Development and Validation of Optimization Methods for the Design of Protein Sequences and Combinatorial Libraries

Thesis by
Benjamin D. Allen

In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

California Institute of Technology
Pasadena, California
2009
(Defended February 19, 2009)

Acknowledgements

- My adviser: Steve.
- My committee: Steve, Doug, Dave, and Frances.
- My lab-mates.
- My parents.
- Amie, and Orson.

Abstract

To facilitate the design of protein sequences with desired properties, simulation techniques have been developed to allow large portions of amino acid sequence space to be evaluated by computer. These computational protein design methods apply optimization algorithms to sort through the enormity of sequence space and find desirable variants.

Simple modifications to the stochastic optimization algorithm FASTER enhanced its performance by two orders of magnitude without loss of accuracy, and rendered it more efficient than its major competitor by a factor of 10. These improvements allowed higher-quality amino acid solutions to be found more quickly, and accelerated the pace at which users could perform cycles of design and model adjustment.

This success prompted research into techniques for a protein design formulation that allows simulation in the context of multiple states simultaneously. This multi-state design can be used to wield explicit control over structural, binding, or catalytic specificity, and changes the scope of design goals that can be addressed by computation. Evaluation of multi-state FASTER indicated that it performed radically better than its major competitor in a variety of design contexts, and that in most cases it found solutions better than those that could ever be found using a lesser method.

Multi-state optimization using FASTER was applied to test the influence of various types of input structural data on the design of a small protein. To facilitate this evaluation, methods for the design and high-throughput stability screening of combinatorial libraries were developed. Screening of libraries based on single structures and structural ensembles indicated the success of multi-state modeling. Our results also

suggested that the exhaustive screening of designed libraries can help to elucidate the origins of design model failures. Finally, they showed that success of a design procedure does not hinge on its ability to correlate experimental and simulated measures of fitness, and prompted greater consideration of design methods that target explicitly conformational specificity.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 3.

CHAPTER 4.

APPENDIX I.

LIST OF TABLES

CHAPTER 2.

CHAPTER 3.

CHAPTER 4.

APPENDIX I.

Chapter 1

Optimization Strategies for the Design of Protein Sequences and Combinatorial Mutation Libraries

Some language in this chapter was adapted from manuscripts coauthored with Christina L. Vizcarra, Oscar Alvizo, and Stephen L. Mayo.

Alvizo, O.; Allen, B. D.; Mayo, S. L., Computational protein design promises to revolutionize protein engineering. *Biotechniques* **2007,** *42* (1), 31–39.

Vizcarra, C. L.; Allen, B. D.; Mayo, S. L., Progress and challenges in computational protein design. Submitted **2008**.

High-throughput protein engineering

Proteins are linear heteropolymers, built from 20 standard amino acid monomers. Synthesized inside cells, they perform a vast majority of the structural, catalytic, sensory, and regulatory functions that characterize living systems as we understand them today. These myriad roles are made possible by the ability of proteins to self-assemble into well-defined structures specified by their amino acid sequences. Although only a small fraction of all possible protein sequences can assume a folded, functional form,[1, 2] the modular nature of the protein platform allows existing functions to be altered and enhanced for new or fluctuating requirements by selection for fitness from a heterogeneous population. The requisite diversity develops though copying errors and recombination of nucleic acid sequences that encode the proteins expressed by members of the population. This process is slow and relies on serendipity to discover beneficial variants. Nevertheless, it represents the only plausible description of how the complexity of life we see today could have ultimately arisen from simpler chemical systems.

Before sufficiently powerful tools for genetic manipulation were even available, it was postulated that existing biological systems for protein fabrication could be harnessed to produce nano-scale molecular machines with designed functions.[3] Major successes along these lines have been achieved via directed evolution, in which screening or *in vivo* selection is applied to isolate active variants from populations of 10^0 to 10^{15} members. Molecular diversity for directed evolution can be produced, *in vitro* or *in vivo*, using methods including error-prone DNA polymerization, recombination, gene shuffling, combinatorial libraries made using synthetic oligonucleotides, host organism mutator strains, and the humoral immune system.[4, 5] Active variants can then be isolated from

these pools with techniques such as screening with various levels of automation, selection for survival in auxotrophic strains, and affinity-based or cytometric separation of individual proteins linked to or compartmentalized with the nucleic acids that encode them.[4, 5] Notable protein engineering feats facilitated by directed evolution include enhancement of the substrate specificity, thermostability, selectivity, and solvent tolerance of enzymes,[6–9] as well as the engineering of metabolic pathsteelyways.[10]

Despite the potential of laboratory evolution to cull through vast numbers of sequences, experimental concerns, such a lack of appropriate high-throughput assays, selection systems, or instrumentation, can dramatically limit the diversity that may practically be addressed. Furthermore, the largest libraries that could conceivably be approached by experimental methods are miniscule compared to the total possible diversity of even modestly sized proteins. The inherent limitations of experimental protein engineering methods, and the hope that sequence design might eventually be completely automated, have motivated the development of computational tools for the virtual screening of astronomical pools of diversity. In some cases, the use of computational protein design (CPD) methods has allowed stable, well-folded, and functional proteins variants with many mutations away from their wild-type counterparts to be designed directly *in silico* and experimentally validated. In most situations, however, the successful design of proteins using CPD alone has been inconsistent.

This thesis describes the conception, implementation, and testing of optimization methods meant to improve the frequency with which useful and interesting sequences can be predicted by computational techniques. The initial goal was to facilitate the discovery of sequences of minimum energy given a standard model of computational protein

design. Such sequences are desirable to the extent that the design model is accurate, and are expected to help suggest possible improvements when it is not. However, the enhanced optimization methods also proved to be directly applicable to more sophisticated design models that treat multiple conformational states simultaneously. In order to better assess the utility of these more realistic methods, a method was developed to allow the design of combinatorial mutation libraries to be driven by the results of computational protein design calculations in a model-independent manner. The synergistic application of the methods described here enabled the first experimental test of computational design based on large structural ensembles.

Computational protein design by inverse folding

CPD was first conceived as the inverse of the protein-folding problem, since its most basic goal is to generate amino acid sequences that preferentially adopt a specific three-dimensional structure.[11] At its core, the inverse-folding design model consists of a search for optimal amino acid side chains at one or more positions in a fixed structural model of the protein main chain. Various amino acid types are modeled at each designed position, and potential mutations are evaluated based on their pairwise interaction energies. The continuous flexibility available to each amino acid side chain is approximated using a discrete set of low-energy conformations called rotamers.[12, 13] The goal is thus to find an optimal choice of rotamer, of any allowed amino acid type, at each designed position. A configuration of the virtual protein system described by the inverse folding model corresponds to a set of atomic coordinates for some particular amino acid sequence. To the extent that the approximations inherent in the scoring functions and

design model are appropriate, amino acid sequences specified by low-energy configurations of the system are expected to stabilize the required fold and thereby facilitate the desired activity.

Enthusiasm for computational protein design by inverse folding was piqued when an algorithm for the generation of provably optimal solutions was reported. This method, based on the dead-end elimination (DEE) theorem, specifies criteria by which rotamers at particular positions can be definitively excluded from the global minimum energy configuration (GMEC) of the system.[14] The availability of a rigorous framework for combinatorial optimization in the inverse folding model promoted confidence that protein design was computationally tractable. Accordingly, many of the initial successes in CPD were achieved via DEE-based optimization methods.[15–19] A comparison of DEE with stochastic optimization routines such as genetic algorithms (GA) and Monte Carlo with simulated annealing (MC) indicated that the inexact routines often failed to find GMEC solutions when applied with equivalent computational effort.[20] During this time, DEE-based methods were improved with a number of additional elimination strategies and heuristics that rendered them amenable to CPD problems with more variable positions and more rotamers per position.[21]

Despite these advances, the poor performance scaling of DEE has caused the field to shift toward stochastic optimization methods such as MC[22–24] and FASTER[25] so that larger and more complex designs, such as those involving enzyme substrates and transition state models, could be attempted.[23, 26–30] Rather than eliminate particular rotamers from consideration until only a single sequence remains, these stochastic methods sample sequence space by choosing perturbations to make to a fully instantiated

configuration of the system, and accepting or rejecting the perturbations based on their energetic consequences. Although these methods do not produce solutions that are provably optimal in a global sense, they can be relied upon to find local minima quickly, and their running times can be extended for as long as desired in an attempt to improve existing solutions.

In Chapter 2, I describe enhancements to the FASTER optimization procedure that dramatically improve its ability to converge to a single lowest-energy solution. In every case tested, this FASTER-based solution was either identical to the solution produced by DEE, or was the lowest-energy solution ever found (by any method) if DEE was not able to converge. In some cases, these FASTER-derived optimal solutions could not be found with extremely long runs of MC, suggesting that the improved FASTER procedure is preferable to currently available alternatives. Our experiences with FASTER strongly indicate that the combinatorial optimization problem for design by inverse folding is essentially solved: FASTER is able to quickly converge to low-energy solutions for any problem that could be meaningfully addressed with pure inverse folding simulations.

Beyond single-state inverse folding: multi-state design

The inverse-folding design model has made practical the *in silico* screening of more than 10^{200} amino acid sequences in a single design. However, it also presents significant challenges to the development of atomic scoring functions that accurately predict the fitness of particular sequences. Although initial explorations of CPD evaluated interactions between rotamers with energy functions such as Leonard-Jones

and bonded-term potentials used in molecular dynamics simulations,[31, 32] researchers soon introduced significant complications to the scoring model in an effort to make it applicable to a broader range of design goals. Some of these supplementary energy terms, such as orientation-dependent hydrogen bonding functions and implicit solvation models, have clear physical justifications and are generally accepted in the wider protein simulation community.[23, 33–36] Others, such as penalties for the exposure of nonpolar groups and unfolded state energies based on amino acid composition,[23, 37] are seldom used outside the realm of CPD. These heuristic negative design terms have been adopted primarily to solve problems peculiar to the comparison of different sequences by their molecular mechanics energies, and to overcome the rigidity of the inverse-folding design model.

Ultimately, the viability of any particular sequence depends on the degree to which it populates an entire ensemble of conformational states, including active/native states, misfolded and unfolded states, and aggregated states. While native states can be understood through high-resolution structures derived from experiment, general and tractable atomic-resolution models of alternate conformational states have not yet been developed. Implicit negative design terms like those mentioned above help the design procedure to assess how potential amino acid substitutions might affect the tendency of a sequence to assume poorly defined nonnative states. Nevertheless, these terms are often insufficient to allow selection of reasonable sequences in the context of large designs. For example, RosettaDesign, a CPD procedure based on a highly parameterized forcefield with many heuristic and implicit negative design terms, cannot effectively produce reasonable amino acid compositions when applied to the surfaces of β-sheets.[23]

For this reason, hydrophobic amino acids had to be excluded from β-sheet surface positions during the design of top7, a sequence successfully designed to assume a novel fold.[23]

The inverse-folding paradigm can be extended to allow issues in negative design to be explored in a more systematic manner and offer an alternative to the heuristic terms discussed above. Most notably, Harbury and coworkers have applied explicit negative design algorithms to directly engineer specificity into coiled-coil systems and recapitulate sequences that bind small-molecule ligands with high affinity.[38, 39] In each case, the explicit modeling of alternate states, such as undesired associated and unbound states in the coiled-coil case, and unbound and unfolded states in the ligand-binding case, was crucial for the computational design of variants exhibiting the desired functional properties. Computational multi-state design (MSD) procedures can also be used in a purely positive-design sense to find sequences able to assume several distinct conformations. For example, Ambroggio and Kuhlman used MSD to design a protein switch that assumes completely different folds and association states in the presence versus the absence of zinc.[40]

Given that specificity is crucial to the proper folding, stability, solubility, and activity of proteins, it would seem natural for explicit multi-state modeling to be applied frequently to structure-based computational protein design. Surprisingly, experience with MSD in the CPD community consists essentially of only those investigations just mentioned. Several complications have thus far limited the general utility of MSD in computational protein design.

The first major impediment to the application of MSD in CPD relates to how the various relevant states are specified. In cases with well-characterized target and competing states, one can use high-resolution experimental structures to model each of the desired states. For example, competition between homodimer and heterodimer coiled coils has been modeled by threading the relevant sequences onto identical main chain models from a crystal structure.[39] Similarly, the unbound state in a ligand-binding system was modeled by removing the ligand from a crystal structure of the bound complex.[38] Two different crystal structures were used to model the two target states in the molecular switch design.[40] In general, the astronomical range of conformations available to a sequence must be approximated with a much smaller, computationally tractable set devised to represent the entire ensemble. Unfortunately, methods for the construction of general and accurate models for some important alternative states are not yet available. Although aggregated and unfolded states have been treated in MSD using native conformations in low-dielectric media and random chain ensembles, respectively,[38, 39] the degree to which these simplified models can realistically capture the relevant properties of these states is unclear. Because unfolded and aggregated ensembles are likely quite varied and diverse, explicit treatment of them in MSD will require accurate and efficient methods to generate structural models that adequately represent them, as well as powerful MSD optimization methods that can efficiently sample sequence space given a large number of states.

The second major problem in the general application of MSD relates to how sequences should be evaluated given an ensemble of structural states. Because a single amino acid sequence can assume completely different conformations in each relevant

state, MSD methods require a two-level optimization procedure in which an outer routine samples amino acid sequences and an inner routine evaluates the energy of a sequence in the context of each state separately by rotamer optimization. The energies that result from these individual rotamer optimization calculations must be combined to yield a single fitness score that can be used to evaluate the sequence. No consensus has yet been reached on what energy combination function should be used for this purpose; in fact, different functions may be appropriate for different design goals. One attractive approach is to assess fitness according to the probability, P, that one of the desired target conformations would be fulfilled. P can be computed using basic statistical mechanics, given a finite set of desired target states, undesired competing states, and their energies:

$$P = \sum_{i \in S_T} e^{-E_i / RT} \Big/ \sum_{i \in (S_T \cup S_C)} e^{-E_i / RT} \qquad (1)$$

where S_T is the set of target states, S_C is the set of competing states, and E_i is the energy of state i.[38, 39]

This strategy cannot be used when competing states are not explicitly modeled, because the probability computed with equation 1 would always be unity in this case. When competing states are not considered, one possibility would be to simply average or sum the energies of a sequence on each state.[40] This is appropriate when the design goals require that all specified states be satisfied, as in the design of a protein switch; however, biases can arise if the magnitudes of the energies in different states are significantly different. One could also evaluate the fitness of a sequence by computing the free energy, A, of the system based on all modeled states and their energies:

$$A = -kT \log(\sum_i e^{-E_i / kT}) \qquad (2)$$

This strategy is applicable when the target state ensemble consists of similar structures intended to approximate realistic conformational flexibility, and the incompatibility of a sequence with a small fraction of the available states is relatively inconsequential. MSD scoring schemes like those based on equations 1 and 2 are expected to provide better accuracy as the number of modeled states increases. Because each individual state relies on energy calculations in the context of a rigid main chain, atomic clashes in a few states can unrealistically effect sequence selection when the total number of modeled states is small.

