the GMEC, and we observe a significant reduction in total computation time.

Vegas

Vegas reduces the number of rotamers in protein design calculations by applying a rejection criterion after obtaining a score for each rotamer at each position. This is done by fixing the rotamer to be scored and using various optimization algorithms to generate a rotamer sequence for the rest of the molecule. The rotamer's score is the energy of the resulting solution. In this report, two optimization algorithms were used: one based on Monte Carlo (MC) methods,[24] and another based on self-consistent mean field theory (SCMF).[24] In addition, a rotamer's score was also obtained by evaluating an expression that provided a lower bound energy (Bound)[15,22] for the fixed rotamer [eq. (9) in ref. 15]. Rotamers with scores above the best score for that position plus a user-defined threshold value are eliminated. Remaining rotamers are then optimized with HERO,[15] an extension of DEE.

Results

We used two test cases to assess the effectiveness of Vegas. We started with the designs of different regions of a very small protein and increased the computational complexity with the second test case. Vegas's effectiveness was evaluated by its ability to retain the GMEC and increase computational efficiency. To check Vegas's performance in not eliminating GMEC rotamers, the GMEC was first obtained without Vegas in a reference calculation using HERO alone. The different versions of Vegas are referred to with an underscore between Vegas and the method used to obtain the rotamer score. For example, use of MC with Vegas is referred to as Vegas_MC.

Test case 1

We performed designs of the core, boundary, and surface regions of the β1 domain of protein G (Gβ1).[25] These small, relatively simple designs were done to demonstrate the ability of Vegas to safely apply a rejection criterion to eliminate rotamers without sacrificing the GMEC. **Table III-1** lists the number of rotamers eliminated as the threshold value is increased. All versions of Vegas performed equally well for core and boundary designs; the most aggressive threshold value (5 kcal/mol) allowed about 90% of rotamers to be eliminated without losing the GMEC. Elimination was more difficult with surface residues. Compared to Vegas_MC, Vegas_SCMF, and Vegas_Bound allowed for more aggressive threshold values to be applied without losing the GMEC.

Test case 2

A boundary design of a 51 amino acid fragment of the engrailed homeodomain (ENH)[26] was performed to determine Vegas's ability to increase computational efficiency without compromising accuracy **(Figs. III-1 and III-2)**. Vegas_MC and Vegas_SCMF retained the GMEC when threshold values of 10 kcal/mol and larger were used. At 10 kcal/mol, 72% and 64% of the 3571 total rotamers in the calculation were eliminated with Vegas_MC and Vegas_SCMF, respectively. Interestingly, a threshold of 5 kcal/mol for Vegas_MC produced the same amino acid sequence as the one in the GMEC; however, the conformations of some of the amino acids were different. We could not be as aggressive with Vegas_Bound; a minimum of 20 kcal/mol was required to obtain the GMEC. At this threshold, 41% of the rotamers were eliminated.

Although Vegas_MC and Vegas_SCMF allowed the use of more aggressive threshold values while retaining the GMEC, comparison of total calculation times shows Vegas_Bound to be more efficient **(Fig. III-2)**. At a relatively conservative threshold value of 40 kcal/mol, Vegas_Bound obtained the GMEC almost four times faster than the reference calculation. At 20 kcal/mol, it produced the GMEC in only 8 processor hours—a 12-fold improvement over the reference calculation. In comparison, Vegas_MC was only able to achieve a twofold overall speed enhancement. Vegas_SCMF, on the other hand, actually caused the calculation to run two times slower than the reference calculation.

Discussion

Vegas is an efficient protein design tool that can reduce computational complexity without sacrificing the ability to obtain ground-state solutions. Its computational efficiency becomes more pronounced with increasing problem size. Vegas produced a 12-fold reduction in the time required to solve the boundary design of ENH, decreasing the total processing time from 92 to 8 hours. This increase in computational speed resulted from elimination of about 41% of the rotamers, without losing rotamers in the GMEC. The high efficiency of Vegas_Bound for this design compared to Vegas_MC and Vegas_SCMF **(Fig. III-2)** can be attributed to a dramatic difference in time for scoring the rotamers. The rotamer scoring times for Vegas_MC and Vegas_SCMF were 45 and 198 processor hours, respectively, while Vegas_Bound scored rotamers in less than 1 min on a single processor.

The accuracy and increased efficiency provided by Vegas can extend the capabilities of protein design. For example, Vegas allows the use of larger rotamer libraries, which may provide lower energy solutions to design problems. Larger rotamer libraries have been shown to improve accuracy in side-chain placement calculations.[23] The use of Vegas can also allow more difficult designs to be performed and can facilitate the design of many features including functionally important properties.

A recent side-chain placement algorithm called FASTER[27] has shown promise when adapted to protein design (data not shown). Elements of FASTER could be implemented as an additional rotamer-scoring method within Vegas. Vegas_FASTER, as well as Vegas with other optimization algorithms, is a viable option in the future. We used Vegas here as a preprocessor to HERO; however, Vegas is a general preprocessing method and can be combined with any relevant optimization algorithm.

Methods

Computational methods

A description of force field potential functions and their parameters can be found in previous work.[5,7,28-30] We used an expanded version of the backbone dependent rotamer library described by Dunbrack and Karplus.[3] An automated algorithm was employed that classified residue positions as core, boundary, or surface.[5] For core positions, we allowed the selection of the amino acids A, V, L, I, F, Y, and W. For surface positions, we allowed A, S, T, D, N, H, E, Q, K, and R, and for boundary positions, we allowed all amino acids except G, P, C, and M. HERO and the bounding expression were implemented as described by Gordon et al.,[15] and MC and SCMF were implemented as described previously.[24] For MC, we used 5 annealing cycles of 10^6 steps per cycle. Low and high annealing temperatures were 150 K and 4000 K, respectively. For SCMF, we used initial and final temperatures of 20,000 K and 300 K, respectively, with the temperature lowered in 100 K increments. A convergence criterion of 0.001 and a pair-energy threshold of 100 kcal/mol were used.

Test case designs

In test case 1, we designed the core, boundary, and surface regions of Gβ1 (PDB code 1pga).[25] Core positions were 3, 5, 7, 9, 20, 26, 30, 34, 39, 41, 52, and 54. Boundary positions were 1, 12, 16, 18, 23, 25, 27, 29, 31, 33, 37, 43, 45, 50, and 56. Surface positions were 2, 4, 6, 8, 10, 11, 13, 14, 15, 17, 19, 21, 22, 24, 28, 32, 35, 36, 38, 40, 42, 44, 46, 47, 48, 49, 51, 53, and 55. Design of a region involved allowing all allowable amino acids for that region, while keeping the other two regions fixed in both identity and conformation. Test case 2 was the boundary design of ENH (PDB code 1enh;[26] positions 1, 3, 10, 14, 19, 21, 25, 30, 47, and 51). Core and surface positions were kept fixed in identity but their conformations were allowed to change. All calculations were performed on an IBM SP3 running 375-MHz Power3 processors.

Acknowledgments

The authors would like to thank Marie Ary for critical review of the manuscript and for offering valuable comments.

References

1. Janin, J., and Wodak, S. 1978. Conformation of amino acid side-chains in proteins. *J Mol Biol* **125:** 357-386.

2. Ponder, J.W., and Richards, F.M. 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.

3. Dunbrack, R.L., Jr., and Karplus, M. 1993. Backbone-dependent rotamer library for proteins. Application to side-chain prediction. *J Mol Biol* **230:** 543-574.

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

5. Dahiyat, B.I., and Mayo, S.L. 1996. Protein design automation. *Protein Sci* **5:** 895-903.

6. Dahiyat, B.I., and Mayo, S.L. 1997. Probing the role of packing specificity in protein design. *Proc Natl Acad Sci U S A* **94:** 10172-10177.

7. Dahiyat, B.I., and Mayo, S.L. 1997. De novo protein design: fully automated sequence selection. *Science* **278:** 82-87.

8. Malakauskas, S.M., and Mayo, S.L. 1998. Design, structure and stability of a hyperthermophilic protein variant. *Nat Struct Biol* **5:** 470-475.

9. Bolon, D.N., and Mayo, S.L. 2001. From the Cover: Enzyme-like proteins by computational design. *Proc Natl Acad Sci U S A* **98:** 14274-14279.

10. Marshall, S.A., and Mayo, S.L. 2001. Achieving stability and conformational specificity in designed proteins via binary patterning. *J Mol Biol* **305:** 619-631.

11. Looger, L.L., Dwyer, M.A., Smith, J.J., and Hellinga, H.W. 2003. Computational design of receptor and sensor proteins with novel functions. *Nature* **423:** 185-190.

12. Gordon, D.B., and Mayo, S.L. 1998. Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *J Comput Chem* **19:** 1505-1514.

13. Pierce, N.A., Spriet, J.A., and Mayo, S.L. 2000. Conformational splitting: A more powerful criterion for dead-end elimination. *J Comput Chem* **21:** 999-1009.

14. Looger, L.L., and Hellinga, H.W. 2001. Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: implications for protein design and structural genomics. *J Mol Biol* **307:** 429-445.

15. Gordon, D.B., Hom, G.K., Mayo, S.L., and Pierce, N.A. 2003. Exact rotamer optimization for protein design. *J Comput Chem* **24:** 232-243.

16. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. 1953. Equation of state calculations by fast computing machines. *J Chem Phys* **21:** 1087-1092.

17. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. 1983. Optimization by simulated annealing. *Science* **220:** 671-680.

18. Holland, J.H. 1992. *Adaptation in natural and artificial systems*. The MIT Press, Cambridge, Massachusetts.

19. Desjarlais, J.R., and Handel, T.M. 1995. De novo design of the hydrophobic cores of proteins. *Protein Sci* **4:** 2006-2018.

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

21. Koehl, P., and Delarue, M. 1996. Mean-field minimization methods for biological macromolecules. *Curr Opin Struct Biol* **6:** 222-226.

22. Gordon, D.B., and Mayo, S.L. 1999. Branch-and-terminate: a combinatorial optimization algorithm for protein design. *Structure Fold Des* **7:** 1089-1098.

23. De Maeyer, M., Desmet, J., and Lasters, I. 1997. All in one: a highly detailed rotamer library improves both accuracy and speed in the modelling of sidechains by dead-end elimination. *Fold Des* **2:** 53-66.

24. Voigt, C.A., Gordon, D.B., and Mayo, S.L. 2000. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *J Mol Biol* **299:** 789-803.

25. Gallagher, T., Alexander, P., Bryan, P., and Gilliland, G.L. 1994. Two crystal structures of the B1 immunoglobulin-binding domain of streptococcal protein G and comparison with NMR. *Biochemistry* **33:** 4721-4729.

26. Clarke, N.D., Kissinger, C.R., Desjarlais, J., Gilliland, G.L., and Pabo, C.O. 1994. Structural studies of the engrailed homeodomain. *Protein Sci* **3:** 1779-1787.

27. Desmet, J., Spriet, J., and Lasters, I. 2002. Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins* **48:** 31-43.

28. Gordon, D.B., Marshall, S.A., and Mayo, S.L. 1999. Energy functions for protein design. *Curr Opin Struct Biol* **9:** 509-513.

29. Street, A.G., and Mayo, S.L. 1999. Computational protein design. *Structure Fold Des* **7:** R105-109.

30. Street, A.G., and Mayo, S.L. 1998. Pairwise calculation of protein solvent-accessible surface areas. *Fold Des* **3:** 253-258.

Table III-1. Number of rotamers eliminated with varying threshold values for core, boundary, and surface designs of the β1 domain of protein G: comparison using Vegas_MC, Vegas_SCMF, and Vegas_Bound.