The final major issue in the wider adoption of multi-state design is simply that it presents a more taxing optimization problem than standard single-state design (SSD). The greater difficulty arises because a single amino acid sequence might assume completely different conformations in each relevant MSD state. This prohibits the amino-acid-ignorant rotamer optimization strategies that accelerate convergence in single state design, and requires the two-level optimization procedure described above. Because MSD must perform what essentially amounts to multiple small, independent design calculations in order to assess the fitness of a single amino acid sequence, the diversity of sequences that may be effectively sampled in MSD is dramatically limited relative to SSD. Furthermore, whereas SSD sampling in the inverse folding model is made significantly more efficient by precomputing all possible pairwise energies between rotamers at different positions and using this energy matrix as a lookup table during rotamer optimization, current limitations on physical memory render simple adaptations of this strategy untenable for MSD problems with more than a few states. Unfortunately, as discussed above, issues with the specification of representative conformational states

and the aggregation of state energies into a single fitness score should be better ameliorated when the total number of modeled states increases. Thus, both technical and scientific concerns necessitate more sophisticated and powerful optimization methodologies for acceptable sampling performance in MSD to be achieved. Although DEE-based methods have begun to be adapted to MSD problems,[41] our experiences with SSD suggest that such methods will not provide a "silver bullet" for MSD.

In Chapter 3, I present an optimization framework for multi-state CPD that can easily handle hundreds of states, and whose running time scales linearly with the number of states that are treated. Furthermore, I describe the development of an MSD-capable version of the FASTER optimization algorithm within this framework. The test calculations I report indicate that MSD-FASTER offers significant performance enhancements compared to an MSD-enabled implementation of Monte Carlo with simulated annealing (MSD-MC), that MSD-FASTER finds low-energy sequences more quickly, and that, in some cases, the lowest-energy sequences found by MSD-FASTER cannot necessarily be found at all by MSD-MC during a sampling run of reasonable length. The simulation tools developed in Chapter 3 provide a robust framework on which to base future investigations of ensemble design, explicit negative design, and new atomic-resolution models of unfolded, misfolded, and aggregated states in CPD.

MSD might be used to help overcome the inaccuracies inherent to the application of inverse folding to a single, fixed, main-chain structure. By designing sequences to satisfy an ensemble of related main-chain conformations, a MSD procedure can account, at least partially, for both the tendency of real proteins to relax in order to accommodate mutations, and the contribution of conformational entropy to protein stability. The most

obvious sources of input structural data for this purpose are nuclear magnetic resonance (NMR) experiments, for which results are widely available in the protein data bank (PDB), and molecular dynamics (MD) simulations starting from crystallographic conformations, which can be performed using a variety of accessible commercial and open-source software packages.

In Chapter 4, I describe the computational design and experimental stability assessment of several combinatorial libraries based on different sources of input structural information for the same protein. The input models include a crystal structure, an NMR ensemble, a constrained, minimized average NMR structure, and constrained and unconstrained MD ensembles. Experimental analysis of these libraries indicates that the use of an MD ensemble may help to mitigate design failures that occur due to energy function inaccuracies and the approximations of conformational discretization, but also that care must be taken in constructing an ensemble to use for this purpose.

Beyond pure computational protein design: library design

Approximations in the molecular mechanics and heuristic energy functions used in CPD, a lack of accurate structural models for all the relevant conformational states, incomplete sequence and conformational sampling, and failures to model dynamics and chemical transformations all contribute to render extremely challenging the direct *in silico* design of functional proteins. Towards this goal, progress in algorithms, physical chemical models, and computing hardware must be coupled with the frequent and rigorous comparison of computational predictions with experimental reality. Furthermore, continuing development of CPD will not be sustained without evidence that

current methods can facilitate or expedite real-world protein engineering efforts. For these reasons, recent investigations in the field have begun to focus on the synergistic coupling of CPD calculations with experimental screening and selection methods developed for use in directed evolution.

The results of protein design simulations have been used to help determine particular residues that might be especially amenable to site-saturation mutagenesis or site-directed recombination,[42–44] and have facilitated the creation of combinatorial mutation libraries.[45–47] Given appropriate laboratory automation hardware, lists of CPD-derived sequences can also be individually encoded, expressed, and assayed in high-throughput fashion.[48] Laboratory evolution procedures have also been applied to improve the lower levels of activity found in *de novo* computationally designed enzymes.[28]

For the purposes of validating and improving CPD, library design methods that maintain a closer relationship between the sequences actually tested and the sequences produced by the calculations are preferred. Thus, it might seem that simply constructing the top *n* sequences produced by a design calculation would be ideal in this case. However, practical considerations often prohibit this strategy. Few academic researchers have the resources necessary to construct and test more than tens of individual sequences for a given design problem. Furthermore, the availability of an efficient high-throughput screen or selection vastly increases the diversity that can be assayed far beyond what would be possible through gene assembly of individual sequences at any cost. In these cases, a designed combinatorial gene library can provide a more appropriate match, because libraries with arbitrary numbers of members can be synthesized economically and easily, even without laboratory automation.

Although several reported methods allow the results of CPD calculations to drive the design of combinatorial mutation libraries,[45, 47, 49] each suffers from several drawbacks that limit its generality or reduce the clarity with which the libraries it produces reflect on the predictions of the original design calculations. In Chapter 4, I describe the development and implementation of a new algorithm for the computational design of combinatorial mutation libraries based on arbitrary lists of scored amino acid sequences, such as those generated by CPD. In contrast to any competing method suggested so far, this method fulfills all of the following desired qualities: (1) it considers CPD energies explicitly; (2) it allows the user to directly specify the range of viable library sizes; (3) it allows complete control over which sets of amino acids can be considered; (4) it does not rely on heuristics to reduce the computational complexity of the problem by eliminating potentially viable libraries. This combinatorial library design algorithm was used to generate the sets of sequences that we tested for each of the designs based on different sources of structural information as described in Chapter 4.

Our results indicate that this method allows CPD to extend directly to the design of combinatorial libraries that exhibit a high proportion of stable, well-folded members. In addition to validating the new library design method, our results provide a stronger basis on which to recommend library design than was allowed by previous reports, which focused on larger libraries and displayed less obvious connections between the contributions of the computational design and the experimental results.

Conclusions

The work described here illustrates how the development of enhanced sampling and optimization procedures can crucially aid the progress of method refinement and improvement in CPD. The discovery of more efficient optimization procedures, originally intended for single-state design, prompted their application to multi-state design methods that allow many conformational states to be modeled simultaneously. The availability of these MSD methods and a general procedure for the automated design of combinatorial mutation libraries together allowed an investigation of the dependence of design results on the type and quality of input structural data. The results of these experiments provide important clues about how CPD methodology improvements should proceed. As CPD simulations become more realistic, we expect the development of more efficient sampling methods to become more central to the success of CPD, and energy function development to become less so. As more aspects of protein structure and stability begin to be modeled explicitly, the implicit and heuristic negative design terms intended to account for them can be discarded. With additional advances in computational power, conformational sampling methods, multi-state design sequence optimization algorithms, and general representations of alternate states, the set of theoretically defensible energy functions used in other types of protein simulation may one day be sufficient for the accurate computational design of protein sequences.

References

1. Keefe, A. D.; Szostak, J. W., Functional proteins from a random-sequence library. *Nature* **2001,** *410* (6829), 715–718.

2. Taylor, S. V.; Walter, K. U.; Kast, P.; Hilvert, D., Searching sequence space for protein catalysts. *Proceedings of the National Academy of Sciences of the United States of America* **2001,** *98* (19), 10596–10601.

3. Drexler, K. E., Molecular Engineering — an Approach to the Development of General Capabilities for Molecular Manipulation. *Proceedings of the National Academy of Sciences of the United States of America* **1981,** *78* (9), 5275–5278.

4. Bershtein, S.; Tawfik, D. S., Advances in laboratory evolution of enzymes. *Current Opinion in Chemical Biology* **2008,** *12* (2), 151–158.

5. Jackel, C.; Kast, P.; Hilvert, D., Protein design by directed evolution. *Annual Review of Biophysics* **2008,** *37*, 153–173.

6. Arnold, F. H., Combinatorial and computational challenges for biocatalyst design. *Nature* **2001,** *409* (6817), 253–257.

7. Hult, K.; Berglund, P., Engineered enzymes for improved organic synthesis. *Current Opinion in Biotechnology* **2003,** *14* (4), 395–400.

8. Krishna, S. H., Developments and trends in enzyme catalysis in nonconventional media. *Biotechnology Advances* **2002,** *20* (3–4), 239–267.

9. Turner, N. J., Directed evolution of enzymes for applied biocatalysis. *Trends in Biotechnology* **2003,** *21* (11), 474–478.

10. Chatterjee, R.; Yuan, L., Directed evolution of metabolic pathways. *Trends in Biotechnology* **2006,** *24* (1), 28–38.

11. Pabo, C., Molecular Technology — Designing Proteins and Peptides. *Nature* **1983,** *301* (5897), 200.

12. Dunbrack, R. L.; Cohen, F. E., Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science* **1997,** *6* (8), 1661–1681.

13. Ponder, J. W.; Richards, F. M., Tertiary Templates for Proteins — Use of Packing Criteria in the Enumeration of Allowed Sequences for Different Structural Classes. *Journal of Molecular Biology* **1987,** *193* (4), 775–791.

14. Desmet, J.; Demaeyer, M.; Hazes, B.; Lasters, I., The Dead-End Elimination Theorem and Its Use in Protein Side-Chain Positioning. *Nature* **1992,** *356* (6369), 539–542.

15. Bolon, D. N.; Mayo, S. L., Enzyme-like proteins by computational design. *Proceedings of the National Academy of Sciences of the United States of America* **2001,** *98* (25), 14274–14279.

16. Dahiyat, B. I.; Mayo, S. L., De novo protein design: Fully automated sequence selection. *Science* **1997,** *278* (5335), 82–87.

17. Dwyer, M. A.; Looger, L. L.; Hellinga, H. W., Computational design of a Zn2+ receptor that controls bacterial gene expression. *Proceedings of the National Academy of Sciences of the United States of America* **2003,** *100* (20), 11255–11260.

18. Looger, L. L.; Dwyer, M. A.; Smith, J. J.; Hellinga, H. W., Computational design of receptor and sensor proteins with novel functions. *Nature* **2003,** *423* (6936), 185–190.

19. Malakauskas, S. M.; Mayo, S. L., Design, structure and stability of a hyperthermophilic protein variant. *Nature Structural Biology* **1998,** *5* (6), 470–475.

20. Voigt, C. A.; Gordon, D. B.; Mayo, S. L., Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *Journal of Molecular Biology* **2000,** *299* (3), 789–803.

21. Gordon, D. B.; Hom, G. K.; Mayo, S. L.; Pierce, N. A., Exact rotamer optimization for protein design. *Journal of Computational Chemistry* **2003,** *24* (2), 232–243.

22. Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P., Optimization by Simulated Annealing. *Science* **1983,** *220* (4598), 671–680.

23. Kuhlman, B.; Dantas, G.; Ireton, G. C.; Varani, G.; Stoddard, B. L.; Baker, D., Design of a novel globular protein fold with atomic-level accuracy. *Science* **2003,** *302* (5649), 1364–1368.

24. Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E., Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics* **1953,** *21* (6), 1087–1092.

25. Desmet, J.; Spriet, J.; Lasters, I., Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins* **2002,** *48* (1), 31–43.

26. Jiang, L.; Althoff, E. A.; Clemente, F. R.; Doyle, L.; Rothlisberger, D.; Zanghellini, A.; Gallaher, J. L.; Betker, J. L.; Tanaka, F.; Barbas, C. F.; Hilvert, D.; Houk, K. N.; Stoddard, B. L.; Baker, D., De novo computational design of retro-aldol enzymes. *Science* **2008,** *319* (5868), 1387–1391.

27. Lassila, J. K.; Privett, H. K.; Allen, B. D.; Mayo, S. L., Combinatorial methods for small-molecule placement in computational enzyme design. *Proceedings of the National Academy of Sciences of the United States of America* **2006,** *103* (45), 16710–16715.

28. Rothlisberger, D.; Khersonsky, O.; Wollacott, A. M.; Jiang, L.; DeChancie, J.; Betker, J.; Gallaher, J. L.; Althoff, E. A.; Zanghellini, A.; Dym, O.; Albeck, S.; Houk, K. N.; Tawfik, D. S.; Baker, D., Kemp elimination catalysts by computational enzyme design. *Nature* **2008,** *453* (7192), 190–U4.

29. Zanghellini, A.; Jiang, L.; Wollacott, A. M.; Cheng, G.; Meiler, J.; Althoff, E. A.; Rothlisberger, D.; Baker, D., New algorithms and an in silico benchmark for computational enzyme design. *Protein Science* **2006,** *15* (12), 2785–2794.

30. Shah, P. S.; Hom, G. K.; Ross, S. A.; Lassila, J. K.; Crowhurst, K. A.; Mayo, S. L., Full-sequence computational design and solution structure of a thermostable protein variant. *Journal of Molecular Biology* **2007,** *372* (1), 1–6.

31. Desjarlais, J. R.; Handel, T. M., De-Novo Design of the Hydrophobic Cores of Proteins. *Protein Science* **1995,** *4* (10), 2006–2018.

32. Hellinga, H. W.; Richards, F. M., Construction of New Ligand-Binding Sites in Proteins of Known Structure .1. Computer-Aided Modeling of Sites with Predefined Geometry. *Journal of Molecular Biology* **1991,** *222* (3), 763–785.

33. Dahiyat, B. I.; Gordon, D. B.; Mayo, S. L., Automated design of the surface positions of protein helices. *Protein Science* **1997,** *6* (6), 1333–1337.

34. Dahiyat, B. I.; Mayo, S. L., Protein design automation. *Protein Science* **1996,** *5* (5), 895–903.

35. Kortemme, T.; Morozov, A. V.; Baker, D., An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein-protein complexes. *Journal of Molecular Biology* **2003,** *326* (4), 1239–1259.

36. Lazaridis, T.; Karplus, M., Effective energy function for proteins in solution. *Proteins—Structure Function and Genetics* **1999,** *35* (2), 133–152.

37. Dahiyat, B. I.; Mayo, S. L., Probing the role of packing specificity in protein design. *Proceedings of the National Academy of Sciences of the United States of America* **1997,** *94* (19), 10172–10177.

38. Boas, F. E.; Harbury, P. B., Design of protein-ligand binding based on the molecular-mechanics energy model. *Journal of Molecular Biology* **2008,** *380* (2), 415–424.

39. Havranek, J. J.; Harbury, P. B., Automated design of specificity in molecular recognition. *Nature Structural Biology* **2003,** *10* (1), 45–52.

40. Ambroggio, X. I.; Kuhlman, B., Computational design of a single amino acid sequence that can switch between two distinct protein folds. *Journal of the American Chemical Society* **2006,** *128* (4), 1154–1161.

41. Yanover, C.; Fromer, M.; Shifman, J. M., Dead-end elimination for multistate protein design. *Journal of Computational Chemistry* **2007,** *28* (13), 2122–2129.

42. Endelman, J. B.; Silberg, J. J.; Wang, Z. G.; Arnold, F. H., Site-directed protein recombination as a shortest-path problem. *Protein Engineering Design & Selection* **2004,** *17* (7), 589–594.

43. Voigt, C. A.; Martinez, C.; Wang, Z. G.; Mayo, S. L.; Arnold, F. H., Protein building blocks preserved by recombination. *Nature Structural Biology* **2002,** *9* (7), 553–558.

44. Voigt, C. A.; Mayo, S. L.; Arnold, F. H.; Wang, Z. G., Computational method to reduce the search space for directed protein evolution. *Proceedings of the National Academy of Sciences of the United States of America* **2001,** *98* (7), 3778–3783.

45. Hayes, R. J.; Bentzien, J.; Ary, M. L.; Hwang, M. Y.; Jacinto, J. M.; Vielmetter, J.; Kundu, A.; Dahiyat, B. I., Combining computational and experimental screening for rapid optimization of protein properties. *Proceedings of the National Academy of Sciences of the United States of America* **2002,** *99* (25), 15926–15931.

46. Mena, M. A.; Treynor, T. P.; Mayo, S. L.; Daugherty, P. S., Blue fluorescent proteins with enhanced brightness and photostability from a structurally targeted library. *Nature Biotechnology* **2006,** *24* (12), 1569–1571.

47. Treynor, T. P.; Vizcarra, C. L.; Nedelcu, D.; Mayo, S. L., Computationally designed libraries of fluorescent proteins evaluated by preservation and diversity of function. *Proceedings of the National Academy of Sciences of the United States of America* **2007,** *104* (1), 48–53.

48. Cox, J. C.; Lape, J.; Sayed, M. A.; Hellinga, H. W., Protein fabrication automation. *Protein Science* **2007,** *16* (3), 379–390.

49. Mena, M. A.; Daugherty, P. S., Automated design of degenerate codon libraries. *Protein Engineering Design & Selection* **2005,** *18* (12), 559–561.

Chapter 2

Dramatic Performance Enhancements for the FASTER Optimization Algorithm

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

Allen, B. D.; Mayo, S. L., Dramatic performance enhancements for the FASTER optimization algorithm. *Journal of Computational Chemistry* **2006,** *27* (10), 1071–1075.

Abstract

FASTER is a combinatorial optimization algorithm useful for finding low-energy side-chain configurations in side-chain placement and protein design calculations. We present two simple enhancements to FASTER that together improve the computational efficiency of these calculations by as much as two orders of magnitude with no loss of accuracy. Our results highlight the importance of choosing appropriate initial configurations, and show that efficiency can be improved by stringently limiting the number of positions that are allowed to relax in response to a perturbation. The changes we describe improve the quality of solutions found for large-scale designs and allow them to be found in hours rather than days. The improved FASTER algorithm finds low-energy solutions more efficiently than common optimization schemes based on the dead-end elimination theorem and Monte Carlo. These advances have prompted investigations into new methods for force field parameterization and multiple state design.

Introduction

Computer programs for protein design and structure prediction typically include a module used to optimize side-chain coordinates in the context of fixed backbone coordinates. To perform this type of calculation, side-chain conformations (rotamers) of one or more amino acid types are oriented onto each residue position, and all possible pairwise rotamer-backbone and rotamer-rotamer interaction energies are calculated using a molecular mechanics force field. This system of interactions is then optimized to find a rotamer configuration of low molecular mechanics energy. The difficulty of finding the lowest-energy configuration increases dramatically with the number of positions designed and the number of rotamers allowed at each position.[1] Useful optimization strategies include Monte Carlo with simulated annealing (MC),[1–4] methods based on dead-end elimination (DEE),[5, 6] methods based on self-consistent mean field theory,[1, 7] genetic algorithms,[1, 8, 9] and the FASTER method.[10] The DEE-based methods have proven especially useful because they ensure that the global minimum energy configuration (GMEC) is identified when they converge.[5] This feature allows researchers to conclude with certainty that any deviations between simulation and experiment are due to problems with the energy functions or simulation model, and are not the result of incomplete optimization. However, current DEE-based algorithms often fail to converge to a single solution when challenged with difficult optimization problems.[6] For this reason, we have begun to favor the FASTER algorithm described by Desmet, Spriet, and Lasters[10] for difficult designs.

Like Monte Carlo, FASTER is a stochastic optimization algorithm that makes perturbations to intermediate solutions and keeps the improvements that it finds.

However, FASTER discovers low-energy solutions far more efficiently, and frequently finds the GMEC as determined by DEE-based algorithms. In cases for which DEE does not converge, it cannot be determined whether or not the solution produced by FASTER is optimal. We typically treat these cases by running many FASTER trajectories in parallel with different random number seeds until the lowest-energy solution has been found multiple times. At this point the solution is considered satisfactory; we refer to such a solution as a FASTER-determined minimum energy configuration (FMEC). This procedure can be time-consuming for problems with many positions and many rotamers at each position. In this paper we present two simple modifications to the published FASTER algorithm that improve the efficiency with which it finds FMEC solutions by as much as two orders of magnitude. In our laboratory, this improvement has reduced the turnaround time for very large designs from days to hours, and has allowed us to begin developing new methods for force field parameterization and multiple state design.

Improvements to FASTER

Original FASTER

As originally described,[10] a FASTER optimization trajectory is computed by executing the following five steps in order: backbone-derived minimum energy configuration (BMEC), iterative batch relaxation (iBR), conditional iBR (ciBR), single perturbation and relaxation (sPR), and double perturbation and relaxation (dPR). The output rotamer configuration of each step is used as input for the next, as follows. BMEC: Generate a starting rotamer configuration by choosing the rotamer at each position with the most favorable interactions with the backbone; rotamer-rotamer

interactions are ignored. iBR: At each position, find the best rotamer in the context of the input configuration at all other positions. Simultaneously update the rotamers at every position after all positions have been considered. Repeat until convergence or cyclic behavior is detected. ciBR: Proceed as in iBR, but randomly accept the new rotamer found at each position with 0.8 probability. sPR: One position at a time, perturb the structure by fixing a rotamer at that position, and allow all other positions to relax as in one round of iBR. The resulting configuration is accepted only if it has the lowest energy found so far. Pick positions for perturbation in random order. Repeat until convergence. dPR: Proceed as in sPR, but perturb pairs of rotamers at different positions together.

Improvement to starting configurations

Regarding the choice of initial rotamer configuration to use as input to FASTER, Desmet et al. noted that the positions of many side-chains can be accurately placed on the protein backbone without considering interactions with other side-chains.[10] Although they showed that this BMEC can serve as an adequate input to FASTER for side-chain placement calculations, our results indicate that the BMEC is suboptimal when FASTER is applied to more difficult protein design problems. Because rotamer-rotamer interactions are ignored, the BMEC is usually a poor solution in terms of amino acid sequence and energy compared to the optimized solutions found by FASTER and other algorithms. Furthermore, the optimization scheme we employ involves computing many separate FASTER trajectories with different random number seeds; because neither the BMEC nor iBR are stochastic, all trajectories are identical until the ciBR step. We hypothesized that FASTER would be able to find the FMEC more effectively if a pool of

partially optimized solutions were generated and initial configurations drawn from that pool. Therefore, we replace the BMEC step at the beginning of each trajectory with a short Monte Carlo run starting from a random configuration. This procedure gives diverse starting solutions with energies significantly better than the BMEC at negligible computational cost.

Improvement to sPR via selective relaxation

As described above, a step of sPR or dPR involves perturbation of the rotamer configuration at one or two positions, followed by relaxation of all the remaining positions in response to the perturbation. In general, however, only a subset of the other positions actually interact significantly with a perturbed position. Thus, the time spent selecting a new rotamer at each of the potentially numerous uncoupled or weakly coupled positions is essentially wasted. This problem can be addressed by limiting the set of positions that are relaxed after every perturbation to those that interact most strongly with the perturbed position. The interaction between a perturbed position and a potential relaxing position may be assessed according to the absolute value of the pairwise interaction energy between the positions before the perturbation. Before a position is perturbed, all the other positions are sorted into a list based on their interactions with the position to be perturbed. The positions to be relaxed are then chosen either by using a number cutoff (the *n* most strongly interacting positions), or an energy cutoff. The optimal value for an energy cutoff depends on the magnitudes of the energies produced by the force field, whereas a number cutoff does not. Therefore, we report calculations

performed with number cutoffs, so that our results might be more useful to researchers using different energy functions.

Methods

The performance of FASTER was tested on four full sequence designs using each method for generating initial configurations (BMEC and MC), and with the number of relaxing positions limited to various values of *n*. We calculated designs for a 28-residue DNA-binding domain of mouse zinc finger Zif268 (PDB code 1AAY, residues 133–160),[11] the 34-residue WW domain from human rotamase Pin1 (1PIN, residues 6–39),[12] the 56-residue B1 domain of streptococcal protein G (1PGA),[13] and the 66-residue cold-shock protein *Bc*-Csp from *Bacillus caldolyticus* (1C9O, chain A).[14] These small, stable, monomeric domains have been the targets of several protein design and stability studies.[15–18]

For each of the four designs, all nonprotein atoms and residues outside the ranges given above were removed; hydrogens were added using REDUCE.[19] All positions were designated core, boundary, or surface as described previously.[15] The amino acids Ala, Val, Leu, Ile, Met, Phe, Tyr, and Trp were allowed at core positions; Ala, Ser, Thr, Asp, Asn, His, Glu, Gln, Lys, and Arg were allowed at surface positions; amino acids from the combination of both sets were allowed at boundary positions. All positions were designed except those with proline or glycine in the wild-type sequence. We used the Dunbrack backbone-dependent rotamer library[20] with expansions of +/- one standard deviation around $\chi 1$ and $\chi 2$ for aromatic amino acids and around $\chi 1$ for hydrophobic amino acids. The average number of rotamers per position over all four designs was 212. Pairwise

energies were computed using energy functions as previously described,[6, 21] except the polar hydrogen burial term was omitted. The design choices reported here reflect the procedures typically used in our laboratory for full-sequence designs.

Optimizations with FASTER were performed as follows. First, rotamers with rotamer-backbone interaction energies greater than 20 kcal/mol or pairs with pairwise interaction energies greater than 50 kcal/mol were eliminated from consideration.[6, 22] Then, simple Goldstein DEE singles elimination was applied until no further rotamers could be eliminated.[6, 23] The input configuration for each trajectory was either the BMEC or the result of a short MC run. The MC was performed by starting with a random configuration and optimizing for 1 cycle of $1x10^6$ steps using a linear temperature gradient from 4500 K to 150 K, followed by quenching[1] of the best-energy sequence that was found. iBR was applied to the input configuration until convergence, followed by 20 cycles of ciBR. Finally, sPR was run with a user-defined value of *n* until convergence. dPR was deemed too computationally expensive to use on all trajectories, and was only applied to the 10 best solutions from each calculation in order to assess whether the FMEC was optimal.

For comparison with FASTER, we also optimized the designs using Monte Carlo. The Monte Carlo optimization was performed according to the procedure described above for FASTER, except that the iBR, ciBR, and sPR passes were skipped, the number of Monte Carlo steps was increased to $2x10^7$, and the low temperature decreased to 0 K. For each design, we computed the same number of trajectories using this Monte Carlo procedure as we had when using FASTER. We also attempted to optimize the designs

using our DEE-based hybrid exact rotamer optimization algorithm (HERO), according to the published procedure.[6]

Results and discussion

The four designs described above were each optimized using 10 different combinations of parameters. We tested values of *n* (the number of positions to relax) from the set (5, 10, 15, 20, N), where N is the total number of positions in the protein. For each *n* tested, we tried FASTER starting from the BMEC solution, and also starting from solutions generated by MC. Starting from the BMEC and setting *n* = N corresponds to FASTER as originally reported by Desmet et al.[10] For each of the four designs, and for each of the 10 parameter combinations tested, we computed 2000 separate FASTER trajectories (8000 for 1AAY). The results of these calculations are presented in Table 1.

Whereas a typical FASTER run might comprise 100 trajectories, here we examined at least 2000 in each case to more accurately assess how easily the FMEC could be found. In particular, we note that when using the original FASTER procedure (BMEC and *n* = N) for 1AAY, as few as 0.01% of the trajectories actually found the FMEC. In this case, the probability of finding the FMEC during a standard run of 100 trajectories approaches zero.

Table 1: **Test calculations illustrating performance enhancements for FASTER**

Design	n [a]	# FMEC[b]		% FMEC[c]		t (minutes)[d]		S[e]		x [f]	
		BMEC	MC	BMEC	MC	BMEC	MC	BMEC	MC	BMEC	MC
1AAY	5	4	29	0.05	0.36	0.24	0.25	485	69	14	98
	10	5	42	0.06	0.53	0.38	0.41	604	79	11	86
	15	5	41	0.06	0.51	0.53	0.59	848	114	8	59
	20	4	23	0.05	0.29	0.69	0.74	1370	257	5	26
	N=28	1	25	0.01	0.31	0.85	0.85	6780	273	1	25
1PIN	5	112	53	5.60	2.65	0.26	0.22	5	8	15	9
	10	113	71	5.65	3.55	0.37	0.36	7	10	11	7
	15	105	77	5.25	3.85	0.50	0.47	10	12	7	6
	20	98	87	4.90	4.35	0.60	0.56	12	13	6	6
	N=34	23	65	1.15	3.25	0.82	0.74	71	23	1	3
1PGA	5	0	9	0.00	0.45	1.9	1.7	—	378	—	16
	10	10	73	0.50	3.65	3.1	2.8	620	77	10	78
	15	10	110	0.50	5.50	4.6	4.0	920	73	7	83
	20	21	110	1.05	5.50	6.2	5.2	590	95	10	63
	N =56	4	116	0.20	5.80	12.0	14.0	6000	241	1	25
1C9O	5	0	12	0.00	0.60	1.3	1.4	—	233	—	99
	10	1	26	0.05	1.30	2.0	1.8	4000	138	6	166
	15	2	35	0.10	1.75	3.0	2.6	3000	149	8	155
	20	1	36	0.05	1.80	3.9	3.2	7800	178	3	129
	N=66	1	54	0.05	2.70	11.5	8.8	23000	326	1	71

[a] The number of positions relaxed after every perturbation during sPR

[b] The number of trajectories that found the FMEC

[c] The percent of trajectories that found the FMEC. The total number of trajectories attempted was 8000 for 1AAY and 2000 for all others.

[d] The time in processor-minutes required to compute a single trajectory, averaged over all trajectories in the run

[e] The score S, representing the number of processor-minutes required, on average, to find the FMEC once. Calculated as $S = t / f$, where f is the fraction of trajectories that found the FMEC. Smaller values are better. "—" indicates that S is undefined because $f = 0$.

[f] The multiplicative factor of improvement compared to the original FASTER protocol

Each combination of parameters may be compared via the score $S = t / f$, where t is the average number of processor-minutes required to compute a single trajectory, and f is the probability that a trajectory would find the FMEC, estimated using the data in Table 1. Thus, S represents the number of processor-minutes it would take, on average, to find the FMEC once; smaller values are better. Using this score as our metric, an improvement in efficiency may occur due to an increase in the fraction of trajectories that find the FMEC, or a decrease in the average convergence time per trajectory, or both.

Table 1 clearly illustrates the utility of starting with an MC solution rather than with the BMEC; when n = N, the improvements in efficiency x observed on switching to MC range from a factor of 3 (1PIN) to a factor of 71 (1C9O). Improvements in this range are also observed for most other values of n we tested; notable exceptions are the 1PIN designs with smaller values of n, for which the BMEC was more effective. In each case, the observed improvements in efficiency when using MC were predominantly due to the greater fraction of trajectories that found the FMEC. For each trajectory, the running time was dominated by the sPR step, and the additional cost of MC was negligible.

With the choice of BMEC/MC held constant, observed changes in f due to the reduction of n from N to (20,15,10) have different magnitudes and signs in the four designs. However, the average time t required to complete a single trajectory was always reduced, typically by a factor of 3–5 when n = N is compared with n = 10. Thus, significant improvements in the computational efficiency S were always observed when reducing n to the range of 10–20. For 1PGA and 1C9O when n = 5, the FMEC was never found when the BMEC was used as an input structure; we therefore avoid the use of n

smaller than 10. Although we have not systematically evaluated parameter combinations for designs larger than 66 positions, we do not anticipate problems using values of n in the range of 10–20 for larger designs.

The overall performance of FASTER is dramatically improved when both enhancements are used together. When using MC instead of the BMEC and with $n = 10$, the computational efficiency S of the 1AAY calculation was improved compared to the original FASTER by a factor of 86. Optimizations for the other designs 1PIN, 1PGA, and 1C9O were improved by factors of 7, 78, and 166, respectively. We note that this improvement in efficiency is not only a convenience. Because users have limited time and computer resources, they will rarely be able to compute as many trajectories for a given design as we describe in this paper. Thus, the improvements allow protein designers to find solutions that are better than those they would have found with the original FASTER protocol, and not merely to find the same solutions more rapidly.