Threshold	Core (413)[a]			Boundary (2663)[a]			Surface (4971)[a]		
(kcal/mol)	Vegas_MC	Vegas_SCMF	Vegas_Bound	Vegas_MC	Vegas_SCMF	Vegas_Bound	Vegas_MC	Vegas_SCMF	Vegas_Bound
5	373	373	362	2254	2319	2357	4540[b]	3795[b]	3355[b]
10	332	337	323	1371	1495	1516	2995[b]	1901	1536
20	262	269	225	336	360	371	700	536	496
40	183	186	173	130	129	128	278	272	269
80	141	143	137	96	96	96	225	222	219
160	120	20	117	84	0	87	165	10	163

[a] Initial number of rotamers.
[b] Calculation failed to produce GMEC.

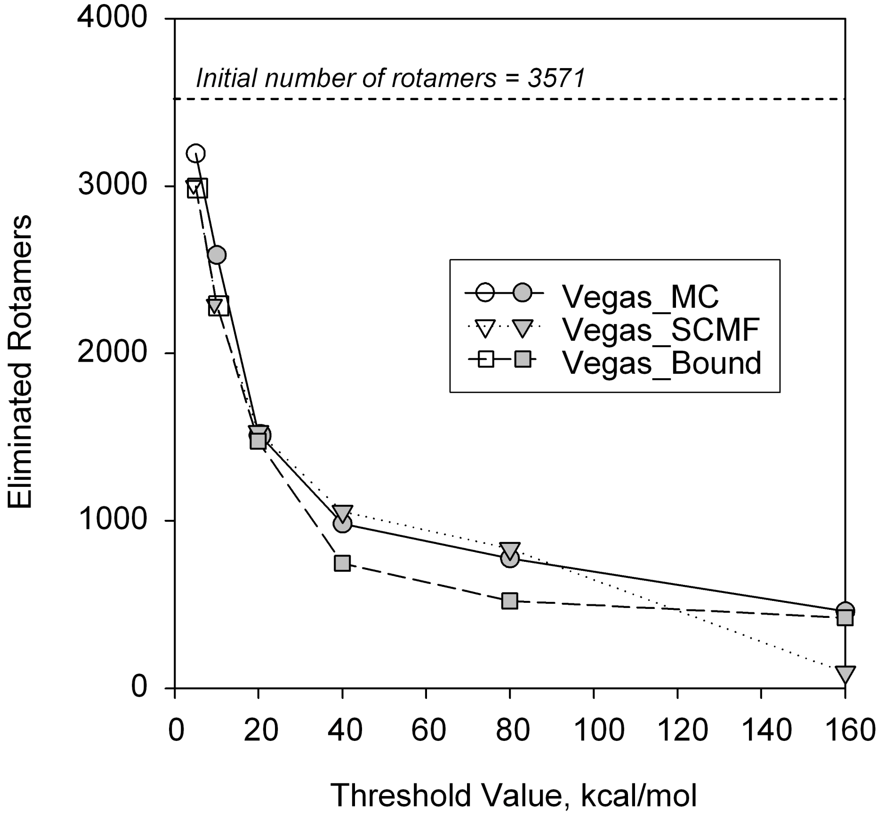

Figure III-1. Number of rotamers eliminated with varying threshold values for the boundary design of engrailed homeodomain. The reference calculation (i.e., with HERO alone) contained 3571 rotamers. Threshold values that failed to produce the GMEC are shown with open symbols.

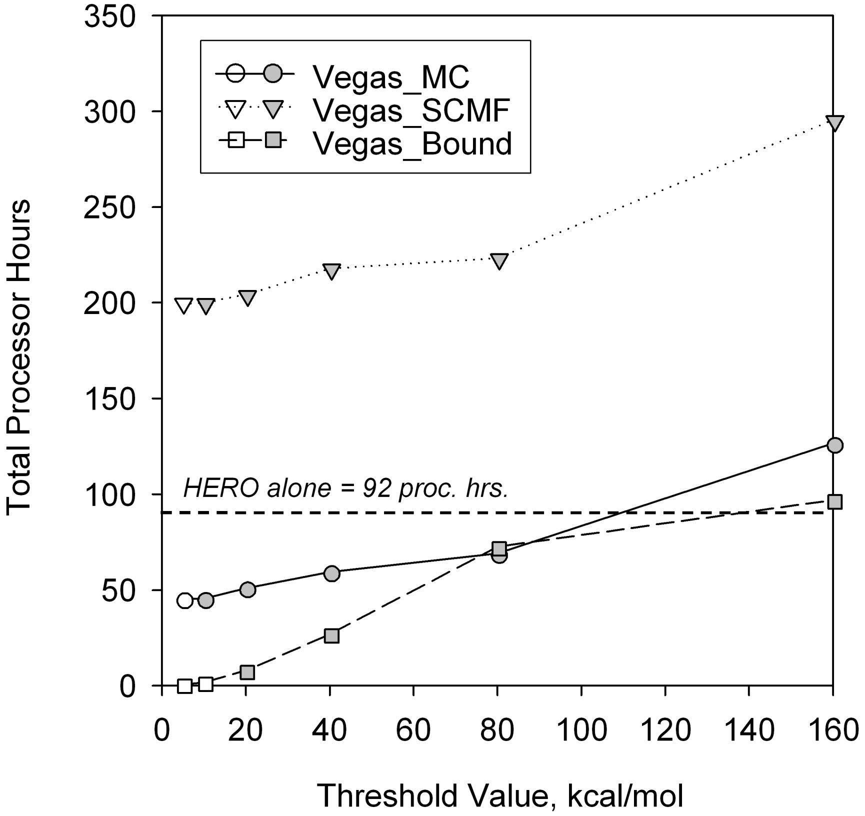

Figure III-2. Total calculation times for the boundary design of engrailed homeodomain. The reference calculation (i.e., with HERO alone) took 92 processor hours. Threshold values that failed to produce the GMEC are shown with open symbols.

Chapter IV

Thermodynamic and structural characterization of full-sequence designs

The text of this chapter has been adapted from a manuscript that was coauthored with Premal S. Shah, Scott A. Ross, and Professor Stephen L. Mayo.

P. S. Shah,* G. K. Hom,* S. A. Ross, and S. L. Mayo. 2005. *To be submitted.*
(*P. S. Shah and G. K. Hom contributed equally to this work.)

Abstract

Sequence optimization algorithms based on the dead-end elimination (DEE) theorem are preferred in computational protein design because, if they converge, their solutions are guaranteed to be the ground-state solutions. However, the increasing size and complexities of designs can cause DEE-based algorithms to stall, failing to deliver a solution. We have used three alternate sequence optimization algorithms in concert with the ORBIT protein design software to simultaneously optimize every position of a 51-amino acid fragment of the *Drosophila* engrailed homeodomain. Two of the sequences obtained from the calculations were studied in detail. The optimized sequences share no statistical similarity to any known sequence and differ from the wild-type sequence by approximately 80%. Based on physical studies of the optimized variants, we conclude that the proteins are nearly identical to each other, displaying hallmarks of well-folded,

all α-helical proteins. The thermodynamic stabilities of the designed variants were enhanced by approximately 2 kcal/mol over the wild-type protein at 25°C. In addition, the designed variants have melting temperatures in excess of 100°C compared to 43°C for the wild-type protein. We solved the solution structure of one of the designed variants and found that the protein folds accurately into the desired target fold. Knowledge that non-DEE-based sequence optimization algorithms can be used for large, challenging problems leading to variants with markedly improved stability and high specificity for the target fold allows for more ambitious protein design problems to be undertaken.

Introduction

Computational protein design seeks to find amino acid sequences compatible with a target fold. In general, the global minimum energy conformation (GMEC) is desired, since this sequence and conformation confers optimal stability for the fold, provided the physical forces governing protein structure and stability are accurately modeled. Obtaining the GMEC while simultaneously optimizing every position in a protein is a challenging combinatorial problem; for a relatively small 50-residue protein, the GMEC must be identified from 10^{65} possible amino acid sequences. When different conformers of amino acids (rotamers) are included, the complexity grows substantially, requiring the consideration of over 10^{100} rotamer sequences.

Many difficult designs[1-5] have been performed using algorithms based on the dead-end elimination[6] (DEE) theorem. DEE-based algorithms are ideal because if they converge, their solutions are guaranteed to be the GMEC. However, increasingly challenging design problems can prevent even the most effective DEE-based algorithms[7-

[10] from converging in any practical amount of time. Furthermore, in some cases, these algorithms stall and fail to converge entirely. As an alternative, non-DEE-based algorithms may be employed to obtain sequences compatible with a target fold. However, these algorithms also have their limitations: they do not necessarily provide the GMEC, and their performance has been shown to decay as the size of the design increases.[11]

Our goal was to determine whether the use of non-DEE-based algorithms on large, complex designs can provide solutions that are stable and assume the target fold. We undertook the full sequence design of a 51-amino acid fragment of the *Drosophila* engrailed homeodomain (ENH). Non-DEE-based algorithms were required because DEE-based algorithms failed to converge. We used three algorithms: Monte Carlo[12,13] (MC), Vegas,[14] and FASTER.[15] MC is a commonly used stochastic search algorithm, Vegas is a rotamer pruning algorithm recently developed in our laboratory that is efficient for large designs, and FASTER is a fast and accurate side-chain placement method, which we adapted for protein design applications. The protein variants predicted with these algorithms were expressed, purified, and characterized thermodynamically. Furthermore, the solution structure of one of the variants was solved in order to assess whether the designed proteins adopt the desired target fold. This work adds to the small number of full-sequence designs performed to date for which thermodynamic and structural studies have been perfomed.[16,17]

Results

Computational sequence optimization

We divided ENH[18] into core, boundary, and surface regions with an automated residue classification algorithm[19] and modeled the physical forces within each region with a potential energy function that includes van der Waals, electrostatic, solvation, and hydrogen bonding terms.[19-22] Only nonpolar amino acids were allowed in the core, while on the surface, only polar amino acids were considered. A fixed binary pattern was used that assigned boundary positions to either the core or the surface based on exposed surface area;[3] this fixed binary pattern has been shown to confer added stability to the ENH fold.[3] The amino acid identities of positions involved in helix capping and helix dipoles were further restricted as described previously.[4] To account for the torsional flexibility of amino acids, a backbone-dependent rotamer library,[23] based on that of Dunbrack and Karplus,[24] was employed. The total initial search space for this calculation was 10^{111} rotamer sequences.

Our laboratory has successfully used DEE-based sequence optimization algorithms[7-10,16] to generate sequences for many design problems.[1,2,16] In this study, we initially attempted optimization with HERO,[10] an extension of DEE that performs more efficiently on large calculations. However, HERO stalled and failed to provide an answer. As a result, three non-DEE-based sequence optimization algorithms, MC, Vegas, and FASTER, were used to predict sequences compatible with the target ENH fold. The best rotamer sequences generated by Vegas and FASTER are identical and have simulation energies of -225.0 kcal/mol. This sequence (FSM1_VF) is a 39-fold mutant of the wild-type sequence **(Fig. IV-1)**. The best MC solution (FSM1_MC) has a slightly higher simulation energy (-223.4 kcal/mol) and is a 40-fold mutant of wild-type ENH and an 11-fold mutant of FSM1_VF. A BLAST[25] search indicated that the two optimized variants have no statistically significant similarity to any known sequence.