In an attempt to show that the FMEC solutions found by FASTER were optimal, we performed DEE-based optimizations using HERO. HERO converged for the 1PIN design, yielding a sequence and energy identical to the FMEC found by the FASTER trajectories; the other three HERO calculations failed to converge, and so the optimality of the FMEC solutions for the 1AAY, 1PGA, and 1C9O designs is not known. We also tested the optimality of the FMEC solutions by applying dPR until convergence to the top ten solutions found in every FASTER calculation. In no case did this dPR optimization yield a better solution than the FMEC, giving us further confidence that the FMECs used to generate the values in Table 1 are the best solutions that FASTER can provide.

To determine whether the improved FASTER procedure we describe performs better than when Monte Carlo is used alone, we repeated the optimizations with a more extensive MC section and with the FASTER-specific passes omitted, as described above. Table 2 shows that the improved FASTER algorithm is able to find the FMEC solution for each design much more frequently than MC alone, even though the MC trajectories used somewhat more processor time than the FASTER trajectories. Notably, the pure Monte Carlo procedure was never able to find the FMEC for the 1PGA design. For the 1AAY, 1PIN, and 1C9O designs, the improved FASTER algorithm was more efficient than Monte Carlo alone by factors of 10, 7, and 8, respectively. Interestingly, the improvement factors reported in Table 2 also indicate that Monte Carlo is actually more powerful for these three designs than the original FASTER algorithm. Nevertheless, the improved FASTER procedure we report is clearly preferable for all four designs.

Table 2: Comparison of the improved FASTER to Monte Carlo

Design	Opt[a]	# FMEC[b]		% FMEC[c]		*t* (minutes)[d]		*S*[e]		*x*[f]	
		w/[g]	w/o[g]	w/	w/o	w/	w/o	w/	w/o	w/	w/o
1AAY	Monte	12	10	0.15	0.1	1.13	1.25	753	1000	9	7
	Faster	42	25	0.53	0.3	0.41	0.90	78	288	86	24
1PIN	Monte	53	26	2.65	1.3	1.28	1.43	48	110	1	1
	Faster	71	66	3.55	3.3	0.36	0.80	10	24	7	3
1PGA	Monte	0	0	0.00	0.0	3.63	3.73	—	—	—	—
	Faster	73	80	3.65	4.0	2.80	5.05	77	126	78	48
1C9O	Monte	11	6	0.55	0.3	5.68	5.70	1033	1900	22	12
	Faster	26	17	1.30	0.8	1.80	3.92	138	461	166	50

[a] The optimization strategy that was used. Monte: pure MC trajectories as described in Methods. Faster: FASTER trajectories as described in Methods; the number of interacting residues in sPR was limited to 10, and the BMEC step was replaced with MC. The total number of trajectories attempted for both Monte and Faster was 8000 for 1AAY and 2000 for all other designs.

[b–e] See Table 1.

[f] The multiplicative factor of improvement compared to data for the original FASTER protocol reported in Table 1

[g] Indicates whether or not Goldstein singles elimination was performed before the other optimization steps.

The improved FASTER algorithm and Monte Carlo were also assessed without the pre-elimination of singles by Goldstein DEE. Table 2 shows that the DEE step significantly improved the convergence times of FASTER trajectories, and slightly improved the convergence times for the MC trajectories. Furthermore, the use of DEE typically increased the fraction of trajectories that found the FMEC for both FASTER and MC, improving overall efficiency by a factor of 2–4 for FASTER and by close to 2 in

one case for MC. We conclude that the pre-elimination of singles by Goldstein DEE is a worthwhile enhancement to these optimization strategies.

Conclusions

FASTER is a stochastic optimization algorithm that can efficiently find low-energy solutions to difficult protein design problems. We report two simple enhancements to FASTER that together result in up to two orders of magnitude better computational performance with no loss of accuracy. The first improvement replaces the backbone-derived initial configuration with a short Monte Carlo run. The second improvement limits the number of relaxing positions in the perturbation and relaxation steps to a fixed value. The dramatic performance enhancements provided by these changes make FASTER significantly more powerful than alternative methods, and allow better solutions to be found more quickly for larger, more complex designs. We expect the improved algorithm to facilitate the development of next-generation protein design tools that treat multiple states and explicit backbone flexibility.

References

1. Voigt, C. A.; Gordon, D. B.; Mayo, S. L., Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *Journal of Molecular Biology* **2000,** *299* (3), 789–803.

2. Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P., Optimization by Simulated Annealing. *Science* **1983,** *220* (4598), 671–680.

3. Kuhlman, B.; Dantas, G.; Ireton, G. C.; Varani, G.; Stoddard, B. L.; Baker, D., Design of a novel globular protein fold with atomic-level accuracy. *Science* **2003,** *302* (5649), 1364–1368.

4. Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E., Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics* **1953,** *21* (6), 1087–1092.

5. Desmet, J.; Demaeyer, M.; Hazes, B.; Lasters, I., The Dead-End Elimination Theorem and Its Use in Protein Side-Chain Positioning. *Nature* **1992,** *356* (6369), 539–542.

6. Gordon, D. B.; Hom, G. K.; Mayo, S. L.; Pierce, N. A., Exact rotamer optimization for protein design. *Journal of Computational Chemistry* **2003,** *24* (2), 232–243.

7. Koehl, P.; Delarue, M., Application of a Self-Consistent Mean-Field Theory to Predict Protein Side-Chains Conformation and Estimate Their Conformational Entropy. *Journal of Molecular Biology* **1994,** *239* (2), 249–275.

8. Desjarlais, J. R.; Handel, T. M., De-Novo Design of the Hydrophobic Cores of Proteins. *Protein Science* **1995,** *4* (10), 2006–2018.

9. Jones, D. T., De-Novo Protein Design Using Pairwise Potentials and a Genetic Algorithm. *Protein Science* **1994,** *3* (4), 567–574.

10. Desmet, J.; Spriet, J.; Lasters, I., Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins* **2002,** *48* (1), 31–43.

11. Elrod-Erickson, M.; Rould, M. A.; Nekludova, L.; Pabo, C. O., Zif268 protein-DNA complex refined at 1.6 angstrom: A model system for understanding zinc finger-DNA interactions. *Structure* **1996,** *4* (10), 1171–1180.

12. Ranganathan, R.; Lu, K. P.; Hunter, T.; Noel, J. P., Structural and functional analysis of the mitotic rotamase Pin1 suggests substrate recognition is phosphorylation dependent. *Cell* **1997,** *89* (6), 875–886.

13. Gallagher, T.; Alexander, P.; Bryan, P.; Gilliland, G. L., 2 Crystal-Structures of the B1 Immunoglobulin-Binding Domain of Streptococcal Protein-G and Comparison with NMR. *Biochemistry* **1994,** *33* (15), 4721–4729.

14. Mueller, U.; Perl, D.; Schmid, F. X.; Heinemann, U., Thermal stability and atomic-resolution crystal structure of the Bacillus caldolyticus cold shock protein. *Journal of Molecular Biology* **2000,** *297* (4), 975–988.

15. Dahiyat, B. I.; Mayo, S. L., De novo protein design: Fully automated sequence selection. *Science* **1997,** *278* (5335), 82–87.

16. Kraemer-Pecore, C. M.; Lecomte, J. T. J.; Desjarlais, J. R., A de novo redesign of the WW domain. *Protein Science* **2003,** *12* (10), 2194–2205.

17. Malakauskas, S. M.; Mayo, S. L., Design, structure and stability of a hyperthermophilic protein variant. *Nature Structural Biology* **1998,** *5* (6), 470–475.

18. Perl, D.; Schmid, F. X., Electrostatic stabilization of a thermophilic cold shock protein. *Journal of Molecular Biology* **2001,** *313* (2), 343–357.

19. Word, J. M.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C., Asparagine and glutamine: Using hydrogen atom contacts in the choice of side-chain amide orientation. *Journal of Molecular Biology* **1999,** *285* (4), 1735–1747.

20. Dunbrack, R. L.; Cohen, F. E., Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science* **1997,** *6* (8), 1661–1681.

21. Gordon, D. B.; Marshall, S. A.; Mayo, S. L., Energy functions for protein design. *Current Opinion in Structural Biology* **1999,** *9* (4), 509–513.

22. DeMaeyer, M.; Desmet, J.; Lasters, I., All in one: A highly detailed rotamer library improves both accuracy and speed in the modelling of sidechains by dead-end elimination. *Folding & Design* **1997,** *2* (1), 53–66.

23. Goldstein, R. F., Efficient Rotamer Elimination Applied to Protein Side-Chains and Related Spin-Glasses. *Biophysical Journal* **1994,** *66* (5), 1335–1340.

Chapter 3

An Efficient Algorithm for Multi-State Protein Design Based on FASTER

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

Abstract

Most of the methods that have been developed for computational protein design involve the selection of side-chain conformations in the context of a single, fixed main-chain structure. In contrast, multi-state design (MSD) methods allow sequence selection to be driven by the energetic contributions of multiple structural or chemical states simultaneously. This methodology is expected to be useful when the design target is an ensemble of related states rather than a single structure, or when a protein sequence must assume several distinct conformations to function. MSD can also be used with explicit negative design to suggest sequences with altered structural, binding, or catalytic specificity. We report implementation details of an efficient multi-state design optimization algorithm based on FASTER (MSD-FASTER). We subjected the algorithm to a battery of computational tests and found it to be generally applicable to various multi-state design problems; designs with a large number of states and many designed positions are completely feasible. A direct comparison of MSD-FASTER and multi-state-design Monte Carlo indicated that MSD-FASTER discovers low-energy sequences much more consistently. MSD-FASTER likely performs better because amino acid substitutions are chosen on an energetic basis rather than randomly, and because multiple substitutions are applied together. Through its greater efficiency, MSD-FASTER should allow protein designers to test experimentally better-scoring sequences, and thus accelerate progress in the development of improved scoring functions and models for computational protein design.

Introduction

The field of computational protein design provides software tools that facilitate the identification of amino acid sequences with specific desired properties. Most protein design protocols choose amino acid types and side-chain conformations in the context of a single, fixed, main-chain conformation. Given this simplifying approximation, one can precompute all pairwise interaction energies between possible side-chain conformations at different positions and then optimize this system of interactions to find sequences expected to stabilize the fold.[1, 2] The most common optimization algorithms employed for this purpose are based on Monte Carlo with simulated annealing (MC),[3–5] the dead-end elimination theorem (DEE),[5–7] genetic algorithms,[5, 8] and Fast and Accurate Side-Chain Topology and Energy Refinement (FASTER).[9, 10] These single-state design methods have produced several notable successes, when used on their own or in conjunction with main-chain optimization techniques.[1, 3, 11–14] However, single-state design is not necessarily sufficient when design objectives require the explicit consideration of multiple states at once.[15]

For example, we might desire a sequence that is able to assume two distinct folds under different conditions; the single-state design methodology described above does not provide a mechanism for selecting sequences that are simultaneously compatible with both folds. Similarly, single- state design methods do not provide a way to explicitly alter binding specificity, since only one binding partner may be modeled during sequence selection. Likewise, enzyme design methods might be enhanced through the explicit modeling of the substrate, transition state, and product, rather than only one of these at a time. Finally, we note that NMR-derived solution structures have been neglected as

targets for protein design because typical structure determination methods give an ensemble rather than a single set of coordinates.[16] To the extent that the structural diversity of an NMR ensemble reflects realistic conformational flexibility, it will be interesting to investigate the effects of using such an ensemble as the basis for design.

Each of the design goals given above requires sequence selection to be informed by multiple structural or chemical states simultaneously, in what we call multi-state design (MSD). The optimization strategy we apply to MSD problems comprises an outer routine that suggests possible amino acid sequences, and an inner routine that assesses the fitness of a sequence by performing rotamer optimization on each state and combining the individual state energies to yield an overall score. This basic approach has been used by others to design specificity into a self-associating coiled-coil system, to generate a molecular switch, and to recover sequences that bind their cognate ligands with high affinity.[15, 17, 18] Here, we describe a generalization of these strategies that is applicable to any number of states and compatible with any type of scoring function that might be used to combine the energies of sequences threaded on the target states.

For a design problem with *n* states to consider, we use *n* processors of a computer cluster to calculate one optimization trajectory. Each processor holds in memory the pairwise energy matrix for one state, and is responsible for evaluating the energies of candidate sequences in the context of that state only. In general, a candidate sequence is evaluated by performing rotamer optimization using a side-chain placement algorithm based on MC, DEE, or FASTER. One of the processors (the boss) is additionally responsible for identifying amino acid sequences to be scored, communicating this information to the others, collecting the results, and combining the energies to yield an

overall fitness score. Here, we provide implementation details for MSD optimization algorithms with amino acid selection schemes based on MC and FASTER, and give quantitative comparisons of their performance for a variety of multi-state design problems.

Results and discussion

Scoring functions

To solve the multi-state design problem, we employ an extension of the methodology that has been developed for single-state design. In single-state design, the cost function to be optimized is the energy E of the rotamer configuration R. The energy is computed by summing the rotamer/template energies E_i for each of the N residue positions and the interaction energies E_{ij} between all pairs of rotamers at residue positions i and j. Typically, the rotamer configuration is optimized without regard to the amino acid types of the rotamers available at each position.

$$E(R) = \sum_{i=1}^{N} E_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} E_{ij} \qquad (1)$$

In multi-state design, the score σ to be optimized is a function of the amino acid sequence A. In general, an amino acid sequence will not assume the same side-chain conformations in the various states being modeled. If there are n states, then the score is computed using a function of the following form:

$$\sigma(A) = \sigma(E_1(A), E_2(A), ..., E_n(A)) \qquad (2)$$

Each $E_s(A)$ corresponds to the energy of the sequence A threaded on state s, and is computed by single-state rotamer optimization using equation 1. Different energy combination functions σ may be appropriate for different types of design problems. For

example, in a case where the designed sequence is meant to satisfy n distinct states equally well, the simplest scoring function simply computes the average energy across all states:

$$\sigma(A) = \frac{1}{n}\sum_{s=1}^{n} E_s(A) \qquad (3)$$

When the design target is an ensemble of similar states, such as an NMR solution structure, the requirement that a sequence satisfy all states may be too stringent; it cannot be assumed that every member of the ensemble would be significantly populated or relevant for the designed sequence. In this case, a scoring function that applies Boltzmann-weighted averaging may be more useful:

$$\sigma(A) = -kT\log\left(\sum_{s=1}^{n} e^{-Es(A)/kT}\right) \qquad (4)$$

Use of equation 4 prevents sequences that fail to satisfy a few states from being severely penalized. If the design goal is to alter conformational, binding, or catalytic specificity, a scoring function for explicit negative design is warranted. Given one positive design state ρ and one negative design state η, one might apply the following scoring function:

$$\sigma(A) = \Delta E_\rho(A) - W\Delta E_\eta(A) \qquad (5)$$

Here, W is a weighting factor used to control the balance of ρ-state stabilization and η-state destabilization. Each $\Delta E_s(A)$ in equation 5 is the excess energy of sequence A when threaded on state s compared to the optimal sequence A_0 for that state as determined by single-state design:

$$\Delta E_s(A) = E_s(A) - E_s(A_0) \qquad (6)$$

Because $E_s(A_0)$ is the minimum energy of any sequence threaded on state s, $\Delta E_s(A) \geq 0$. The $\Delta E_s(A)$ terms are intended to normalize the energies of the sequences being selected

and to allow a single value of W to be used with various energy functions and design targets.

Over the course of a negative design calculation, sequences may be found that cannot be threaded on the negative design target structure without causing severe van der Waals clashes; use of equation 5 in a multi-state design calculation will cause such sequences to be preferred. Any predicted clash must surely be alleviated by a shift in the distribution of conformational states assumed by a real protein. However, we hypothesize that variants with native states perturbed in this manner will tend to be destabilized, especially when multiple clashes are predicted together. Because the energies assigned to these clashes by a standard Leonard-Jones potential depend strongly on several approximations (such as discrete side-chain rotamers and a fixed main chain), we threshold all rotamer-template and rotamer-rotamer energies on the negative design target state to a positive constant. This effectively causes sequences with a greater number of clashes to be preferred over sequences with a smaller number of larger-magnitude clashes, as desired.[19]

A more rigorous approach to explicit negative design would be to maximize the probability with which the target state is assumed over all explicitly modeled competing states, as computed according to basic statistical mechanics. This approach has been applied to the design of specificity in self-associating and ligand-binding systems.[15, 18] The success of this method relies on the availability of atomic models that accurately represent all target and competing states; unfortunately, general methods for the construction of these models have not yet been developed and validated. For the computational tests reported here, we have sidestepped issues of model construction by

applying equation 5 to a system with one crystal structure as the positive design target, and another as the competing state for negative design.

Multi-state Monte Carlo

Monte Carlo with simulated annealing (MC) is an efficient stochastic optimization technique that is heavily used in computational protein design.[3–5] When used for rotamer optimization, MC can produce high-quality approximate solutions quickly and find low-energy variants in the vicinity of an existing solution.[5] MC is easily applied as the outer routine in multi-state design by making perturbations at the level of amino acid sequence only. In each step of multi-state design MC (MSD-MC, Figure 1), a residue position is picked at random, and a random amino acid substitution is made at that position. The new sequence is scored on each state by rotamer optimization. The decision to accept or reject the perturbation is made based on the change in the score σ and the simulated annealing temperature, which is cycled up and down over the course of the optimization to allow traversal of local maxima and exploration around local minima.

We have applied two enhancements to MSD-MC in an attempt to improve its performance. In the first, random perturbations are chosen uniformly from a list of all allowed amino acid substitutions, without respect the positions at which they occur. This prevents positions that have fewer allowed amino acids than others from being the focus of a disproportionate number of substitution attempts. In the second enhancement, rotamer optimization after a substitution is limited to those positions within a specified C_α- C_α distance cutoff from the perturbed position, reducing the amount of time required

for rotamer optimization and allowing more steps of MSD-MC to be completed per unit time.

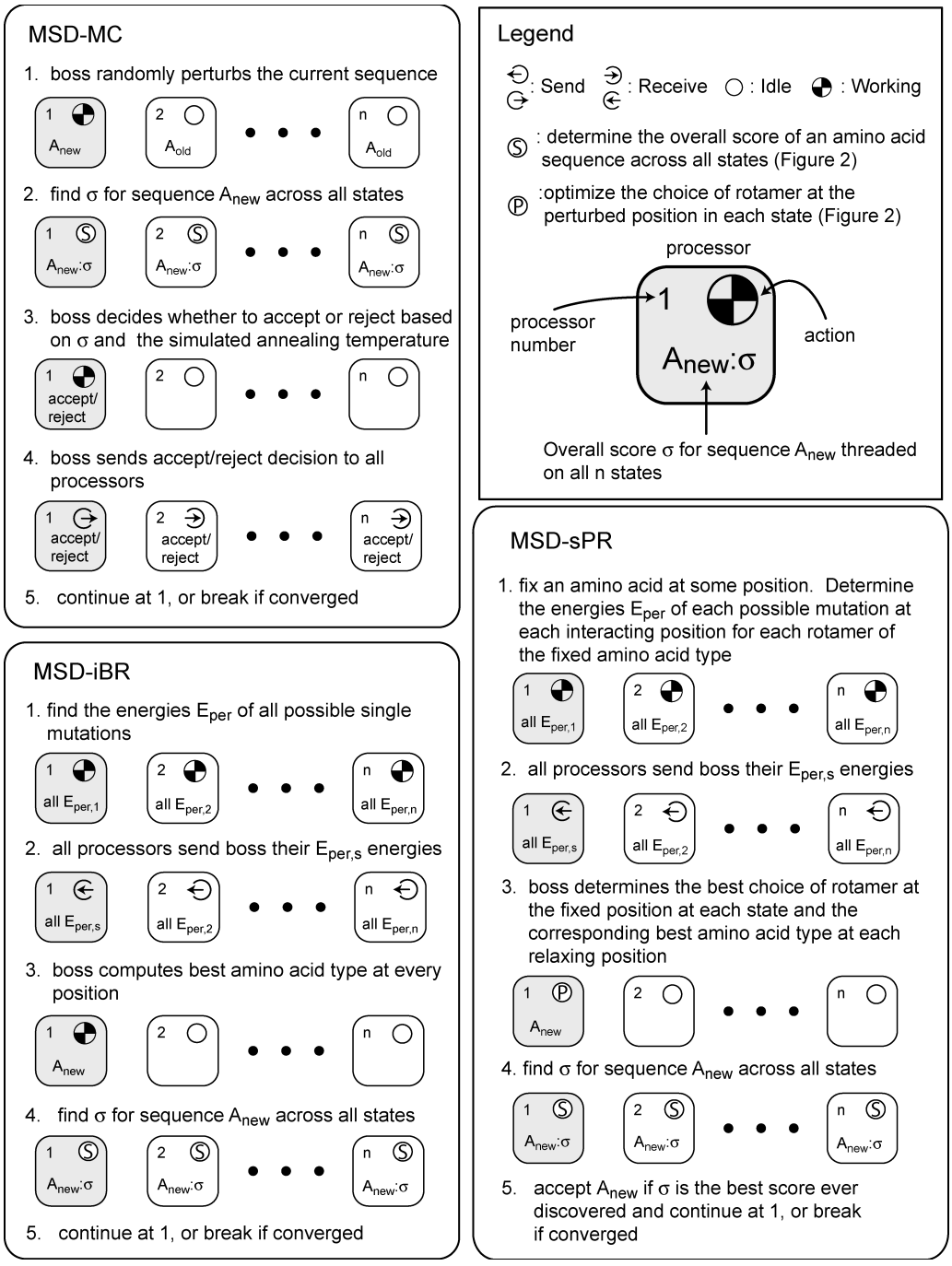

Figure 1: Graphical depictions of the three MSD sequence selection routines described in the text. Legend (upper right panel): explains the symbols used to depict a parallel algorithm. Each box represents a single processor that performs energy calculations on a single state. Fields within the box identify the processor by number, show the current action, and explain the relevant data that the processor holds in memory. The boss processor is shaded in grey. The subroutines **S** and **P** are depicted in Figure 2 and described in the text. Depicted here are: one step of MSD-MC (upper-left panel), one round of MSD-iBR (lower-left panel), and one perturbation in MSD-sPR (lower-right panel).

Multi-state FASTER

Like Monte Carlo, FASTER is a stochastic optimization algorithm that makes perturbations to existing solutions and accepts or rejects them based on their energetic consequences.[10] The two algorithms differ chiefly in the methods by which perturbations are chosen. FASTER has two main components that we have modified for MSD: iterative batch relaxation (iBR), and single perturbation and relaxation (sPR). In each component, amino acid substitutions at several positions are chosen independently and applied together to yield a new solution. Each component is applied iteratively until convergence is detected. In MSD-iBR, convergence is signaled when the user-defined limit for the number of nonproductive rounds (i.e., rounds that fail to improve the energy) is reached. In MSD-sPR, convergence can occur either when the user-defined limit for total rounds is reached or when an entire round has elapsed without an improved solution being found. One trajectory of MSD-FASTER is performed by generating a random initial sequence, applying MSD-iBR until convergence, and then applying MSD-sPR until convergence.

Multi-state iBR

During a round of single-state iBR, the best rotamer at each position of the protein is determined independently in the context of the current rotameric configuration at all other positions. Then, the new rotamers at each position are all updated simultaneously, and the resulting updated configuration of the system is retained regardless of the change in energy. iBR is applied iteratively until a user-defined limit for nonproductive rounds

has been reached. After the detection of convergence, the lowest-energy configuration ever found during the rounds of iBR is selected to move on to sPR.

During a round of MSD-iBR, the best amino acid at each position must be chosen considering all states simultaneously (Figure 1). For each possible amino acid substitution at each position, each processor determines for its own state the best possible total energy of the system when that substitution is made with the current rotamer configuration fixed at all other positions, and sends this information to the boss. If there are p positions and a amino acid types allowed at each position, then each processor needs to communicate pa floating-point values. For each position, the boss computes the overall score of each possible substitution across all states using these values and a scoring function σ. The amino acid identity at each position is then updated with the best-scoring substitution found by the boss in the previous step. Each processor rescores the resulting sequence for its state by rotamer optimization and these energies are again combined to produce an overall score. This process is repeated until convergence, as in single-state iBR.

Multi-state sPR

In a step of single-state sPR, one position is forced to assume a particular rotamer (is "perturbed"), the other positions are allowed to relax independently in the context of the current rotamer configuration, and the rotamers at all relaxing positions are updated at once. The resulting relaxed rotamer configuration is accepted only if its energy is better than any previously observed. In a step of single-state sPR, amino acid substitutions can occur at the perturbed position and also at the relaxing positions, since rotamers are

sampled without regard to their amino acid types. In one round of single-state sPR, each rotamer at each position will be fixed exactly once; positions to fix are picked in random order. Rounds of sPR are performed until an entire round fails to produce a better solution, or until a user-defined limit is reached.

Several significant complications arise when adapting sPR for multi-state design. We would like to fix a particular amino acid at some position and choose the resulting best amino acid substitution at each independently relaxing position (Figure 1). Typically, there will be multiple available rotamers of the fixed amino acid type at the perturbed position in each state. Each of these rotamers will lead to a distinct set of energies for the possible amino acid substitutions at the relaxing positions. Thus, an explicit choice of fixed rotamer at the perturbed position must be made for each state in order to determine the best-scoring amino acid types at the relaxing positions when all states are considered simultaneously. Unfortunately, each processor cannot simply determine the best fixed rotamer in its own state and send the corresponding substitution energies to the boss to be scored. To improve the overall score across all states, a given state may be forced to accept a substitution that is suboptimal when that state is considered by itself. To score that suboptimal substitution correctly, the state may be forced to employ a rotamer at the perturbed position that is different from the one that leads to the best substitutions for that state in isolation. Thus, each processor must communicate substitution energies corresponding to all of the available rotamers of the fixed amino acid type at the perturbed position, and not just of the ones that seem optimal in the context of its own state.

For each rotamer of the fixed amino acid type at the perturbed position, each processor must send the total energy of each possible amino acid substitution at each of the relaxing positions. If there are *r* rotamers of the fixed amino acid type at the perturbed position, *p* relaxing positions, and *a* amino acid types available at each of the relaxing positions, then each processor must send *rpa* floating-point values to the boss.

A given assignment of fixed rotamers to states allows a preferred amino acid substitution at each relaxing position and its MSD score to be computed using a σ function, as described in the MSD-iBR section above. Thus, if there are *n* relaxing positions allowed, there will be *n* separate MSD score values σ_r. In order to determine the best relaxed sequence given an amino acid perturbation, we optimize the sum of these σ_r (subroutine **P** in Figure 2). The optimization comprises a quick Monte Carlo run of 10,000 steps along a linear temperature gradient from 4000 K to 1 K with a nonproductive steps limit of 100. In each step of MC, a random state is chosen, a random fixed rotamer for that state is selected, and the corresponding sum of MSD substitution scores at the relaxing positions is determined; the new fixed rotamer configuration is accepted or rejected based on the Boltzmann criterion. This protocol generates a favorable choice of fixed rotamer for each state and incurs negligible computational expense. After the amino acids at the relaxing positions are chosen, each processor evaluates the energy of the new sequence threaded on its state by rotamer optimization. The energies are then combined into an overall score using a σ function as described above.

Although the technique just described is expected to perform well for most MSD problems, there is some reason to believe that it may be inadequate when used in the

context of explicit negative design. Because subroutine **P** attempts to choose rotamers of the fixed amino acid type that result in designed sequences that minimize σ, it preferentially selects sequences that clash with the chosen fixed rotamers in competing states, even though these clashes might be relaxed away during the subsequent rotamer optimization step. This single-minded focus on sequences that clash most strongly prior to rotamer optimization could inhibit the ability of the algorithm to find those sequences with the most favorable scores after rotamer optimization. To address these concerns, we have implemented and tested two modifications that allow the fixed rotamer configuration (and resulting relaxed amino acid sequence) to be chosen completely randomly, or randomly from one of the top *r* configurations found during subroutine **P**. Comparison with these simple modifications should allow the overall utility of the original procedure to be assessed.

We recently reported that the efficiency of single-state FASTER can be improved by allowing only the positions that interact most strongly with the perturbed position to be relaxed.[9] When applied to MSD-sPR, this improvement also limits the amount of data that must be communicated between processors and improves the efficiency with which the optimal fixed rotamers for each state can be determined. In MSD-sPR, the potential relaxing positions are ranked according to the absolute values of the σ_r scores calculated from their interactions with the perturbed position. The initial rotamer configurations in each state prior to the perturbation are used to assess these interactions.

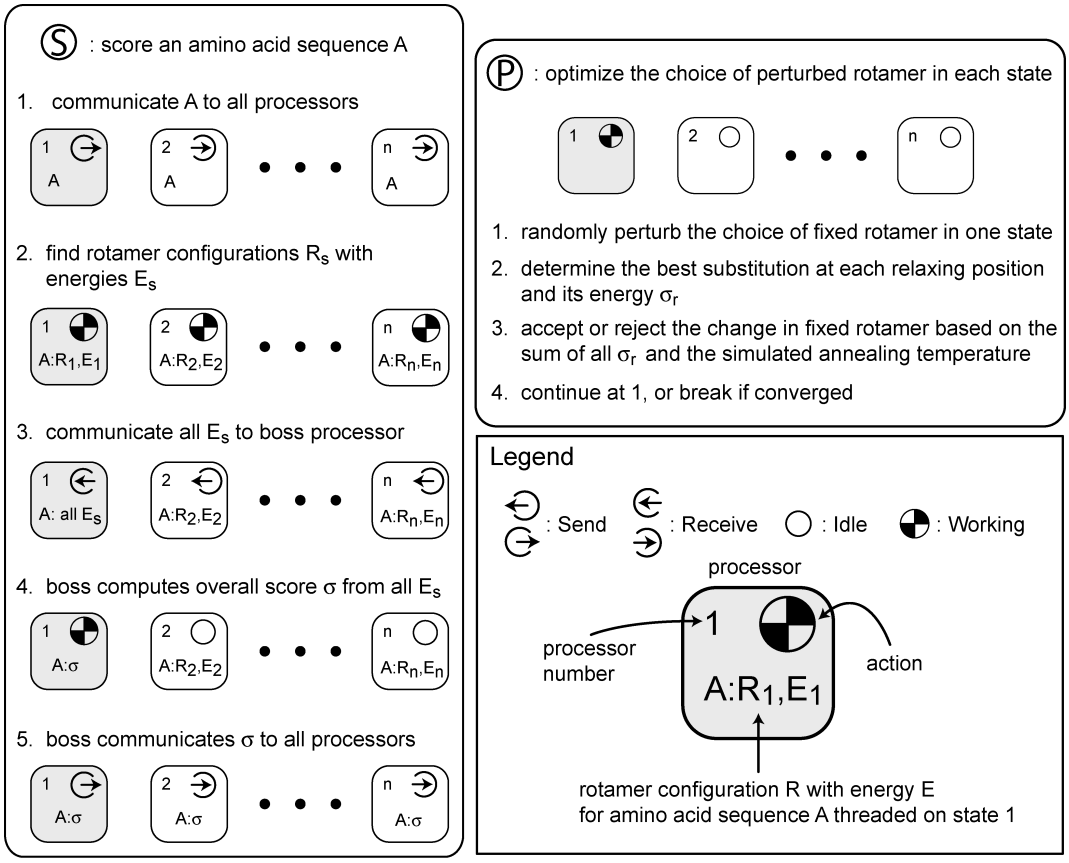

Figure 2: Subroutines used by the MSD sequence selection algorithms. **S**: the subroutine used to assign an overall score to a given amino acid sequence based on input from all of the states. **P:** the subroutine used to determine an optimized choice of fixed rotamer at the perturbed position in each state during MSD-sPR. The boss processor runs this routine using data accumulated from all processors.