Physical characterization of ENH variants

Far-ultraviolet (UV) circular dichroism (CD) spectroscopy of FSM1_VF and FSM1_MC revealed spectra characteristic of α-helical proteins **(Fig. IV-2)**. The spectra for the two variants are almost superimposable and are characteristic of α-helical proteins with minima at 208 and 222 nm. The spectra are also very similar to those for wild-type ENH as well as other well-folded ENH variants produced in our laboratory.[3,4,26] 1D ^{1}H nuclear magnetic resonance (NMR) spectroscopy performed on both proteins produced spectra displaying the sharp, moderately dispersed lines expected of a well-folded protein **(Fig. IV-3)**.

Thermal denaturations monitored by CD at 222 nm revealed that both proteins do not complete their unfolding transitions by 99°C, indicating that they are still folded at this temperature (data not shown). In comparison, the wild type has a T_m of 43°C **(Table IV-1)**.[26] Chemical denaturations using guanidinium hydrochloride were performed to determine unfolding free energies (ΔG_{unfold}). The variants were over 2 kcal/mol more stable than the wild-type protein under similar conditions **(Table IV-1)**.[27] This is a remarkable result considering that approximately 80% of the wild-type sequence was mutated to obtain our designed sequences.

ANS (1-anilino-napthalene-8-sulfonate) binding was used to further validate the structural integrity of the ENH variants. ANS selectively binds molten globule states of proteins.[28] Molten globules exhibit pronounced secondary structure and compactness but lack packed tertiary structure. Hen egg-white lysozyme (HEWL) in 25% HFA (hexafluoroacetone hydrate) was used as a positive control; under this condition, HEWL binds ANS and exhibits molten globule characteristics.[29] Although the ENH variants showed some evidence of ANS binding, it was almost 8-fold lower than HEWL (data not shown). This slight ANS binding is most likely due to exposed hydrophobic patches rather than the result of binding to a molten globule state (see below).[28] Overall, the spectral and thermodynamic data indicate that the designed variants are very stable and are physically and structurally similar.

Solution structure of FSM1_VF

The solution structure of FSM1_VF was solved by NMR. Evidently due to the helical structure and relatively low sequence diversity of FSM1_VF **(Fig. IV-1)**, its NMR spectra display considerable chemical shift degeneracy. Thus, it was necessary to use both HNCACB/CBCA(CO)NH and HNCO/HN(CA)CO experiment pairs on uniformly ^{15}N, ^{13}C-labeled protein to sequentially assign backbone atom chemical shifts. Other standard double and triple resonance NMR experiments were then sufficient to achieve nearly complete assignment of side-chain atom chemical shifts. Over 1300 loose geometric constraints (interproton distances from NOEs, dihedral angles, and hydrogen bonds) on the structure were derived from NMR data **(Table IV-2)**. The program ARIA[30] was used both to assign many of these constraints and to calculate an ensemble of structures consistent with them **(Fig. IV-4)**. The ensemble is of a precision typical for homeodomain NMR structures,[31] with 0.59 Å root mean square (r.m.s.) deviation to the mean for backbone heavy atoms of residues 3–45; the ensemble is also of good stereochemical quality, with 96.6% of residues in most-favored or allowed regions of ϕ,ψ space.

The calculated ensemble shows that FSM1_VF adopts the anticipated ENH fold. Helices 1 and 2 are well-defined, as is the tight turn between helices 2 and 3 and the first two turns of helix 3. The termini are poorly localized, as well as residues 18–20 in the loop between helices 1 and 2. Paucity of data makes the origin of this imprecision uncertain for the loop residues. However, intermediate $^3J_{HNHA}$ coupling constant values for residues 1–5, 46, and 48–51 suggest that the termini are disordered. Disorder in the backbone in these portions of the sequence is accompanied by side-chain disorder as indicated by low χ_1 and χ_2 angular order parameters for nominal core residues W3, F43, F44, and F47.

We compared the FSM1_VF solution structure to the ENH crystal structure. The experimental structure closest to the mean of the ensemble in Figure 4-4 has a backbone r.m.s. deviation of 2.5 Å from the crystal structure for C_α atoms of residues 3-45 (Figure 4-5). The largest differences from the crystal structure were found at the termini and in the orientation of helix 3 with respect to helices 1 and 2. Indeed, solution structures of homoedomains uncomplexed to DNA frequently show disorder in both the N terminus and the C-terminal portion of helix 3.[31] In addition, the starting structure is a truncated version of the crystal structure due to lack of electron density at the C terminus. The crystal structure of ENH is thus quite possibly a nonphysical template for these regions of the molecule in solution. Furthermore, the different orientation of helix 3 could easily be an effect propagated from the disordered C terminus, and the disordered aromatic side chains in the termini could account for the modest ANS binding observed. For the remainder of the structure, FSM1_VF matches the template closely.

Discussion

Use of non-DEE-based algorithms

Non-DEE-based algorithms have been used to produce stable proteins;[17,32-35] however, most of these designs were restricted to the core and were less complex than the design performed here. A quantitative comparison showed that the performance of non-DEE-based algorithms decreases as the complexity of the problem increases.[11] Performance was defined as the fraction of rotamers predicted incorrectly compared to the GMEC. The goal of the present study was to determine the effectiveness of non DEE-based algorithms on complex problems such as full-sequence designs; that is, the ability to yield stable proteins that retain high structural specificity for the target fold. Baker and colleagues recently performed full sequence designs using MC with reasonable success;[17] however, the structures of the proteins have not yet been solved. In this study, we clearly demonstrate that three alternatives to DEE-based algorithms (MC, Vegas, and FASTER) can be used on complex problems to predict sequences with protein stabilities much higher than wild type. In addition, we verified that the designed variants have the same topology as the target fold, as shown by the solution structure of FSM1_VF.

These results suggest that many highly stable proteins can be obtained for complex design problems without identifying the GMEC. In fact, an MC search performed around the FSM1_VF sequence showed that there are at least 900 unique amino acid sequences with simulation energies between FSM1_VF (-225.0 kcal/mol) and our other stable variant, FSM1_MC (-223.4 kcal /mol). It is certainly plausible that all of these sequences would yield proteins that are equally stable and target-fold specific. Taken further, there are likely many sequences with simulation energies higher than that of FSM1_MC that would also adopt the target fold and possess stabilities higher than wild type.

The knowledge that very large, previously intractable designs can be successfully performed with non-DEE-based algorithms allows protein designers to tackle more ambitious problems. Catalytic activity can be designed onto larger scaffolds, improved stabilities can be obtained for larger proteins, and complex protein-protein interactions can be studied. Larger rotamer libraries can also be used to enhance the accuracy of the solutions generated.

Methods

Computational modeling

Description of potential functions and parameters can be found in our previous work.[19-22,36,37] For ENH, we identified 11 core positions (7, 11, 15, 29, 33, 34, 35, 39, 40, 43, and 44), 11 boundary positions (1, 3, 10, 14, 19, 21, 25, 26, 30, 47, and 51), and 29 surface positions (2, 4, 5, 6, 8, 9, 12, 13, 16, 17, 18, 20, 22, 23, 24, 27, 28, 31, 32, 36, 37, 38, 41, 42, 45, 46, 48, 49, and 50). The fixed binary pattern of the B6 design in the Marshall and Mayo study[3] was applied to boundary residues. Residues 4, 22, and 36 were treated as helix N-capping positions; residues 5, 6, 23, 24, 37, and 38 as helix N-terminal dipole positions, and residues 16, 17, 31, 32, 49, and 50 as helix C-terminal dipole positions. The rules that govern these positions are described in previous work.[4]

Construction of mutants, protein expression, and purification

Genes encoding the ENH variants were made using recursive PCR techniques[38] and cloned into a modified pET11a (Novagen) vector. Recombinant protein was expressed by IPTG induction in BL21(DE3) hosts (Stratagene) and isolated using a freeze/thaw method.[39] Purification was accomplished using a linear acetonitrile/water gradient containing 0.1% TFA. Molecular weights were verified by mass spectrometry. The resultant protein was a 52-mer, with a methionine at the N terminus.

CD analysis

CD data were collected on an Aviv 62DS spectrometer equipped with a thermoelectric unit and an autotitrator. Wavelength scans and thermal denaturation experiments were performed in a 1 mm path length cell with 50 μM protein in 50 mM sodium phosphate at pH 5.5. Thermal melts were monitored at 222 nm. Data were collected every 1°C with an equilibration time of 2 min and an averaging time of 30 sec. Guanidinium chloride denaturations were done in a 1 cm path length cell with 5 μM protein in 50 mM sodium phosphate at pH 5.5 and 25°C. To keep the protein concentration constant, a saturated solution of guanidinium chloride was prepared with buffer that also included 5 μM protein. A 10 min mixing time and 100 sec averaging time were used. Data were fit and $\Delta G_{\mathrm{unfold}}$ values were obtained using the linear extrapolation method.[40]

NMR spectroscopy

NMR experiments were performed at 20°C on a Varian INOVA 600 spectrometer. Data was processed using nmrPipe[41] and analyzed using NMRview.[42] Backbone chemical shift assignments were obtained from 3D HNCACB, CBCA(CO)NH, HNCO, HN(CA)CO and HNHA spectra. 2D DQF-COSY and 3D C(CO)NH-TOCSY, ^{15}N-TOCSY-HSQC, and HCCH-TOCSY spectra were used to assign aliphatic side-chain atom chemical shifts. Aromatic resonances were assigned from 2D DQF-COSY and TOCSY spectra and from 2D ^{13}C-CT-HSQC and (HB)CB(CGCD)HD and (HB)CB(CGCDCE)HE spectra. Exchange of backbone amide hydrogen atoms was monitored by ^{15}N-HSQC spectra following suspension of protiated ^{15}N-labeled protein in deuterated buffer.

Structure determination

Distance restraints were derived from two 3D ^{13}C-NOESY-HSQC spectra (aliphatic and aromatic), a 3D ^{15}N-NOESY-HSQC spectrum, and a 2D ^{1}H NOESY spectrum. All NOESY spectra were acquired with a 75-ms mixing time. $^{3}J_{HNHA}$ coupling constants were extracted from the HNHA spectrum. These were used, in combination with TALOS[43] analysis of chemical shifts, in the selection of dihedral angle restraints. Where TALOS and coupling constant analyses were consistent, both ϕ and ψ restraints were included. Where TALOS failed to make a prediction, a ϕ restraint was included if warranted by the coupling constant. Error bounds on dihedral restraints were set to ±30°.

A set of 586 manually assigned NOE-derived distance restraints and 57 dihedral angle restraints were used as initial input for ARIA1.2.[30] ARIA identified 659 additional

NOESY cross peaks, for a total of 953 unambiguous and 292 ambiguous distance restraints. At this stage, separate ARIA calculations were carried out fixing the methyl group stereochemistry of each V or L residue in the sequence in turn to obtain stereospecific assignments. In each case, one choice of assignments yielded an ensemble of structures with lower energies, lower χ_1 (and χ_2 for L residues) circular order parameters, and fewer NOE restraint violations than the alternate choice. Finally, the ensemble was examined for likely hydrogen bonds. Hydrogen bonds were judged to be present, and restraints included, if the amide proton had a hydrogen exchange protection factor ≥1000 and if the residue was in a helix. Nineteen residues were thus restrained (1.3 Å $< d_{NH\text{-}O} < 2.5$ Å and 2.3 Å $< d_{N\text{-}O} < 3.5$ Å). Of 100 structures generated in a final ARIA calculation using all of these restraints, 43 had no NOE restraint violations >0.5 Å and no dihedral angle restraint violations >5°. This subset was analyzed with MOLMOL[44] and PROCHECK.[45]

References

1. Malakauskas, S.M., and Mayo, S.L. 1998. Design, structure and stability of a hyperthermophilic protein variant. *Nat Struct Biol* **5:** 470-475.