Rotamer optimization (RO) algorithms

The MSD sequence selection algorithms described above require that the energy of specific sequences be evaluated in the context of each target state (subroutine **S** in Figure 2). Any of the rotamer optimization (RO) algorithms that have been developed for single-state protein design and side-chain placement, such as MC, DEE, and FASTER, can be used to evaluate these energies. When used for rotamer optimization in this work, one cycle of MC comprised a simulated annealing schedule that varied linearly from high temperature to low. When FASTER was used, rounds of iBR and then of sPR were applied in series; each pass was terminated when convergence was detected or the user-defined rounds limit was reached. In a step of sPR, the set of positions allowed to relax in response to the perturbation was limited to the ten that interact most strongly with the perturbed position.[9] DEE-based rotamer optimizations were performed as previously described,[7] except that the split-DEE and bounding steps were omitted. For some amino acids sequences, DEE failed to converge to a single solution; in these cases, FASTER was automatically invoked to find an approximate solution instead.

When performing rotamer optimization using MC or FASTER, an initial rotamer configuration is required. During multi-state design, RO is applied in subsequent rounds to amino acid sequences that differ at only a few positions; our implementation of MSD exploits this situation to provide better initial rotamer configurations for optimization. In MSD-MC, the amino acid identity at exactly one position will have changed since the most recent rotamer optimization. The rotamer at this position is initialized randomly, while the initial rotamer configuration at each of the unchanged positions is taken directly

from the previous solution. In MSD-sPR, rotamer optimization occurs after each processor has determined the best energies for each amino acid type at several relaxing positions, given a fixed amino acid at a perturbed position. The rotamers at positions that are neither fixed nor relaxed are taken from the previous solution. The rotamer at the fixed position in each state is chosen as described in the section on MSD-sPR above. Reasonable rotamers for each amino acid type at the relaxing positions are also already known; the energies of these rotamers were used to select the sequence being scored. The rotamer solution taken from these three sources can be used to determine directly the energy of the sequence, or additional RO may be performed using it as an initial solution. We refer to the routine that directly determines the energies on each state without further optimization as the Null rotamer optimizer. However, our results below indicate that the Null routine is insufficient for effective MSD sequence optimization.

In each MSD calculation, we employ two different RO modules that we refer to as "weak" and "strong". During rounds of MSD-MC and MSD-sPR, an initial rotamer configuration for each state is available for input to the rotamer optimization routines as described above. Thus, we start from these initial solutions and perform a limited number of rounds of rotamer optimization to save time (weak RO). On the other hand, good initial solutions are not available at the beginning of a round of any MSD algorithm, or at any time during MSD-iBR due to the large number of substitutions that can be made during each round. In these cases, we start from random rotamer configurations and apply more rounds of rotamer optimization to increase our confidence in the resulting energies (strong RO). When DEE is used, it is employed with the same parameters for

both the strong and weak rotamer optimization, because initial solutions cannot be exploited in our implementation of DEE.

Test cases for multi-state design

We tested the performance of the algorithms described here with several different multi-state design problems. The MSD-MC and MSD-FASTER amino acid selection schemes are stochastic and provide no guarantee that the global minimum energy solution will ever be found. We therefore perform many optimization trajectories with different random number seeds, and assess the algorithms based on the distribution of solutions given by these trajectories. When a significant fraction of the trajectories report the same best solution ever found, we take that solution to be optimal. Given the fraction of trajectories f that find the optimal solution, and the average processor-time in minutes t required to compute a trajectory, we compare algorithms using according to the value $S = t / f$. This score represents the total number of processor-minutes required on average to find the optimal solution; smaller values are better. We previously used this metric to analyze the performance of single-state design optimization algorithms.[9]

Single-state design problems

When a MSD algorithm is applied to a design problem with only one target state, its accuracy and efficiency may be compared to well-characterized single-state design algorithms, such as single-state design FASTER (SSD-FASTER). We optimized four full sequence designs that were previously used as test cases for the single-state versions of Monte Carlo and FASTER: 1AAY, 1PIN, 1PGA, and 1C9O. These designs have from

28 to 66 designed positions, and the average number of rotamers per position is 212; a more complete description of these designs is available elsewhere.[9] Because each of these designs had only one target state, the MSD scoring function was simply $\sigma(A) = E(A)$, which is consistent with equations 3 and 4 when $n = 1$.

For each design, we computed 1000 trajectories of MSD-FASTER and MSD-MC with a variety of different weak RO algorithms: Null, MC, iBR, FASTER, and DEE. We refer to a particular pairing of MSD and RO algorithms in a/b format: MSD-FASTER used with FASTER for weak rotamer optimization is called MSD-FASTER/FASTER. For the parameters used in each optimization algorithm formulation, see the materials and methods.

SSD test cases: MSD-FASTER

The results of the MSD-FASTER calculations (Table 1) indicate that the MSD algorithm easily finds the optimal solution (as determined by SSD-FASTER) for each design when paired with weak RO routines based on FASTER, iBR, or MC. For the two smaller designs, 1AAY and 1PIN, MSD-FASTER was actually able to find the lowest-energy solution 20–80% *more* efficiently than SSD-FASTER, because a greater fraction of its trajectories were able to find the optimal solution without requiring significantly more compute time. When applied to the larger and more difficult designs, 1PGA and 1C9O, the performance of MSD-FASTER deteriorated to between 8–18% of the efficiency of SSD-FASTER. This deterioration stemmed both from an increase in the time required to perform simulation trajectories, and a decrease in the fraction of trajectories that were able to find the optimal solution. Ultimately, we were pleased to

discover that, despite the limitations imposed on the algorithm by the requirements of multi-state design, MSD-FASTER can effectively find optimal amino acid sequences among sets of at least 10^{56} alternatives (1C9O). Although MSD-FASTER does not seem to scale to larger problem sizes as well as SSD-FASTER, its performance should allow for the rigorous investigation of new ideas in multi-state computational protein design.

When the results for all four designs are considered simultaneously, the most favorable comparison with SSD-FASTER is offered by MSD-FASTER/FASTER, which allows significant relaxation after each round of MSD-iBR and each step of MSD-sPR. MSD-FASTER also yielded satisfactory performance when MC was used as the weak RO routine, although the number of correct trajectories found per unit time was always fewer than when FASTER was used. As a quicker but less accurate alternative, iBR allowed fewer correct trajectories to be found, but reduced significantly the time required to compute each trajectory, leading to similar overall performance when compared to FASTER and MC. For the 1AAY and 1PIN designs, the most correct trajectories were found when using DEE for rotamer optimization. However, this greater accuracy came at the cost of significantly more processor time required. Furthermore, MSD-FASTER was unable to complete trajectories for the 1PGA design in a reasonable time when RO was performed by DEE (> 100 minutes each), and so the run was aborted. Although DEE-based rotamer optimization may be too slow for sequence selection in nontrivial design problems, it can still be useful to rescore a list of sequences produced using a quicker but more approximate RO method. When no weak RO was performed at all (MSD-FASTER/Null), the optimal solution was found for the 1AAY and 1PIN designs, but not the two larger ones. We note that the average time per trajectory for these designs was

only slightly lower than when iBR was used, indicating that most of the time in MSD-FASTER/iBR is spent choosing sequences to score rather than scoring them by rotamer optimization. Rotamer optimization of some kind seems to be required for the efficient convergence of nontrivial multi-state design problems using MSD-FASTER.

Table 1: **Performance of MSD-FASTER when applied to four difficult single-state design problems**

Design	Size[a]	Opt[b]	% correct[c] ($f \times 100$)	t[d]	S[e]	p[f]
1AAY	28	SSD-FASTER	1.0	0.5	46	1.00
		MSD-FASTER/Null	0.2	0.3	153	0.30
		MSD-FASTER/MC	1.8	0.6	35	1.31
		MSD-FASTER/iBR	1.3	0.4	32	1.44
		MSD-FASTER/FASTER	2.5	0.6	25	1.84
		MSD-FASTER/DEE	3.8	3.6	94	0.49
1PIN	34	SSD-FASTER	2.1	0.6	28	1.00
		MSD-FASTER/Null	1.5	0.4	26	1.08
		MSD-FASTER/MC	3.0	0.7	23	1.22
		MSD-FASTER/iBR	2.5	0.5	20	1.40
		MSD-FASTER/FASTER	3.2	0.7	23	1.22
		MSD-FASTER/DEE	3.6	4.3	118	0.24
1PGA	56	SSD-FASTER	4.2	1.9	46	1.00
		MSD-FASTER/Null	0	3.1	—	—
		MSD-FASTER/MC	1.1	6.2	562	0.08
		MSD-FASTER/iBR	1.5	4.9	327	0.14
		MSD-FASTER/FASTER	3.3	8.5	258	0.18
		MSD-FASTER/DEE	—[g]	—[g]	—	—
1C9O	66	SSD-FASTER	2.0	1.4	71	1.00
		MSD-FASTER/Null	0.0	2.5	—	—
		MSD-FASTER/MC	0.9	5.7	629	0.11
		MSD-FASTER/iBR	0.7	4.3	610	0.12
		MSD-FASTER/FASTER	1.5	7.6	507	0.14
		MSD-FASTER/DEE	1.1	16.4	1486	0.05

a. The number of variable positions in the design
b. The optimization strategy that was used, as described in the text. The term after the slash indicates the weak rotamer optimization routine that was used.
c. The percentage of trajectories that found the best known solution ($f \times 100$), as determined by SSD-FASTER. 1000 total trajectories were computed in each MSD or SSD calculation.
d. The average time, in minutes, required to perform each trajectory on one processor
e. The score $S = t / f$, as described in the text. Smaller values are better, indicating that the optimal solution can be found more quickly. "—" indicates that S is undefined because $f = 0$.
f. The multiplicative factor p measures the deterioration in performance compared to SSD-FASTER. For example, $p = 0.17$ indicates that the MSD algorithm was 17% as efficient as the SSD algorithm.
g. When optimizing the 1PGA design using MSD-FASTER/DEE, the runs were aborted when it was determined that trajectories would take longer than 100 minutes each to complete.

SSD test cases: MSD-MC

To compare the performance of MSD-MC to MSD-FASTER, we repeated the single-state test designs using Null, iBR, MC, and FASTER for rotamer optimization. In the course of these test calculations, it was determined that MSD-MC performed the best when applied with uniform sampling of amino acid substitutions and with the positions to be optimized after a substitution limited to those within 15 Å C_α-C_α of the substituted position, as described above. For brevity, we report only the results of this best MSD-MC formulation here. To make the comparison between MSD-MC and MSD-FASTER as fair as possible, we adjusted the number of Monte Carlo steps in MSD-MC so that the average time per trajectory would be similar to when MSD-FASTER was used (see materials and methods); many more amino acid substitutions can be attempted per unit time if the total time for rotamer optimization per substitution is reduced.

Even using this best formulation, the ability of MSD-MC to find correct solutions to these SSD problems was dramatically worse than that of MSD-FASTER (Table 2). When paired with the Null rotamer optimizer or with iBR, MSD-MC was able to find the optimal solutions to the two smaller design problems, albeit with much lower frequency than MSD-FASTER despite longer sampling times. The relative success of MSD-MC with less rigorous rotamer optimization routines reflects the fact that MSD-MC is strongly limited by the number of amino acid substitutions it is able to test; implementations with less expensive rotamer optimization can afford to test more sequences per unit time, and therefore perform better.

The optimal solutions to the two larger design problems were never found using any implementation of MSD-MC. Because the *S* and *p* scores that were used to compare

the efficiencies of the MSD-FASTER algorithms are undefined when the fraction of correct trajectories is zero, we report two different metrics for MSD-MC. ΔE is the difference in simulation energy between the best sequence found by the MSD-MC algorithm and the optimal sequence found by SSD-FASTER; N_m is the number of positions that differ between the two sequences. Although the 1PGA and 1C9O calculations were not able to find the optimal solution, they can be evaluated based on how close they came (i.e., how close ΔE and N_m are to zero). In terms of ΔE and N_m, these two larger designs showed significant deviations, with differences in simulation energy of 2–4 kcal/mol and 4–7 mutations away from the best-scoring sequence found using SSD-FASTER and MSD-FASTER. Even these suboptimal sequences were found only a few times in the aggregate simulation run, rather than the numerous times the optimal sequence was found by the MSD-FASTER protocols. In addition to various combinations of uniform sampling and restricted sets of positions for rotamer optimization, we attempted various simulated annealing schedules and temperature ranges in MSD-MC, as well as applying fewer trajectories of longer length, all to no avail (data not shown). Compared to MSD-FASTER, the optimization ability of MSD-MC is clearly unacceptable for designs of this difficulty.

Table 2: The performance of MSD-MC when applied to four difficult single-state design problems

Design	Size[a]	Opt[b]	% correct[c] ($f \times 100$)	ΔE[d]	N_m[e]	t[f]
1AAY	28	SSD-FASTER	1.0	0.0	0	0.5
		MSD-MC/Null	0.2	0.0	0	2.5
		MSD-MC/MC	0.2	0.0	0	8.4
		MSD-MC/iBR	0.8	0.0	0	3.2
		MSD-MC/FASTER	0.0	0.7	2	2.6
1PIN	34	SSD-FASTER	2.1	0.0	0	0.6
		MSD-MC/Null	0.3	0.0	0	3.0
		MSD-MC/MC	0.0	0.5	5	9.7
		MSD-MC/iBR	0.1	0.0	0	3.8
		MSD-MC/FASTER	0.0	1.2	9	3.3
1PGA	56	SSD-FASTER	4.2	0.0	0	1.9
		MSD-MC/Null	0.0	3.9	7	5.5
		MSD-MC/MC	0.0	7.8	16	18.1
		MSD-MC/iBR	0.0	1.5	5	16.7
		MSD-MC/FASTER	0.0	11.2	12	9.9
1C9O	66	SSD-FASTER	2.0	0.0	0	1.4
		MSD-MC/Null	0.0	1.6	4	6.7
		MSD-MC/MC	0.0	5.6	14	24.3
		MSD-MC/iBR	0.0	2.0	5	22.7
		MSD-MC/FASTER	0.0	12.4	20	11.0

a. The number of variable positions in the design
b. The optimization strategy that was used, as described in the text. The term after the slash indicates the weak rotamer optimization routine that was used. The number of steps of MSD-MC was adjusted for each algorithm combination so that the average times per trajectory would be similar to those for MSD-FASTER (Table 1).
c. The percentage of trajectories that found the optimal solution ($f \times 100$), as determined by SSD-FASTER. 1000 total trajectories were computed in each MSD or SSD calculation.
d. The difference in simulation energy (kcal/mol) between the best sequence found by MSD-MC and the optimal sequence found by SSD-FASTER
e. The number of residue positions that differ between the best sequence found by MSD-MC and the optimal sequence found by SSD-FASTER
f. The average time, in minutes, required to perform each trajectory on one processor

Multi-state design of protein G

To compare MSD-FASTER and MSD-MC in the context of positive design, we designed two separate areas of 1GB1, a 60-member NMR ensemble of the β1 domain of streptococcal protein G.[20] Single-state designs based on the crystal structure of this protein have found several stabilized variants,[13, 21] but to our knowledge no designs based on an NMR ensemble of this molecule have yet been characterized experimentally. In the first design, we varied all 25 non-Gly positions classified as core or boundary, and in the second we varied all 27 non-Gly positions classified as surface.

For the MSD-FASTER calculations, we dispensed with the evaluation of the several possible rotamer optimization routines, and relied on FASTER only for this purpose. However, given our concerns about potential problems with fixed rotamer selection schemes during MSD-sPR, we tested three implementations in MSD-FASTER. In two cases, (r = 1 and r = 5 in Table 3), the choice of fixed rotamer in each state was determined as described above; the relaxed amino acid sequence to be scored by rotamer optimization was either produced from the best fixed rotamer configuration found, or was produced from a randomly chosen member of the top five configurations found, respectively. In the final case (r = rand), the fixed rotamer optimization was skipped entirely, and the relaxed amino acid sequence to be rescored was determined with fixed rotamers of the perturbed amino acid type chosen randomly for each state. Calculation parameters for MSD-FASTER and the strong and weak rotamer optimization routines were identical to those described for the single-state design test cases above.

We tested a variety of formulations of MSD-MC in an attempt to find one that would compare favorably to MSD-FASTER when applied to many target states

simultaneously. Implementation details that were varied included the type of rotamer optimization performed, the application of uniform sampling of amino acid substitutions, and the use of the distance-based cutoff to limit the expense of rotamer optimization; several of these combinations are shown in Table 3.

In contrast to the SSD test cases described above, the optimal solutions to these two MSD problems are not known except through the calculations we report here. In the absence of additional information, we sampled as rigorously as possible with each MSD algorithm and assumed the best-scoring sequence ever found to be optimal. We typically use this strategy when optimizing single-state designs with stochastic algorithms as well.[9]

For the core+boundary design, all the formulations of MSD-FASTER and MSD-MC we tested found the same lowest-energy solution (Table 3). All three implementations of MSD-FASTER achieved essentially identical performance, indicating that method used to choose fixed rotamers in MSD-sPR was not a significant determinant of optimization power in this design problem. Among the MSD-MC formulations we tested, MSD-MC/iBR performed slightly better than any of the MSD-FASTER implementations, whereas all other performed significantly worse. The preference for a rotamer optimization routine of intermediate expense is consistent with the results of our SSD test calculations (Table 2). It illustrates that, for efficient sampling in MSD-MC to be achieved, a delicate balance must be struck between the accuracy of sequence-rescoring and the number of individual sequences that are evaluated.

Analysis of the surface design calculations shows a stark contrast between the performance of MSD-FASTER and MSD-MC. Whereas all three MSD-FASTER implementations each found the same top sequence in a significant fraction of the

attempted trajectories, this sequence was never found by any of the MSD-MC formulations we tried, despite their greater computational expense. This more difficult design problem also allowed differentiation between the three MSD-FASTER implementations; randomly chosen fixed rotamers (r = rand) resulted in a 5-fold drop in optimization efficiency compared to the use of fixed-rotamer optimization in MSD-sPR (r = 1).

When the states in a MSD calculation are very similar, one might ask whether the MSD-optimal solution could have been found by performing single-state design on each state and rescoring the resulting SSD-derived sequences using MSD. In the case of the core+boundary design described here, the MSD-optimal sequence was never found during single-state design of the individual states; the MSD-optimal sequence for the surface design was also the SSD-optimal sequence for only one of the 60 states. Use of the MSD strategy thus seems warranted for design problems with multi-state requirements; the SSD-based strategy cannot be generally relied upon to produce the same sequences as a true MSD procedure.

The results of the 1GB1 designs show that both MSD-MC and MSD-FASTER can efficiently find low-energy sequences based on a large NMR structural ensemble. Although one formulation of MSD-MC performed slightly better than MSD-FASTER in the core+boundary design, the failure of all MSD-MC formulations when applied to the surface design prompts greater confidence in the consistency and general utility of MSD-FASTER. When applying MSD-FASTER to a large conformational ensemble, the optimization of fixed rotamer choice in MSD-sPR may help to improve the efficiency of sampling in some design problems, and can be recommended on this basis.

Table 3: Multi-state design of 1GB1, a 60-member NMR ensemble of protein G

Design	Size[a]	Opt[b]	r[c]	% correct[e] ($f \times 100$)	t[f]	S[g]
Core	25	MSD-FASTER	1	5.4	3.0	55
+		MSD-FASTER	5	4.8	2.9	60
Boundary		MSD-FASTER	rand	4.1	2.3	56
			US/CPL[d]			
		MSD-MC/FASTER	no	0.7	4.2	593
		MSD-MC/FASTER	yes	2.9	4.3	147
		MSD-MC/Null	yes	0.2	3.4	1712
		MSD-MC/iBR	yes	9.1	4.2	46
			r[c]			
Surface	27	MSD-FASTER	1	5.6	2.8	50
		MSD-FASTER	5	3.8	2.8	75
		MSD-FASTER	rand	1.0	2.6	261
			US/CPL[d]			
		MSD-MC/FASTER	no	0.0	4.2	—
		MSD-MC/FASTER	yes	0.0	4.4	—
		MSD-MC/Null	yes	0.0	3.4	—
		MSD-MC/iBR	yes	0.0	4.3	—

a. The number of variable positions in the design
b. The optimization strategy that was used, as described in the text
c. After optimizing the choice of fixed rotamer in all states during a step of sPR, the amino acid sequence to score by rotamer optimization is chosen randomly from the top r fixed rotamer configurations. "rand" indicates that the fixed rotamer optimization step is skipped, and the amino acid sequence to score results from randomly chosen fixed rotamers in each state.
d. Indicates whether or not uniform substitution sampling is applied in MSD-MC and a close position limit of 15 Å is applied during each rotamer optimization.
e. The percentage of trajectories that found the optimal MSD solution, as defined in the text. 1000 trajectories were computed for each design.
f. The average time, in minutes, required to perform each trajectory using 60 processors
g. The score $S = t / f$, as described in the text. Smaller values are better, indicating that the optimal solution can be found more quickly. "—" indicates that S is undefined because $f = 0$.

Negative design of calmodulin

Calmodulin (CaM) is a second messenger protein that, in the presence of Ca^{2+}, binds to different recognition sequences on various proteins with high affinity and low specificity.[22] CaM variants with increased specificity have been engineered by performing single-state design on a crystal structure of CaM bound to a target peptide from smooth muscle myosin light chain kinase (smMLCK).[23, 24] Experimentally, the variants bound the smMLCK peptide with similar affinity to wild type, and bound most other target peptides with weaker affinity than wild type. Although those experiments showed that single-state design was sufficient to alter binding specificity in this system, we anticipate that more delicate control over such properties may be allowed through the use of explicit negative design. To assess the utility of MSD-FASTER and MSD-MC for negative design, we attempted to design CaM sequences that would bind smMLCK and fail to bind another natural CaM target, CaM kinase I (CaMKI). This sequence selection was performed via a two-state design with a smMLCK-CaM crystal structure as the positive design target state (1CDL),[25] and a CaMKI-CaM crystal structure as the negative design target state (1MXE).[26]

Table 4 compares the application of SSD-FASTER, MSD-FASTER, and MSD-MC to this simple negative formulation of negative design. First, we evaluated the previously published technique for implicit computational negative design. In this case, we applied SSD-FASTER to the positive design target state only, rescored the resulting best sequence against the negative design target state by rotamer optimization, and combined these two energies into an overall score using equation 5. These calculations

indicate a partial clash when the SSD-optimal sequence is threaded on the negative design target state, and a predicted increase in binding specificity.

As with the protein G NMR ensemble calculations, we dispensed with the evaluation of each rotamer optimization routine in the context of MSD-FASTER, and relied on FASTER only. Furthermore, we again tested the fixed rotamer selection schemes during MSD-sPR corresponding to $r = 1$, $r = 5$, and $r =$ rand.

Interestingly, all three techniques found the same best-scoring sequence in 15–20% of their trajectories, and all three incurred roughly the same amount of computational expense. According to the simulations, this sequence is destabilized by only 0.4 kcal/mol in the context of the positive design target state compared to the optimal sequence for that state, and is predicted to clash more significantly when threaded on the negative design target than the sequence found using SSD-FASTER alone. The similarity between the results and performance of the three implementations of MSD-FASTER/FASTER tested here inspires confidence that the utility of MSD-FASTER does not hinge on the particulars of the scheme used to choose rotamers of the fixed amino acid type during MSD-sPR.

We also tested the same set of formulations for MSD-MC as we did for the 1GB1 designs described above, in an attempt to find one that would compare favorably to MSD-FASTER for explicit negative design (Table 4). Despite our best efforts, and even with substantially more computational time devoted to the problem, no version of MSD-MC was able to find the solution produced by MSD-FASTER even once. Furthermore, no MSD-MC calculation converged on any particular consensus solution, indicating that either much longer simulation times or a much better algorithm formulation would be

required for a user to have confidence in the results produced by MSD-MC for this design problem. The best solutions that were found using MSD-MC all exhibited destabilization in the context of the positive design target state in addition to several clashes in the negative design target state; however, only extensive experimental validation will conclusively show whether these differences in simulation energy are meaningful in the context of the potential functions and rigid structural models we have used here. To the extent that predicted clashes correlate with destabilization of the negative design target state, both MSD algorithms are expected to be more useful than single-state design for the explicit manipulation of specificity. Based on our results, MSD-FASTER should be preferred over MSD-MC due to the higher efficiency with which it is able to discover favorable sequences and the greater confidence inspired by its ability to repeatedly discover the optimal solution.

Table 4: Explicit negative design to increase the binding specificity of calmodulin

Opt[a]		% correct[d] ($f \times 100$)	t^e	$\Delta E_P{}^f$	$\Delta E_N{}^g$	σ^h	N^i
SSD-FASTER		0.0	0.9	0.0	37.6	-1.5	2
	r^b						
MSD-FASTER/FASTER	1	18.5	13.9	0.4	54.4	-1.8	0
MSD-FASTER/FASTER	5	19.5	13.4	0.4	54.4	-1.8	0
MSD-FASTER/FASTER	rand	15.1	13.7	0.4	54.4	-1.8	0
	US/CPL[c]						
MSD-MC/FASTER	no	0.0	24.1	4.2	92.0	0.5	6
MSD-MC/FASTER	yes	0.0	27.3	4.0	110.6	-0.4	2
MSD-MC/Null	yes	0.0	14.1	6.0	100.2	2.0	6
MSD-MC/iBR	yes	0.0	15.2	5.7	139.8	0.1	6

a. The optimization strategy that was used, as described in the text. In SSD-FASTER, sequences were optimized in the context of the positive design target only, and then rescored against both targets.
b. After optimizing the choice of fixed rotamer in all states during a step of sPR, the amino acid sequence to score by rotamer optimization is chosen randomly from the top r fixed rotamer configurations. "rand" indicates that the fixed rotamer optimization step is skipped, and the amino acid sequence to score results from randomly chosen rotamers of the fixed amino acid type in each state.
c. Indicates whether or not uniform substitution sampling is applied for MSD-MC and a close position limit of 15 Å is applied during each rotamer optimization.
d. The percentage of trajectories that found the optimal MSD solution, as defined in the text. 1000 trajectories were performed for each MSD calculation, and 6400 were performed for the SSD-FASTER calculation.
e. The average time, in minutes, required to perform each trajectory using 2 processors (MSD), or 1 processor (SSD)
f. The excess energy of the best sequence threaded on the positive design target (equation 6)
g. The excess energy of the best sequence threaded on the negative design target (equation 6). The pairwise energies that are summed to yield this value are each capped at 50 kcal/mol.
h. The overall score of the best sequence found (equation 5)
i. The number of amino acid differences between this sequence and the best designed sequence determined using MSD-FASTER

Conclusions

We have presented implementation details of a new optimization algorithm for multi-state protein design based on FASTER, determined acceptable parameters for its use, and compared its performance to a multi-state implementation of Monte Carlo. Accurate scoring of sequences suggested by the MSD algorithms is required for efficient multi-state optimization; rotamer optimization routines for side-chain placement based on MC, FASTER, and iBR can all provide acceptable performance. Our results indicate that both MSD algorithms can find favorable sequences in realistic test cases for positive and negative design. Both algorithms can accommodate design problems with many states; even a 60-member NMR ensemble was designed without difficulty. In our hands, MSD-MC scales poorly compared to MSD-FASTER as the complexity of the design problem increases; the observed difference is much more pronounced than what has been reported for the single-state versions of these algorithms.[9] Due to this effect, the efficiency and consistency of MSD-FASTER was better than MSD-MC in every class of design problem we tested. In most cases, MSD-MC could not ever find the low-energy consensus solutions produced by MSD-FASTER. Given that the evaluation of each sequence is relatively time-consuming in MSD, MSD-FASTER likely performs better because it tends to make multiple substitutions simultaneously, and because substitutions are selected for scoring based on energetic considerations rather than randomly.

Although the general approach to multi-state design used by these MSD algorithms has met with several experimental successes already,[15, 17, 18] rigorous evaluation of energy functions and multi-state scoring functions will be required to prove

and improve the usefulness of this methodology. Realistic design procedures based on the explicit modeling of many native and non-native conformational states cannot be implemented without efficient optimization techniques to drive them. We hope that the greater optimization power of MSD-FASTER will help to accelerate progress in this area via its improved speed and accuracy compared to alternative methods.

Materials and methods

Design parameters: single-state design test cases

The energy functions and designed positions used for the single-state design problems were as previously described.[9]

For rotamer optmization, four of the weak RO algorithms (Null, MC, iBR, and FASTER) were paired with a strong rotamer optimizer utilizing two trajectories of FASTER with a maximum of 5 rounds of iBR and 3 rounds of sPR. When DEE was used as the weak rotamer optimizer, it was also used as the strong rotamer optimizer, as explained above. For the weak RO algorithms iBR and FASTER, the maximum number of nonproductive iBR rounds was 5. For FASTER, the iBR pass was followed by exactly one round of sPR. For those sequences for which DEE failed to converge, the strong FASTER rotamer optimization routine described above was automatically employed to find a reasonable approximate solution. The simulated annealing regimen for MC when used for weak RO comprised 1 cycle of 2.0×10^4 steps with a high temperature of 400 K and a low temperature of 1 K.

In MSD-FASTER, the FASTER parameters for sequence selection were: maximum nonproductive rounds in iBR, 5, maximum rounds in sPR, 5, and number of

relaxing positions in each step of sPR, 10.[9] In every MSD-MC calculation, the high and low temperatures for sequence selection were also set to 400 K and 1 K, respectively. The number of cycles and steps of MSD-MC was set in each calculation so that total time used by MSD-FASTER and MSD-MC would be comparable. The following simulated annealing schedules were used for sequence selection in each algorithm combination: MSD-MC/Null, 10 cycles of 1.0×10^6 steps; MSD-MC/MC, 1 cycle of 2.5×10^4 steps; MSD-MC/iBR, 1 cycle of 1.0×10^5 steps; MSD-MC/FASTER, 1 cycle of 1.5×10^4 steps.