2. Bolon, D.N., and Mayo, S.L. 2001. From the Cover: Enzyme-like proteins by computational design. *Proc Natl Acad Sci U S A* **98:** 14274-14279.

3. Marshall, S.A., and Mayo, S.L. 2001. Achieving stability and conformational specificity in designed proteins via binary patterning. *J Mol Biol* **305:** 619-631.

4. Marshall, S.A., Morgan, C.S., and Mayo, S.L. 2002. Electrostatics Significantly Affect the Stability of Designed Homeodomain Variants. *J Mol Biol* **316:** 189-199.

5. Looger, L.L., Dwyer, M.A., Smith, J.J., and Hellinga, H.W. 2003. Computational design of receptor and sensor proteins with novel functions. *Nature* **423:** 185-190.

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

7. Gordon, D.B., and Mayo, S.L. 1998. Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *J Comput Chem* **19:** 1505-1514.

8. Gordon, D.B., and Mayo, S.L. 1999. Branch-and-terminate: a combinatorial optimization algorithm for protein design. *Structure Fold Des* **7:** 1089-1098.

9. Pierce, N.A., Spriet, J.A., and Mayo, S.L. 2000. Conformational splitting: A more powerful criterion for dead-end elimination. *J Comput Chem* **21:** 999-1009.

10. Gordon, D.B., Hom, G.K., Mayo, S.L., and Pierce, N.A. 2003. Exact rotamer optimization for protein design. *J Comput Chem* **24:** 232-243.

11. Voigt, C.A., Gordon, D.B., and Mayo, S.L. 2000. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *J Mol Biol* **299:** 789-803.

12. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. 1953. Equation of state calculations by fast computing machines. *J Chem Phys* **21:** 1087-1092.

13. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. 1983. Optimization by simulated annealing. *Science* **220:** 671-680.

14. Shah, P.S., Hom, G.K., and Mayo, S.L. 2004. Preprocessing of rotamers for protein design calculations. *J. Comput. Chem.* **25:** 1797-1800.

15. Desmet, J., Spriet, J., and Lasters, I. 2002. Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins* **48:** 31-43.

16. Dahiyat, B.I., and Mayo, S.L. 1997. De novo protein design: fully automated sequence selection. *Science* **278:** 82-87.

17. Dantas, G., Kuhlman, B., Callender, D., Wong, M., and Baker, D. 2003. A large scale test of computational protein design: folding and stability of nine completely redesigned globular proteins. *J Mol Biol* **332:** 449-460.

18. Clarke, N.D., Kissinger, C.R., Desjarlais, J., Gilliland, G.L., and Pabo, C.O. 1994. Structural studies of the engrailed homeodomain. *Protein Sci* **3:** 1779-1787.

19. Dahiyat, B.I., and Mayo, S.L. 1996. Protein design automation. *Protein Sci* **5:** 895-903.

20. Dahiyat, B.I., and Mayo, S.L. 1997. Probing the role of packing specificity in protein design. *Proc Natl Acad Sci U S A* **94:** 10172-10177.

21. Dahiyat, B.I., Gordon, D.B., and Mayo, S.L. 1997. Automated design of the surface positions of protein helices. *Protein Sci* **6:** 1333-1337.

22. Street, A.G., and Mayo, S.L. 1998. Pairwise calculation of protein solvent-accessible surface areas. *Fold Des* **3:** 253-258.

23. Dunbrack, R. 2002. Rotamer Libraries in the 21(st) Century. *Curr Opin Struct Biol* **12:** 431.

24. Dunbrack, R.L., Jr., and Karplus, M. 1993. Backbone-dependent rotamer library for proteins. Application to side-chain prediction. *J Mol Biol* **230:** 543-574.

25. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D.J. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* **25:** 3389-3402.

26. Morgan, C.S. 2000. Full sequence design of an alpha-helical protein and investigation of the importance of helix dipole and capping effects in helical protein design. In *Biology*. California Institute of Technology, Pasadena.

27. Mayor, U., Johnson, C.M., Daggett, V., and Fersht, A.R. 2000. Protein folding and unfolding in microseconds to nanoseconds by experiment and simulation. *Proc Natl Acad Sci U S A* **97:** 13518-13522.

28. Semisotnov, G.V., Rodionova, N.A., Razgulyaev, O.I., Uversky, V.N., Gripas, A.F., and Gilmanshin, R.I. 1991. Study of the "molten globule" intermediate state in protein folding by a hydrophobic fluorescent probe. *Biopolymers* **31:** 119-128.

29. Bhattacharjya, S., and Balaram, P. 1997. Hexafluoroacetone hydrate as a structure modifier in proteins: characterization of a molten globule state of hen egg-white lysozyme. *Protein Sci* **6:** 1065-1073.

30. Nilges, M., Macias, M.J., O'Donoghue, S.I., and Oschkinat, H. 1997. Automated NOESY interpretation with ambiguous distance restraints: the refined NMR solution structure of the pleckstrin homology domain from beta-spectrin. *J Mol Biol* **269:** 408-422.

31. Ledneva, R.K., Alekseevskii, A.V., Vasil'ev, S.A., Spirin, S.A., and Kariagina, A.S. 2001. [Structural aspects of homeodomain interactions with DNA]. *Mol Biol (Mosk)* **35:** 764-777.

32. Desjarlais, J.R., and Handel, T.M. 1995. De novo design of the hydrophobic cores of proteins. *Protein Sci* **4:** 2006-2018.

33. Lazar, G.A., Desjarlais, J.R., and Handel, T.M. 1997. De novo design of the hydrophobic core of ubiquitin. *Protein Sci* **6:** 1167-1178.

34. Koehl, P., and Levitt, M. 1999. De novo protein design. I. In search of stability and specificity. *J Mol Biol* **293:** 1161-1181.

35. Kuhlman, B., O'Neill, J.W., Kim, D.E., Zhang, K.Y., and Baker, D. 2002. Accurate computer-based design of a new backbone conformation in the second turn of protein L. *J Mol Biol* **315:** 471-477.

36. Street, A.G., and Mayo, S.L. 1999. Computational protein design. *Structure Fold Des* **7:** R105-109.

37. Gordon, D.B., Marshall, S.A., and Mayo, S.L. 1999. Energy functions for protein design. *Curr Opin Struct Biol* **9:** 509-513.

38. Prodromou, C., and Pearl, L.H. 1992. Recursive PCR: a novel technique for total gene synthesis. *Protein Eng* **5:** 827-829.

39. Johnson, B.H., and Hecht, M.H. 1994. Recombinant proteins can be isolated from E. coli cells by repeated cycles of freezing and thawing. *Biotechnology (N Y)* **12:** 1357-1360.

40. Santoro, M.M., and Bolen, D.W. 1988. Unfolding free energy changes determined by the linear extrapolation method. 1. Unfolding of phenylmethanesulfonyl alpha-chymotrypsin using different denaturants. *Biochemistry* **27:** 8063-8068.

41. Delaglio, F., Grzesiek, S., Vuister, G.W., Zhu, G., Pfeifer, J., and Bax, A. 1995. NMRPipe: a multidimensional spectral processing system based on UNIX pipes. *J Biomol NMR* **6:** 277-293.

42. Johnson, B.A., and Blevins, R.A. 1994. NMR View - A computer program for the visualization and analysis of NMR data. *J Biomol NMR* **4:** 603-614.

43. Cornilescu, G., Delaglio, F., and Bax, A. 1999. Protein backbone angle restraints from searching a database for chemical shift and sequence homology. *J Biomol NMR* **13:** 289-302.

44. Koradi, R., Billeter, M., and Wuthrich, K. 1996. MOLMOL: a program for display and analysis of macromolecular structures. *J Mol Graph* **14:** 51-55, 29-32.

45. Laskowski, R.A., Rullmannn, J.A., MacArthur, M.W., Kaptein, R., and Thornton, J.M. 1996. AQUA and PROCHECK-NMR: programs for checking the quality of protein structures solved by NMR. *J Biomol NMR* **8:** 477-486.

Table IV-1. Thermodynamic data of variants and wild type.

Thermodynamic data[a]			
	Wild type	FSM1_VF	FSM1_MC
ΔG_{unfold} (kcal/mol)	1.9[b]	4.2	4.2
T_m (°C)	43[c]	>99	>99
m value[d] (kcal/mol M)	0.8[b]	1.3	1.2
C_m (M)[e]	1.5[b]	3.2	3.5

[a] All data were collected with protein in 50 mM phosphate, pH 5.5 unless noted. ΔG_{unfold} was calculated from experiments performed at 25°C using guanidinium hydrochloride denaturation.
[b] Mayor et al.[27] (done at pH 5.8 at 25°C using urea denaturation).
[c] Morgan[26] (done in 5 mM phosphate buffer, pH 4.5).
[d] Slope of ΔG_{unfold} *versus* denaturant concentration.
[e] Midpoint of unfolding transition.

Table IV-2. NMR structure statistics.

NMR structure statistics[a]	
Summary of restraints	
NOE distance restraints	1245
Unambiguous	953
Ambiguous	292
Hydrogen bonds[b]	19
Dihedral angle (ϕ,ψ) restraints[c]	57
R.m.s. deviation from restraints	
NOE restraints (Å)	0.024 ± 0.004
Dihedral restraints (°)	0.26 ± 0.12
R.m.s. deviation from idealized geometry	
Bonds (Å)	0.0037 ± 0.0002
Angles (°)	0.53 ± 0.03
Improper (°)	1.57 ± 0.14
Ensemble atomic r.m.s. deviations from mean structure[d] **(Å)**	
Backbone	0.59
All heavy	1.29
Ensemble Ramachandran statistics[e]	
Residues in most-favored region (%)	83.2
Additionally allowed region (%)	13.4
Generously allowed region (%)	2.3
Disallowed region (%)	1.1

[a] Statistics calculated for the ensemble of 43 structures (out of 100 calculated in ARIA[30]) which had no NOE restraint violations >0.5 Å and no dihedral restraint violations >5°.

[b] Each hydrogen bond yields two experimental restraints.

[c] Dihedral angle restraints were derived from HNHA analysis and chemical shift analysis with TALOS[43]. ψ restraints based on TALOS results were included if the HNHA and TALOS results were in agreement for the corresponding ϕ and if the residue was found to be in a helical conformation in structures calculated in the absence of angle restraints.

[d] Ensemble precision was calculated for residues 3–45.

[e] Ramachandran analysis was performed with Procheck.[45]

```
                                                                    Simulation
                                                                      energy
             ----|----1----|----2----|----3----|----4----|----5-  (kcal mol^-1)
Wild type    TAFSSEQLARLKREFNENRYLTERRRQQLSSELGLNEAQIKIWFQNKRAKI     -117.7
FSM1_VF      KQW|ENVEEK||EFVKRHQRI|QEELH|YAQR||||EA|RQF|EEFEQRK     -225.0
FSM1_MC      KQW|E|VERK||EFVRRHQEI|QETLHEYAQK|||QQA|EQF|REFEQRK     -223.4
```

Figure IV-1. Sequence alignment and simulation energies of the wild-type ENH sequence and the designed variants FSM1_VF and FSM1_MC. Positions that have the same identity as the wild type are indicated with a bar. FSM1_MC has 40 mutations and FSM1_VF has 39 mutations, differing from the wild-type sequence by 79% and 77%, respectively. FSM1_MC and FSM1_VF have all but 11 residues in common.

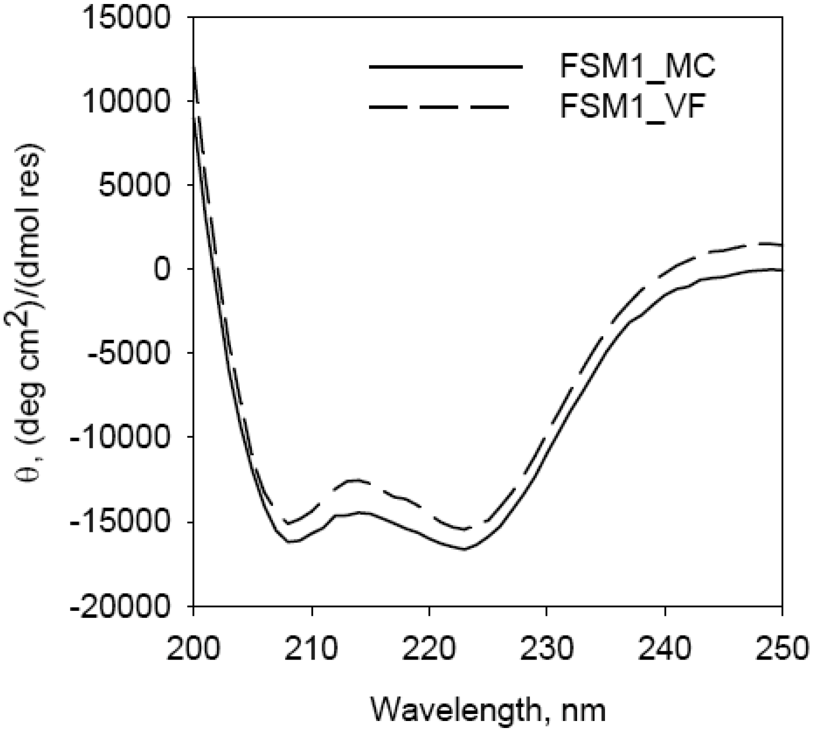

Figure IV-2. Far-UV wavelength spectra of FSM1_VF and FSM1_MC. Spectra were obtained at 25°C in 50 mM phosphate buffer at pH 5.5.

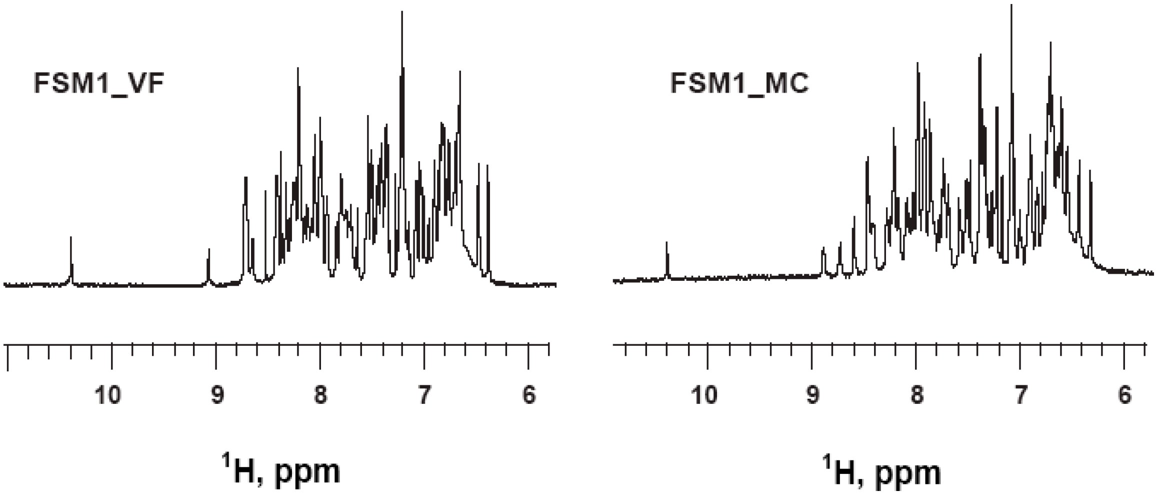

Figure IV-3. 1D ^{1}H NMR spectra of FSM1_VF and FSM1_MC. For clarity, only the amide region is shown. The sharp, dispersed lines are characteristic of well-folded proteins.

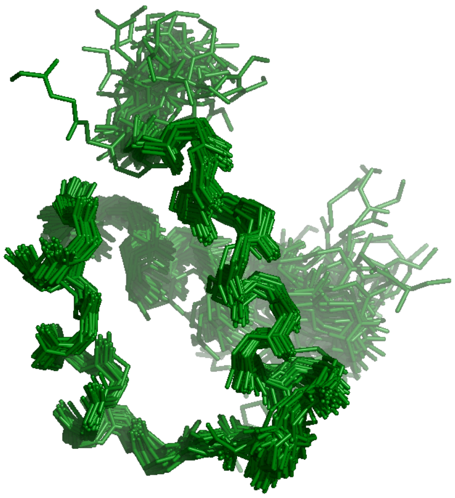

Figure IV-4. Stereoview of the FSM1_VF structure ensemble. Best-fit superposition of 43 simulated annealing structures, showing the backbone. The N terminus is located at the top of the image.

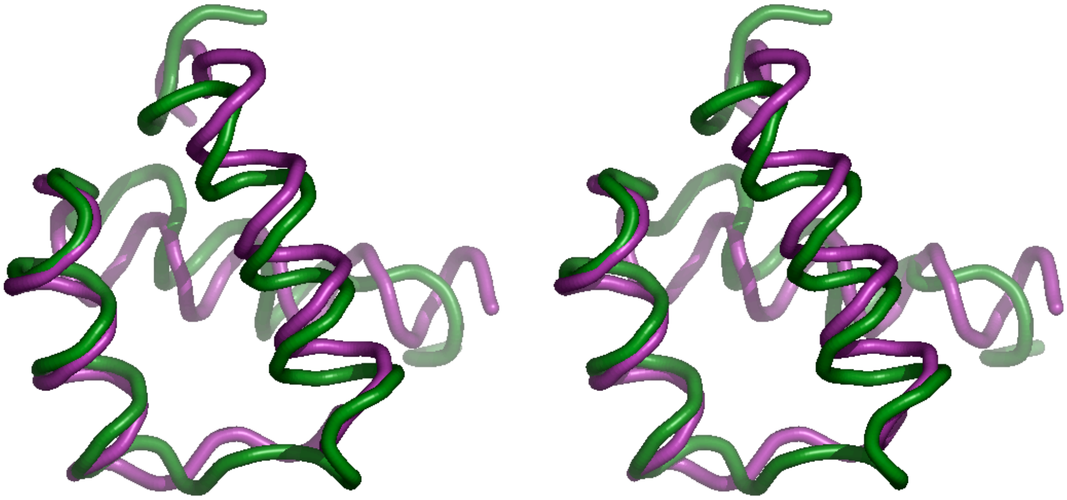

Figure IV-5. Superposition of FSM1_VF with crystal structure. Stereoview of the backbones of FSM1_VF (green) and the crystal structure of ENH (purple). The r.m.s. deviation of C_α atoms of residues 3–45 is 2.5 Å.

Chapter V

Dioxane contributes to the altered conformation and oligomerization state of a designed engrailed homeodomain variant

The text of this chapter has been adapted from a published manuscript that was coauthored with J. Kyle Lassila, Leonard M. Thomas, and Professor Stephen L. Mayo.

G. K. Hom, J. K. Lassila, L. M. Thomas, and S. L. Mayo. 2005. *Protein Sci.*, **14:** 1115–1119.

Abstract

Our goal was to compute a stable, full-sequence design of the *Drosophila melanogaster* engrailed homeodomain. Thermal and chemical denaturation data indicated the design was significantly more stable than the wild-type protein. The data were also nearly identical to those for a similar, later full-sequence design, which was shown by NMR to adopt the homeodomain fold: a three-helix, globular monomer. However, a 1.65 Å crystal structure of the design described here turned out to be of a completely different fold: a four-helix, rodlike tetramer. The crystallization conditions included ~25% dioxane, and subsequent experiments by circular dichroism and sedimentation velocity analytical ultracentrifugation indicated that dioxane increases the helicity and oligomerization state of the designed protein. We attribute at least part of the discrepancy

between the target fold and the crystal structure to the presence of a high concentration of dioxane.

Introduction

The original purpose of this project was to computationally design an amino acid sequence that stably adopts the homeodomain fold. The target fold was the same as for previous homeodomain designs from our lab[1,2]: a 51-residue, crystallographically well-defined fragment of the *Drosophila melanogaster* engrailed homeodomain **[Fig. V-1(a)]**.[3] This fragment is a globular, three-helix monomer. As in our previous homeodomain designs, we did not consider DNA binding but rather focused on protein stability.

We designed two sequences, UMC and UVF.[4] UMC was obtained via a Monte Carlo algorithm.[5] UVF has a slightly lower computed energy and could be obtained via either the Vegas[6] or the FASTER[7] algorithm.

UMC and UVF have 79% sequence identity and also have nearly identical thermal and chemical denaturation profiles. For both proteins, the melting temperature is >99°C and $\Delta G_{unfolding}$ is 4.2 kcal/mol. The one-dimensional ^{1}H NMR spectra of the proteins display the characteristics expected of well-folded proteins, and the NMR-determined structure of UVF matches the homeodomain fold.[4]

The above evidence indicated that UMC also adopts the homeodomain fold. However, a crystal structure of UMC would give direct confirmation of the overall fold and allow for a detailed comparison of crystallographic and computed side-chain conformations, which would provide critical data for improving our protein design algorithm.[8]

Here we report a 1.65 Å crystal structure of UMC. The structure is a rodlike, four-helix tetramer **[Fig. V-1(b)]**, not the expected globular, three-helix monomer. This discrepancy could be due to a lack of explicit negative design in our design algorithm; however, because of the similarity of UMC to the successful UVF design, we investigated if the crystallographic conditions could be responsible for the discrepancy. In particular, the role of dioxane was examined.

Results

The crystal structure of UMC was determined by using single wavelength anomalous diffraction (SAD). Crystallographic statistics are shown in **Table V-1**. The asymmetric unit contains four UMC molecules forming an antiparallel helical bundle with one UMC molecule per helix. Main-chain and side-chain density could not be interpreted for some terminal residues (residues 1–3 of chain A; 1–4, 51–52 of chain B; 1–2, 47–52 of chain C; and 1–4, 51–52 of chain D). The asymmetric unit also contains 2 cadmium atoms, 1 acetate molecule and 10 dioxane molecules. The cadmium atoms are each coordinated by four carboxylate anions: one cadmium is coordinated by four glutamate side chains **[Fig. V-1(c)]**, and the other cadmium is coordinated by three glutamate side chains and an acetate molecule.

Several dioxane molecules mediate helix-helix packing **[Fig. V-1(d)]**. This observation led us to examine the effect of dioxane on the helicity and oligomerization of UMC in solution.

Helicity was examined by far-UV circular dichroism. Ellipticity was virtually unchanged when UMC was exposed to $CdCl_2$ alone (not shown) or $CdCl_2$ with 10%

dioxane **[Fig. V-2(a)]**. However, exposure of UMC to 20% dioxane lowered the minima at 208 and 222 nm and was thus indicative of an increase in helicity. The increase, while significant, was still less than that for 30% trifluoroethanol (TFE), a helix stabilizer.[9] Higher percentages of dioxane did not further increase the helicity significantly (not shown).

Oligomerization was examined by sedimentation velocity analytical ultracentrifugation. Exposure to 20% dioxane significantly decreased the percentage of monomeric UMC, from 81.4% to 62.8%, and concomitantly increased the percentage of dimeric UMC, from 14.8% to 36.3% **[Fig. V-2(b)]**. The frictional ratio, which describes the shape of the sedimenting species, also increased. A sphere has a ratio of ~1.2, whereas rodlike shapes have higher ratios. The frictional ratio increased from 1.22 to 1.42 in the presence of dioxane.

Discussion

Our crystal structure of UMC is quite dissimilar to the target homeodomain fold. Instead of three short helices, each monomer is a single long helix. However, the crystallization conditions, especially the high concentration of dioxane, may induce UMC into a conformation unrepresentative of UMC in solution.

Increased helicity and oligomerization due to dioxane

Dioxane increased the helicity of UMC. While dioxane had a significant effect, the $[\theta]_{222}$ for 20% dioxane (−25,000 deg cm^2 / dmol res) was less negative than for 30% TFE (−31,000 deg cm^2 / dmol res).

The effect of dioxane on increasing helicity has been reported previously.[10-12] The increase in helicity can be explained entropically: nonpolar solvent increases the entropic cost of forming protein hydrogen bonds to water and thus decreases the relative cost of forming helical hydrogen bonds. The use of organic solvents may have played a role in the crystallization of a number of short aminoisobutyric acid–containing peptides, which also adopt extended continuous helical structures.[13]

The sedimentation velocity data showed that dioxane increases the oligomerization state and frictional ratio of UMC. While there was a significant increase in the amount of dimer, there was no evidence of a tetramer, as might be expected from the crystal structure. One explanation is that formation of a tetrameric species requires cadmium. Although $CdCl_2$ alone and $CdCl_2$ with 10% dioxane had no effect on the helicity of UMC, low millimolar amounts of $CdCl_2$ (e.g., 2–5 mM) caused essentially all UMC to precipitate out in the presence of >15% dioxane. The UMC crystals appeared a couple of weeks after precipitate had formed in the well. Perhaps cadmium further increases the dioxane-induced helicity and/or oligomerization of UMC but requires the very slow mixing that occurs in the crystallization well.

Conclusion

Overall, the crystal structure has increased helicity and altered oligomerization compared to the target fold. Both of these differences were inducible by dioxane. We thus attribute at least part of the discrepancy between the target fold and the crystal structure of the designed sequence to the presence of a high concentration of dioxane. Although low concentrations (1%–2%) of dioxane have been reported to improve crystallization of some proteins,[14,15] we suggest that high concentrations of dioxane be used with caution.

Materials and methods

Protein design and purification

The UMC design, construction, expression, and purification were similar to our previous engrailed designs[1,2] and are described in detail elsewhere.[4] A brief summary is below.

The starting model for all engrailed designs was Protein Data Bank (PDB) entry 1enh.[3] Because residue 35 has a positive ϕ angle, it was preserved as glycine. The UMC design protocol was identical to the B6 design protocol of Marshall and Mayo,[1] except that in the UMC design all residues were designed simultaneously, and a Monte Carlo simulation[5] was used instead of a dead-end elimination–based algorithm[16] to find a low-energy sequence. The protein was expressed in *Escherichia coli* and purified via freeze-thaw[17] followed by HPLC using an acetonitrile/water gradient containing 0.1% TFA. Mass spectrometry indicated UMC has an N-terminal methionine.

Crystallization

Crystals were obtained using a modified sitting drop method that utilizes a "reservoir mimic." The well reservoir is minimized to contain only the volatile reagents and NaCl. The nonvolatile reagents normally in the reservoir are kept in a separate solution (the mimic) that is only added to the crystallization drop.

We also used Fluorinert (Hampton Research), which is expected to be denser than the drop and allow the drop to float. Under our conditions, Fluorinert floated above the drop. However, this serendipitously slowed the otherwise rapid crystal degradation that would happen upon well opening and was presumably due to the volatility of dioxane.

The initial crystallization condition was 35% dioxane (Hampton Research Crystal Screen 2). The final crystallization conditions were as follows: the well reservoir contained 500 μL of either 24% or 25% dioxane; the well post contained 1 μL of protein solution (~17 mg/mL UMC, 50 μM sodium citrate at pH 5.5) followed by 1 μL of reservoir mimic solution (0.1 M MES at pH 5.7, 30% PEG 400, 10–15 mM $CdCl_2$); and 20 μL of Fluorinert (Hampton Research) was then added on top of each post. Trays were incubated at 20°C; crystals appeared after about 2 wk. The largest crystals had dimensions of ~150 × 150 × 200 μm.

Structure determination

Data were collected using a Cu source on a Rigaku RU3HR generator with an R–AXIS IV detector at 100 K. Data were processed using the HKL program suite v1.97.9.[18] Initial electron density maps were generated by using SAD phasing as implemented in the program suite ELVES.[19] The final model was determined by subsequent rounds of building and refinement using O[20] and REFMAC[21] from the CCP4 program suite[22] to an R-factor of 22.2% (R_{free} = 27.8%). Final refinement was done with high resolution data collected at beamline 9.2 at the Stanford Synchrotron Radiation Laboratory and produced a final R-factor of 18.7% (R_{free} = 22.7%).

Coordinates and structure factors have been deposited in the PDB under the accession code 1Y66.

Circular dichroism and sedimentation velocity

Circular dichroism data were collected on an Aviv 62DS spectrometer equipped with a thermoelectric unit. Wavelength scans were done from 190-250 nm at 20°C in a 0.1 mm path-length cell. All samples contained 532 µM UMC and 10 mM sodium citrate (pH 5.5). Protein concentration was determined by absorbance at 280 nm in the presence of 8 M guanidine HCl.

Sedimentation velocity data were collected on a Beckman XL-I analytical ultracentrifuge with interference optics. Samples contained 532 µM UMC and 0.1 M sodium citrate (pH 5.5). Samples were dialyzed for 3 h at room temperature against ~100 mL of the corresponding solution without protein. A 12 mm Epon centerpiece and sapphire windows were used. The rotor, an An-60 Ti, was spun at 55,000 RPM at 25°C. Scans were taken every 5 min for ~15 h. Data were analyzed with SEDFIT.[23]

Acknowledgments

We are grateful to Premal Shah, Rhonda Digiusto, Scott Ross, and Karin Crowhurst for assistance with NMR; Doug Rees, James Holton, and J.J. Plecs for assistance with crystallography; Po-Ssu Huang for assistance with sedimentation velocity experiments and PyMOL; and Marie Ary and Jessica Mao for assistance with the manuscript. This work was supported by the Howard Hughes Medical Institute, the Ralph M. Parsons Foundation, the Defense Advanced Research Projects Agency, and an IBM Shared University Research Grant. Portions of this research were carried out at the Stanford Synchrotron Radiation Laboratory (SSRL), a national user facility operated by Stanford University on behalf of the U.S. Department of Energy, Office of Basic Energy Sciences.

The SSRL Structural Molecular Biology Program is supported by the Department of Energy, Office of Biological and Environmental Research, and by the NIH, National Center for Research Resources, Biomedical Technology Program, and the National Institute of General Medical Sciences. We thank the Gordon and Betty Moore Foundation for their support of the crystallographic resources of the Molecular Observatory for Structural Molecular Biology used in this study.

References

1. Marshall, S.A., and Mayo, S.L. 2001. Achieving stability and conformational specificity in designed proteins via binary patterning. *J. Mol. Biol.* **305:** 619-631.
2. Marshall, S.A., Morgan, C.S., and Mayo, S.L. 2002. Electrostatics significantly affect the stability of designed homeodomain variants. *J. Mol. Biol.* **316:** 189-199.
3. Clarke, N.D., Kissinger, C.R., Desjarlais, J., Gilliland, G.L., and Pabo, C.O. 1994. Structural studies of the engrailed homeodomain. *Protein Sci.* **3:** 1779-1787.
4. Shah, P.S., Hom, G.K., Ross, S.A., and Mayo, S.L. 2004. Thermodynamic and structural characterization of full sequence designs. *In preparation.*
5. Voigt, C.A., Gordon, D.B., and Mayo, S.L. 2000. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *J. Mol. Biol.* **299:** 789-803.
6. Shah, P.S., Hom, G.K., and Mayo, S.L. 2004. Preprocessing of rotamers for protein design calculations. *J. Comput. Chem.* **25:** 1797-1800.

7. Desmet, J., Spriet, J., and Lasters, I. 2002. Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins* **48:** 31-43.

8. Dahiyat, B.I., and Mayo, S.L. 1996. Protein design automation. *Protein Sci.* **5:** 895-903.

9. Rohl, C.A., Chakrabartty, A., and Baldwin, R.L. 1996. Helix propagation and N-cap propensities of the amino acids measured in alanine-based peptides in 40 volume percent trifluoroethanol. *Protein Sci.* **5:** 2623-2637.

10. Iizuka, E., and Yang, J.T. 1965. Effect of salts and dioxane on the coiled conformation of poly-L-glutamic acid in aqueous solution. *Biochemistry* **4:** 1249-1257.

11. Tanford, C., De, P.K., and Taggart, V.G. 1960. The role of the α-helix in the structure of proteins. Optical rotatory dispersion of β-lactoglobulin. *J. Am. Chem. Soc.* **82:** 6028-6034.

12. Urnes, P., and Doty, P. 1961. Optical rotation and the conformation of polypeptides and proteins. In *Advances in Protein Chemistry*. (eds. C.B. Anfinsen Jr., M.L. Anson, K. Bailey, and J.T. Edsall), pp. 401-543. Academic Press, New York and London.

13. Karle, I.L. 1992. Folding, aggregation and molecular recognition in peptides. *Acta Crystallogr B* **48 (Pt 4):** 341-356.

14. Matthews, B.W., Sigler, P.B., Henderson, R., and Blow, D.M. 1967. Three-dimensional structure of tosyl-alpha-chymotrypsin. *Nature* **214:** 652-656.

15. Sigler, P.B., Jeffery, B.A., Matthews, B.W., and Blow, D.M. 1966. An x-ray diffraction study of inhibited derivatives of alpha-chymotrypsin. *J Mol Biol* **15:** 175-192.

16. Gordon, D.B., Hom, G.K., Mayo, S.L., and Pierce, N.A. 2003. Exact rotamer optimization for protein design. *J. Comput. Chem.* **24:** 232-243.

17. Johnson, B.H., and Hecht, M.H. 1994. Recombinant proteins can be isolated from E. coli cells by repeated cycles of freezing and thawing. *Biotechnology (N Y)* **12:** 1357-1360.

18. Otwinowski, Z., and Minor, W. 1997. Processing of x-ray diffraction data collected in oscillation mode. *Methods in Enzymology* **276: Macromolecular Crystallography, part A:** 307-326.

19. Holton, J., and Alber, T. 2004. Automated protein crystal structure determination using ELVES. *Proc. Natl. Acad. Sci. U S A* **101:** 1537-1542.

20. Jones, T.A., Zou, J.Y., Cowan, S.W., and Kjeldgaard. 1991. Improved methods for building protein models in electron density maps and the location of errors in these models. *Acta Crystallogr. A* **47 (Pt 2):** 110-119.

21. Murshudov, G.N., Vagin, A.A., Lebedev, A., Wilson, K.S., and Dodson, E.J. 1999. Efficient anisotropic refinement of macromolecular structures using FFT. *Acta Crystallogr. D Biol. Crystallogr.* **55 (Pt 1):** 247-255.

22. Collaborative Computational Project, N. 1994. The CCP4 suite: programs for protein crystallography. *Acta Crystallogr. D Biol. Crystallogr.* **50:** 760-763.

23. Schuck, P. 2000. Size-distribution analysis of macromolecules by sedimentation velocity ultracentrifugation and Lamm equation modeling. *Biophys. J.* **78:** 1606-1619.

Table V–1. X-ray data collection and refinement statistics.

	R-axis IV	SSRL[a]
Unit cell		
a	50.767 Å	50.712 Å
b	52.562 Å	52.646 Å
c	82.147 Å	82.182 Å
Space group	$P2_12_12_1$	$P2_12_12_1$
Wavelength	1.5418 Å	0.8265 Å
Resolution range	81.65–1.90 Å	44.32–1.65 Å
No. of reflections collected	208,991	204,302
No. of unique reflections	17,953	27,060
R_{merge}[b]	5.6% (55.1%)[c]	4.7% (19.6%)
I/σ(I)	10.1 (1.3)	31.8 (8.5)
Completeness	99.9% (99.5%)	99.8% (100.0%)
Final refinement		
R_{cryst}		18.7%
R_{free}[d]		22.7%
Figure of merit		0.863
No. of residues		368
No. of water molecules		168
No. of non-protein molecules		11
Mean B value		28.1 $Å^2$
RMSD from standard stereochemistry		
Bond length		0.017 Å
Bond angle		1.527°
Ramachandran plot statistics		
Most favored regions		99.4%
Additional allowed regions		0.6%
Generously allowed regions		0.0%
Disallowed regions		0.0%

[a] Stanford Synchrotron Radiation Laboratory.

[b] $R_{merge} = \Sigma \left| I - \langle I \rangle \right| / \Sigma (I)$, where I is the observed intensity and <I> is the average intensity.

[c] Numbers in parentheses represent values in the highest resolution shell (1.90–1.99 Å for the R-axis IV data and 1.652–1.695 Å for the SSRL data).

[d] R_{free} was calculated for 5% of randomly selected reflections excluded from refinement.

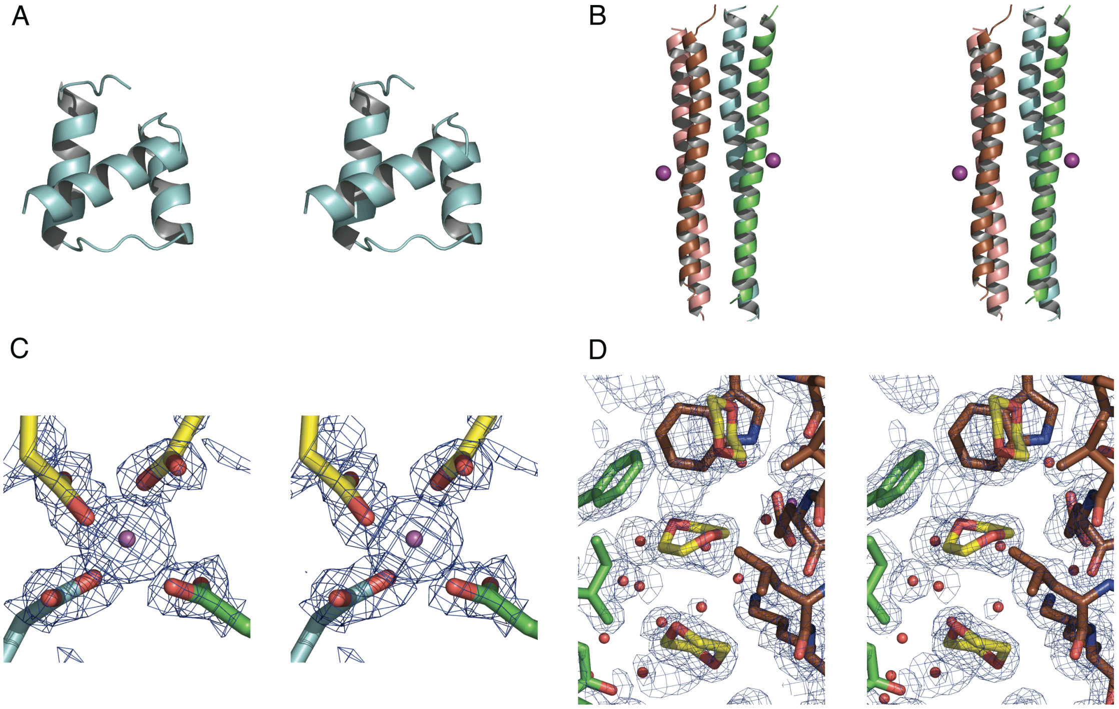

Figure V–1. (a) Target homeodomain fold for UMC. (b) Ribbon diagram of the UMC crystal structure. The coloring is cyan (chain A), green (chain B), brown (chain C), salmon (chain D), and purple (cadmium). (c) Coordination of one of the two cadmium atoms by four glutamates. The coloring is cyan (chain A), green (chain B), yellow (chain C), purple (cadmium), and red (oxygen). The σ_A-weighted density map is contoured at 2σ, up to 3.5 Å from the cadmium. Chain C is from a symmetry-related molecule of that shown in (b). (d) Dioxane molecules mediating helix-helix packing. The coloring is green (chain B), brown (chain C), yellow (dioxane), red (oxygen), and blue (nitrogen). The σ_A-weighted density map is contoured at 1σ, up to 3 Å from the dioxane. Figures were generated in PyMOL (http://www.pymol.org).

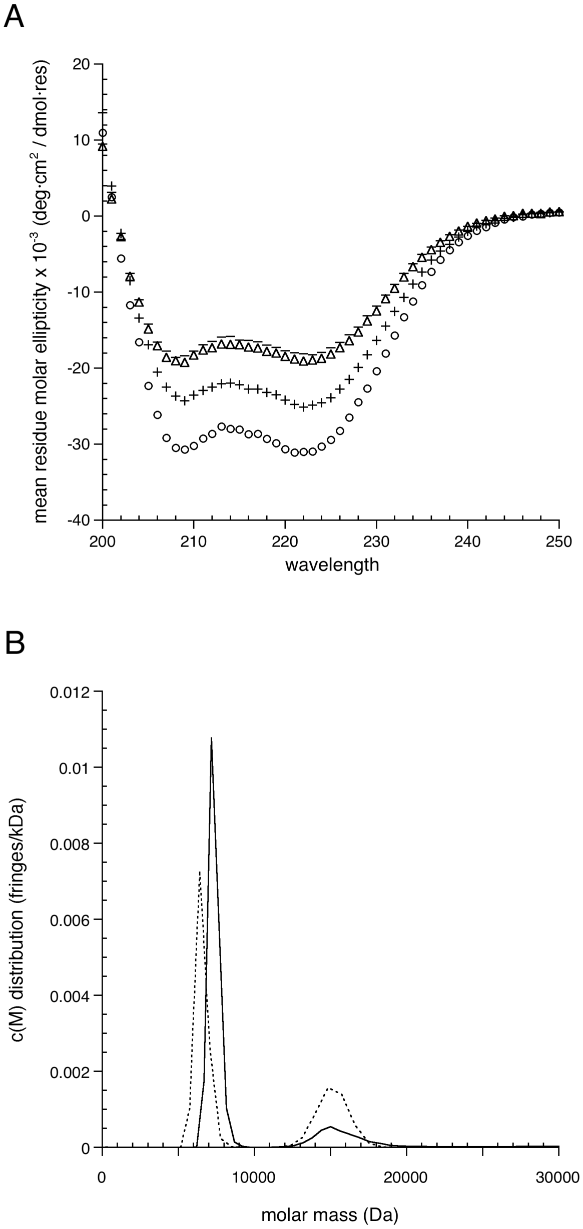

Figure V–2. (a) Far-UV circular dichroism analysis of UMC. The spectra are UMC (dashes); UMC in 5 mM $CdCl_2$ and 10% dioxane (triangles); UMC in 20% dioxane (crosses); and UMC in 30% TFE (circles). (b) Molar mass distribution of UMC as determined by sedimentation velocity: UMC (solid line); UMC in 20% dioxane (dotted line).

Chapter VI

A search algorithm for fixed-composition protein design

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

G. K. Hom and S. L. Mayo˙ 2005. *To be submitted.*

Abstract

We present a computational protein design algorithm for finding low-energy sequences of fixed amino acid composition. The search algorithms used in protein design typically do not restrict amino acid composition. However, the random energy model of Shakhnovich suggests that the use of fixed-composition sequences may circumvent defects in the modeling of the denatured state. Our algorithm, FC_FASTER, links fixed-composition versions of Monte Carlo and the FASTER algorithm. As proof of principle, FC_FASTER was tested on an experimentally validated, full-sequence design of the β1 domain of protein G. For the wild-type composition, FC_FASTER found a lower-energy sequence than the experimentally validated sequence. Also, for a different composition, FC_FASTER found the hypothetical lowest-energy sequence in 14 out of 32 trials.

Introduction

In computational protein design, simulated energies are intended to correlate with experimental free energies of unfolding. As such, a force field should model not only the native state but also the denatured state. The denatured state is commonly assumed to have no residual structure.[1] This is a poor model, however, as theoretical and experimental data suggest that denatured proteins are often very compact, with persistent hydrophobic clustering and considerable residual secondary structure.[2] In addition, it is unclear how to efficiently model the ensemble nature of the denatured state in a way that is meaningful for protein design calculations.

Modeling of the denatured state may be circumvented if the free energies of denatured proteins are identical. According to the random energy model (REM),[3,4] the denatured-state energies should be identical for proteins of identical amino acid composition. While REM cannot be exact for proteins, it is a good approximation.[5] Thus, designs of fixed amino acid composition (FC) should enable development of more accurately tuned force fields, at least for modeling the native state. FC designs may also be useful for imposing fold specificity,[5] thus providing a partial means of negative design.

Koehl and Levitt used a simple two-position version of Monte Carlo (MC)[6] for their FC designs.[5] However, MC has failed considerably for some non-FC design classes,[7] and so MC alone may be insufficient for finding the lowest-energy FC sequences. We have had good results on non-FC designs by using a combination of MC and the FASTER[8] algorithm (B. Allen and S.L. Mayo, in prep.). Accordingly, we modified MC and FASTER for fixed-composition and linked them into a new FC algorithm, FC_FASTER.

FC_FASTER

FC_FASTER has four stages:

1) Fixed-composition MC (FC_MC)
2) Two-position minimization
3) Fixed-composition, single-position perturbation/relaxation (FC_sPR)
4) Two-position minimization

The basic strategy is to find low-energy troughs with FC_MC and then to find the minima of those troughs with FC_sPR.

FC_MC is adapted from Voigt et al.[7]; significant differences are noted below. FC_MC starts with an amino acid sequence and randomizes it for both amino acid order and side-chain conformation. (Henceforth a discrete side-chain conformation will be called a “rotamer.”) Each change, or step, is made by randomly choosing two to four positions, making a random permutation of the corresponding amino acids, and randomly choosing rotamers of those amino acids.

At the end of FC_MC, the lowest-energy sequence undergoes a two-position minimization. All FC rotamer-pair substitutions (i.e., substitutions that preserve the fixed composition) are tried, in a random order. If a substitution results in a new lowest-energy sequence, then the minimization is restarted on this sequence. This process continues until no FC rotamer pair will improve the lowest-energy sequence.

The lowest-energy sequence is passed to FC_sPR, which is adapted from the sPR stage of the side-chain placement algorithm FASTER.[8] FC_sPR is driven by rotamer perturbations of the lowest-energy sequence. For each perturbation, the rest of the sequence is allowed to accommodate, or relax, with the goal of finding a new lowest-energy sequence. The relaxation in FC_sPR occurs iteratively: after each position relaxes, the rotamer sequence is updated prior to relaxation of the next position. The core FC_sPR process has four stages: *perturbation*, *relaxation to restore fixed composition*, *iterative side-chain placement relaxation*, and *adoption/rejection*.

1) Perturbation

At a random position, a rotamer is substituted to form a perturbed sequence. This rotamer's amino acid may be different, and thus the fixed composition may be disrupted. All sequence energies in the next stage are calculated in the background of this perturbed sequence.

2) Relaxation to restore fixed composition

If the fixed composition was disrupted, a compatible position must be found to restore the fixed composition. For example, if the perturbed position went from Arg to Lys, then a Lys position must become Arg. In this case, a Lys position's Arg rotamers are "restoring rotamers." By contrast, rotamers that maintain the amino acid identity at a position, such as a Lys position's Lys rotamers, are "conservative rotamers."

For each compatible position, the difference in sequence energy is calculated between the best restoring rotamer and the best conservative rotamer. (The "best" rotamers have the lowest sequence energies.) Thus the relative cost of switching amino acid identity is determined. The position with the best (most negative) difference is chosen, and the corresponding best restoring rotamer is substituted into the perturbed sequence.

3) *Iterative side-chain placement relaxation*

For each remaining position, the best conservative rotamer is chosen. This process is done iteratively, ordered by how strongly each position interacts with the perturbed position. Interaction strength is evaluated as follows: for all rotamer pairs between a position and the perturbed position, the rotamer pair with the maximum absolute-value interaction energy is determined, and that interaction energy is compared with those of the other positions. For the position that ranks strongest, the best rotamer is calculated in the background of the sequence from stage 2 and then substituted into that sequence. For each subsequent position, the best rotamer is calculated in the background of the most up-to-date sequence and then substituted into that sequence.

4) Adoption/rejection

The most up-to-date sequence is kept if it is of lower energy than the pre-perturbation sequence; i.e., if the most up-to-date sequence is the lowest-energy sequence so far. Otherwise the pre-perturbation sequence is kept.

Perturbation is done on the lowest-energy sequence for all rotamers at all positions; perturbation positions are chosen in random order. FC_sPR is repeated until no perturbation will improve the lowest-energy sequence.

After FC_sPR, the lowest-energy sequence again undergoes a two-position minimization. (This minimization stage, unlike the minimization after FC_MC, was never found to improve the lowest-energy sequence. However, the minimization is relatively fast and so was kept as a safety net.)

Results

As proof of principle, FC_FASTER was tested on a full-sequence design of the β1 domain of protein G (Gβ1) for two compositions. Previously, a different version of FC_MC (FC_MC_original) was tested on this design with the wild-type composition, and the resulting sequence was validated experimentally (see below). We thus tested FC_FASTER with the wild-type composition. In addition, we wanted to see if the lowest-energy sequence could be found for a given fixed composition. Accordingly, we first found the overall lowest-energy sequence for the design using a non-FC search algorithm, and then we tested FC_FASTER with that sequence's composition.

For the above-mentioned test using FC_MC_original, the lowest computed energy was –96.356 kcal/mol and occurred in 1 out of 16 trials; the average lowest energy for all trials was –94.866 **(Table VI-1)**. The lowest-energy sequence, which has 24 amino acid mutations from the wild-type sequence, was synthesized. Wavelength scans and 1D-NMR were consistent with the wild-type fold, and the molecule showed cooperative unfolding in guanidinium (O. Alvizo, personal communication).

FC_FASTER was run on the same design, also using the wild-type composition. We first tested just the first two stages of FC_FASTER: FC_MC plus minimization. The lowest energy was –97.450 and occurred in 1 out of 32 trials; the average energy was –96.400. For the full FC_FASTER algorithm, the lowest energy was also –97.450 and occurred in 1 out of 32 trials; the average energy was –96.571. The full FC_FASTER algorithm (0.9 h/trial) took only slightly longer than FC_MC plus minimization (0.8 h/trial). The lowest-energy sequence has 14 amino acid differences from the synthesized sequence above.

Ideally, FC_FASTER would find the lowest-energy sequence for a given fixed composition, but how to evaluate this is unclear. Indirect evidence might be the occurrence of the same lowest-energy sequence in multiple trials. A more rigorous test could be done if the lowest-energy sequence could be found by other means. The overall lowest-energy sequence *irrespective of composition* would work, because that sequence is also the lowest-energy sequence for its composition. To find the overall lowest-energy sequence, our lab typically uses either the HERO algorithm[9] or a version of FASTER modified for protein design (B. Allen and S. L. Mayo, in prep.). If HERO converges, it will find the lowest-energy sequence; FASTER has found the lowest-energy sequence in all cases we could verify.

Both HERO and FASTER were tried on the design calculation. HERO failed to converge. FASTER's lowest-energy sequence, Best_FASTER, had an energy of –190.996. FC_FASTER was then run using the composition of Best_FASTER, with FC_MC plus minimization being tested first. For FC_MC plus minimization, the lowest energy was –190.830 and occurred in 1 out of 32 trials; the average was –188.930. For the full FC_FASTER algorithm, the lowest energy was –190.996 (the Best_FASTER sequence) and occurred in 14 out of 32 trials; the average energy was –189.770.

Discussion

FC_FASTER found both a lower-energy sequence for an experimentally validated design and also the hypothetical lowest-energy sequence for a different composition. Using just FC_MC with two-position minimization also worked well. However, the addition of FC_sPR in the full FC_FASTER algorithm required relatively little time and was especially better for the Best_FASTER composition, for which FC_FASTER found the lowest-energy sequence with significant frequency.

For the wild-type composition, the synthesized sequence had a computed energy of –96.356, but FC_FASTER found a lower sequence of energy –97.450. While that energy difference may seem insignificant, it belies significant differences in sequence. The lower-energy sequence differs in 14 (out of 56) positions. Also, an FC_MC search showed that at least 1000 amino acid sequences have energies between those of the two sequences above (data not shown). Both directed evolution and non-computational rational methods would be hard-pressed to derive the lower-energy sequence from the synthesized sequence.

Each run of FC_MC_original took ~33 h, compared to only ~1 h for each run of FC_FASTER. This speedup is misleading because just FC_MC with minimization performed well in 1 h. The increase in performance for FC_MC with minimization was attributed primarily to better MC parameterization, necessitating fewer MC cycles.

Incorporation of other FASTER components did not improve FC_FASTER. The iBR and ciBR stages of the original FASTER were modified for fixed-composition and tested after FC_MC and minimization. In some cases, after one of these stages, the lowest and average energies would improve. However, after subsequent FC_sPR and minimization, the lowest and average energies were often significantly worse than those from regular FC_FASTER (data not shown).

Future improvements to FC_FASTER may include optimizing the high temperature in FC_MC, optimizing the number of steps per cycle, saving more sequences for minimization, and alternating more frequently between FC_MC, FC_sPR, and minimization.

Methods

Physical model and test case

Many of the force-field potentials and parameters have been previously described;[1,10-13] changes are noted below. The side-chain/side-chain hydrogen bond well depth was 4.0 kcal/mol. Side-chain/side-chain hydrogen bonds were not allowed at surface positions, and side-chain/backbone hydrogen bonds between immediate neighbors (+1 or –1 positions) were scaled by 0.25. The LK solvation model was used with the published parameter set.[14] In order to balance the solvation energy with other force field terms, the polar desolvation energy was scaled by 0.6.

The starting model for Gβ1 was PDB code 1pga.[15] A backbone-dependent rotamer library[16] was used with expansion of aromatic and hydrophobic residues by one standard deviation about their χ_1 and χ_2 values. The library also included a rotamer for the wild-type conformation of Leu7. To incorporate rotamer probabilities from the library, [–0.3][log(p)] was added to the energy for each rotamer, where p is the probability for that rotamer. Residues were classified into core, boundary, or surface positions by an automated algorithm.[10] The Met position (1) was allowed to change conformation but not amino acid identity, and the Gly positions (9, 14, 38 and 41) were not changed. At all other positions, the amino acids found in the wild-type protein were allowed: A, D, E, F, G, I, K, L, M, N, Q, T, V, W and Y.

Algorithm Parameters

The FC_MC component was run for two cycles of 10^7 steps/cycle, with a high temperature of 500.0 and a low temperature of 150.0. All calculations were run on IBM PowerPC 970 processors running at 1.6 GHz.

Acknowledgements

We are grateful to Ben Allen for help with FASTER, and to Oscar Alvizo for sharing his results on Gβ1 prior to publication and for providing the test file. We also thank them and Marie Ary, Po-Ssu Huang, and Christina Vizcarra for assistance with the manuscript. This work was supported by the Howard Hughes Medical Institute, the Ralph M. Parsons Foundation, and an IBM Shared University Research Grant.

References

1. Gordon, D.B., Marshall, S.A., and Mayo, S.L. 1999. Energy functions for protein design. *Curr Opin Struct Biol* **9:** 509-513.
2. Dill, K.A., and Shortle, D. 1991. Denatured states of proteins. *Annu Rev Biochem* **60:** 795-825.
3. Shakhnovich, E.I., and Gutin, A.M. 1993. A new approach to the design of stable proteins. *Protein Eng* **6:** 793-800.

4. Shakhnovich, E.I., and Gutin, A.M. 1993. Engineering of stable and fast-folding sequences of model proteins. *Proc Natl Acad Sci U S A* **90:** 7195-7199.

5. Koehl, P., and Levitt, M. 1999. De novo protein design. I. In search of stability and specificity. *J Mol Biol* **293:** 1161-1181.

6. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., and Teller, A.H. 1953. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics* **21:** 1087-1092.

7. Voigt, C.A., Gordon, D.B., and Mayo, S.L. 2000. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *J. Mol. Biol.* **299:** 789-803.

8. Desmet, J., Spriet, J., and Lasters, I. 2002. Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins* **48:** 31-43.

9. Gordon, D.B., Hom, G.K., Mayo, S.L., and Pierce, N.A. 2003. Exact rotamer optimization for protein design. *J. Comput. Chem.* **24:** 232-243.

10. Dahiyat, B.I., and Mayo, S.L. 1997. De novo protein design: fully automated sequence selection [see comments]. *Science* **278:** 82-87.

11. Dahiyat, B.I., and Mayo, S.L. 1996. Protein design automation. *Protein Sci.* **5:** 895-903.

12. Street, A.G., and Mayo, S.L. 1999. Computational protein design. *Structure Fold Des* **7:** R105-109.

13. Street, A.G., and Mayo, S.L. 1998. Pairwise calculation of protein solvent-accessible surface areas. *Fold Des* **3:** 253-258.

14. Lazaridis, T., and Karplus, M. 1999. Effective energy function for proteins in solution. *Proteins* **35:** 133-152.

15. Gallagher, T., Alexander, P., Bryan, P., and Gilliland, G.L. 1994. Two crystal structures of the B1 immunoglobulin-binding domain of streptococcal protein G and comparison with NMR. *Biochemistry* **33:** 4721-4729.

16. Dunbrack, R.L., Jr., and Cohen, F.E. 1997. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Sci* **6:** 1661-1681.

Table VI-1. Algorithm results for Gβ1 designs of differing amino acid composition.

	Lowest energy		Average energy[b]	Time
Wild-type composition	(kcal/mol)	Frequency[a]	(kcal/mol)	(h)
FC_MC_original	–96.356	1/16	–94.866	33.3
FC_MC + minimization	-97.450	1/32	-96.400	0.8
FC_FASTER	-97.450	1/32	-96.571	0.9
No fixed composition				
FASTER	–190.996			
Best_FASTER composition[c]				
FC_MC + minimization	–190.830	1/32	-188.930	0.8
FC_FASTER	–190.996	14/32	-189.770	1.0

[a](Number of trials with the overall lowest energy) / (Number of trials in total)

[b]Average of the lowest energy from each trial

[c]Composition of the lowest-energy sequence from FASTER