Design parameters: 1GB1

The 1GB1 ensemble of protein G[20] was prepared and designed as follows. Hydrogens were removed from each ensemble member and added back in optimized positions using REDUCE.[27] Each structure was then standardized via 50 steps of conjugate-gradient minimization with the DREIDING force field.[28] All positions were classified as core, boundary, or surface as described previously[1] based on the coordinates of the crystal structure (1PGA).[29] The core+boundary design comprised positions 1, 3, 5, 7, 11, 12, 16, 18, 20, 23, 25, 26, 27, 29, 30, 33, 34, 37, 39, 43, 45, 50, 52, 54, and 56; the surface design comprised positions 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, and 55. In the core+boundary design, the amino acid types Ala, Val, Leu, Ile, Phe, Tyr, and Trp were allowed at each designed core position; Ala, Val, Leu, Ile, Phe, Tyr, Trp, Ser, Thr, Asn, Gln, Asp, Glu, His, Lys, and Arg were allowed. In the surface design, Ala, Ser, Thr, Asn, Gln, Asp, Glu, His, Lys, and Arg were allowed. For each design, we used rotamers from the Dunbrack backbone-dependent rotamer library.[30] There were an average of 3634 total rotamers per state with

rotamer/template energies better than 20 kcal/mol for the core+boundary design, and 5617 for the surface design. Pairwise energies were computed using energy functions as previously described,[7] except the polar hydrogen burial term was omitted.

For the core+boundary design, the following parameters were used for each MSD-MC algorithm combination: MSD-MC/FASTER (no US/CPL), 1 cycle of 2.0×10^4 steps; MSD-MC/FASTER, 1 cycle of 3.5×10^4 steps; MSD-MC/Null, 1 cycle of 5.0×10^5 steps; MSD-MC/iBR, 1 cycle of 1.0×10^5 steps.

For the surface design, the following parameters were used: MSD-MC/FASTER (no US/CPL), 1 cycle of 6.0×10^3 steps; MSD-MC/FASTER, 1 cycle of 1.3×10^4 steps; MSD-MC/Null, 1 cycle of 5.0×10^5 steps; MSD-MC/iBR, 1 cycle of 6.5×10^4 steps.

The number of MSD-MC steps in each case was chosen to make the average time per trajectory similar to MSD-FASTER. Equation 4 was used with kT = 300 kcal/mol to combine the energies from all 60 ensemble members into overall scores.

Design parameters: CaM

The two CaM structures were prepared and minimized as described above for the 1GB1 structures. Chains B and F were used from the 1CDL structure and chains A and E were used from the 1MXE structure. The amino acid types Ala, Val, Leu, Ile, Phe, Tyr, Trp, Met, and Glu were allowed at each of the following designed positions on the CaM chain: 7, 8, 11, 14, 15, 28, 32, 35, 47, 51, 64, 67, 68, 80, 84, 87, 88, 101, 104, 105, 108, 120, 124, 140, and 141. The 19 positions of the smMLCK peptide in the positive design state and the 25 positions of the CaMKI peptide in the negative design state were allowed to vary side-chain conformation but not amino acid identity. Side-chain conformations at

the variable positions were from the Dunbrack backbone-dependent rotamer library with expansions of ±1 standard deviation about $\chi 1$ and $\chi 2$. The same energy functions were used to compute pairwise energies as for the 1GB1 designs described above. For the multi-state design calculations, all rotamer-backbone and rotamer-rotamer energies on the negative design target state were capped at 50 kcal/mol. To compute σ during the optimizations, equation 5 was used with $W = 0.04$. The single-state design optimizations were performed as described,[9] without the initial elimination of rotamers using DEE.

The following parameters were used for each MSD-MC algorithm combination: MSD-MC/FASTER (no US/CPL), 1 cycle of 2.0×10^3 steps; MSD-MC/FASTER, 1 cycle of 6.0×10^3 steps; MSD-MC/Null, 25 cycles of 1.0×10^6 steps; MSD-MC/iBR, 1 cycle of 3.0×10^4 steps.

Acknowledgements

The authors thank Kyle Lassila, Christina Vizcarra, Jennifer Keeffe, and an anonymous reviewer for their insightful comments. This work was supported by the Howard Hughes Medical Institute, the Ralph M. Parsons Foundation, an IBM Shared University Research Grant, and the Defense Advanced Research Projects Agency.

References

1. Dahiyat, B. I.; Mayo, S. L., De novo protein design: Fully automated sequence selection. *Science* **1997,** *278* (5335), 82–87.

2. Gordon, D. B.; Marshall, S. A.; Mayo, S. L., Energy functions for protein design. *Current Opinion in Structural Biology* **1999,** *9* (4), 509–513.

3. Kuhlman, B.; Dantas, G.; Ireton, G. C.; Varani, G.; Stoddard, B. L.; Baker, D., Design of a novel globular protein fold with atomic-level accuracy. *Science* **2003,** *302* (5649), 1364–1368.

4. Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E., Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics* **1953,** *21* (6), 1087–1092.

5. Voigt, C. A.; Gordon, D. B.; Mayo, S. L., Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *Journal of Molecular Biology* **2000,** *299* (3), 789–803.

6. Desmet, J.; Demaeyer, M.; Hazes, B.; Lasters, I., The Dead-End Elimination Theorem and Its Use in Protein Side-Chain Positioning. *Nature* **1992,** *356* (6369), 539–542.

7. Gordon, D. B.; Hom, G. K.; Mayo, S. L.; Pierce, N. A., Exact rotamer optimization for protein design. *Journal of Computational Chemistry* **2003,** *24* (2), 232–243.

8. Desjarlais, J. R.; Handel, T. M., De-Novo Design of the Hydrophobic Cores of Proteins. *Protein Science* **1995,** *4* (10), 2006–2018.

9. Allen, B. D.; Mayo, S. L., Dramatic performance enhancements for the FASTER optimization algorithm. *Journal of Computational Chemistry* **2006,** *27* (10), 1071–1075.

10. Desmet, J.; Spriet, J.; Lasters, I., Fast and Accurate Side-Chain Topology and Energy Refinement (FASTER) as a new method for protein structure optimization. *Proteins-Structure Function and Genetics* **2002,** *48* (1), 31–43.

11. Jiang, L.; Althoff, E. A.; Clemente, F. R.; Doyle, L.; Rothlisberger, D.; Zanghellini, A.; Gallaher, J. L.; Betker, J. L.; Tanaka, F.; Barbas, C. F.; Hilvert, D.; Houk, K. N.; Stoddard, B. L.; Baker, D., De novo computational design of retro-aldol enzymes. *Science* **2008,** *319* (5868), 1387–1391.

12. Looger, L. L.; Dwyer, M. A.; Smith, J. J.; Hellinga, H. W., Computational design of receptor and sensor proteins with novel functions. *Nature* **2003,** *423* (6936), 185–190.

13. Malakauskas, S. M.; Mayo, S. L., Design, structure and stability of a hyperthermophilic protein variant. *Nature Structural Biology* **1998,** *5* (6), 470–5.

14. Rothlisberger, D.; Khersonsky, O.; Wollacott, A. M.; Jiang, L.; DeChancie, J.; Betker, J.; Gallaher, J. L.; Althoff, E. A.; Zanghellini, A.; Dym, O.; Albeck, S.; Houk, K. N.; Tawfik, D. S.; Baker, D., Kemp elimination catalysts by computational enzyme design. *Nature* **2008,** *453* (7192), 190–U4.

15. Havranek, J. J.; Harbury, P. B., Automated design of specificity in molecular recognition. *Nature Structural Biology* **2003,** *10* (1), 45–52.

16. Wuthrich, K., Protein structure determination in solution by NMR spectroscopy. *Journal of Biological Chemistry* **1990,** *265* (36), 22059–22062.

17. Ambroggio, X. I.; Kuhlman, B., Computational design of a single amino acid sequence that can switch between two distinct protein folds. *Journal of the American Chemical Society* **2006,** *128* (4), 1154–61.

18. Boas, F. E.; Harbury, P. B., Design of protein-ligand binding based on the molecular-mechanics energy model. *Journal of Molecular Biology* **2008,** *380* (2), 415–424.

19. Bolon, D. N.; Grant, R. A.; Baker, T. A.; Sauer, R. T., Specificity versus stability in computational protein design. *Proceedings of the National Academy of Sciences of the United States of America* **2005,** *102* (36), 12724–12729.

20. Gronenborn, A. M.; Filpula, D. R.; Essig, N. Z.; Achari, A.; Whitlow, M.; Wingfield, P. T.; Clore, G. M., A novel, highly stable fold of the immunoglobulin binding domain of streptococcal protein G. *Science* **1991,** *253* (5020), 657–61.

21. Dahiyat, B. I.; Mayo, S. L., Probing the role of packing specificity in protein design. *Proceedings of the National Academy of Sciences of the United States of America* **1997,** *94* (19), 10172–10177.

22. O'Neil, K. T.; DeGrado, W. F., How calmodulin binds its targets: sequence independent recognition of amphiphilic alpha-helices. *Trends in Biochemical Sciences* **1990,** *15* (2), 59–64.

23. Shifman, J. M.; Mayo, S. L., Modulating calmodulin binding specificity through computational protein design. *Journal of Molecular Biology* **2002,** *323* (3), 417–423.

24. Shifman, J. M.; Mayo, S. L., Exploring the origins of binding specificity through the computational redesign of calmodulin. *Proceedings of the National Academy of Sciences of the United States of America* **2003,** *100* (23), 13274–13279.

25. Meador, W. E.; Means, A. R.; Quiocho, F. A., Target enzyme recognition by calmodulin: 2.4 A structure of a calmodulin-peptide complex. *Science* **1992,** *257* (5074), 1251–1255.

26. Clapperton, J. A.; Martin, S. R.; Smerdon, S. J.; Gamblin, S. J.; Bayley, P. M., Structure of the complex of calmodulin with the target sequence of calmodulin-dependent protein kinase I: studies of the kinase activation mechanism. *Biochemistry* **2002,** *41* (50), 14669–14679.

27. Word, J. M.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C., Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *Journal of Molecular Biology* **1999,** *285* (4), 1735–1747.

28. Mayo, S. L.; Olafson, B. D.; Goddard, W. A., Dreiding — a Generic Force-Field for Molecular Simulations. *Journal of Physical Chemistry* **1990,** *94* (26), 8897–8909.

29. Gallagher, T.; Alexander, P.; Bryan, P.; Gilliland, G. L., 2 Crystal-Structures of the B1 Immunoglobulin-Binding Domain of Streptococcal Protein-G and Comparison with NMR. *Biochemistry* **1994,** *33* (15), 4721–4729.

30. Dunbrack, R. L.; Cohen, F. E., Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science* **1997,** *6* (8), 1661–1681.

Chapter 4

Development and Validation of Methods for Multi-State Design and Combinatorial Library Design

Adapted from a manuscript coauthored with Alex Nisthal and Stephen L. Mayo.

Abstract

The stability, activity, and solubility of a protein sequence are determined by a delicate balance of molecular interactions in a wide variety of conformational states, including competing states and native conformational states. Even so, most computational protein design methods model sequences in the context of a single conformation representing the native state. Despite the potential for improved simulation accuracy when the native state is represented by an ensemble of related structures, such calculations have not been attempted due to the lack of sufficiently powerful optimization algorithms for multi-state design. Here, we have applied our multi-state design algorithm to study the potential utility of various forms of input structural data for design.

To facilitate this analysis, we developed new methods for the design and high-throughput stability determination of combinatorial mutation libraries based on protein design calculations. The application of these methods to the core design of a small model system produced many variants with improved thermodynamic stability, and showed that multi-state design methods can be applied to large structural ensembles without requiring the use of different rotamer libraries, energy functions, or design strategies. Stabilized variants were found in libraries based on each type of structural data we tested. Our library design method produced degenerate codon libraries that represented the underlying design calculations, and exhaustive screening of these libraries helped to clarify several sources of error in our designs that would have otherwise been difficult to ascertain.

The complete lack of correlation between our experimental and simulated stability values shows clearly that a design procedure need not reproduce experimental data

directly to generate many successful variants. This surprising result suggests a potential new direction for the improvement of protein design technology.

Introduction

During the past two decades, protein-engineering efforts based on directed evolution have met with considerable success.[1–3] In tandem, structure-based computational protein design (CPD) methods have been developed to allow screening for desirable sequences to be performed *in silico*.[4–6] Despite a number of high-profile results that demonstrate the potential of CPD,[7–14] the routine computational design of functional proteins remains elusive. Thus, many current efforts focus on the improvement of CPD methodology or on the synergistic application of CPD with experimental high-throughput screening or selection.[15] These lines of inquiry need not be orthogonal; the computational design and experimental screening of mutant libraries can facilitate a more thorough evaluation of CPD than studies that focus on the comparison of individual designed sequences.

Here, we have applied this type of hybrid approach to investigate the degree to which X-ray crystallographic structures, NMR solution structures, and ensembles derived from molecular dynamics simulations can serve as useful sources of structural information for CPD. This study was made possible by the development of new methods for the computational design and high-throughput experimental stability determination of combinatorial protein libraries. The results we report here provide simultaneous experimental validation for (1) the application of multi-state protein design methods to large conformational ensembles, (2) the transformation of arbitrary CPD results into combinatorial mutation libraries, and (3) the experimental stability determination of these

libraries by high-throughput gene assembly, protein expression, purification, and screening.

Our work here was motivated by a desire to address one of the major approximations of CPD: the reliance on a single, rigid main-chain conformation. Although the stability, solubility, and activity of a protein depend on the relative energetic contributions of many conformational states, including ensembles of native, unfolded, and aggregated structures,[16] most CPD methods evaluate sequences based on their energies in the context of one fixed backbone structure. This simplification has made design results undesirably sensitive to slight changes in main-chain and side-chain conformation, and has made difficult the selection of sequences with amino acid composition similar to naturally occurring protein. These issues have been approached via the use of high-resolution structural templates, expanded rotamer libraries,[17, 18] energy functions with softened repulsive terms,[11, 19, 20] iteration between structural refinement and sequence design,[11, 21] and composition-based reference energies.[11, 22] Although these strategies can help to mitigate the impact of the fixed-backbone approximation, they do not address the fundamental reality that protein fitness depends on a diverse range of conformational states.

In a handful of cases, multi-state design (MSD) procedures have been used to find sequences that simultaneously stabilize or destabilize a combination of a few different conformational states.[23–25] However, MSD techniques have not yet been applied to ensembles with many conformational states that might better reflect the flexibility of real proteins. The degree to which various energy functions, rotamer libraries, and structural templates of single-state design (SSD) might be appropriate for this type of MSD

calculation is heretofore unknown. We recently developed a framework for MSD that allows for efficient sequence optimization given hundreds of conformational states. Here, we have applied this framework to test the applicability of current CPD methods to large structural ensembles, and to investigate whether the use of such ensembles might result in the selection of more desirable sequences by CPD.

With limited exceptions,[26] a unique native state with at least marginal stability is required for protein function as we understand it today. Accordingly, the most basic goal of CPD has been to optimize interactions between amino acids side chains to promote thermodynamic stability of the native state. Unfortunately, the experimental validation of a new design procedure on this basis is often beset with uncertainty. Standard methods for the measurement of protein stability are too laborious to allow the testing of more than a few designed variants, and the top-scoring sequence produced by a new design procedure does not (yet) sufficiently reflect its general utility. To facilitate the experimental evaluation of larger numbers of designed sequences, higher throughput is required in the assembly of genes, the expression and purification of proteins, and the measurement of stabilities. Fortunately, recent progress in these areas has allowed us to construct an efficient pipeline for the basic evaluation of new procedures in CPD. Gene libraries assembled from degenerate oligonucleotides, a frameshift selection scheme that reduces contamination by erroneous genes,[27] and economical sequence verification make tenable the production of numerous specific designed genes. Commercial microtiter plates for the growth of expression cultures and the purification of hexahistidine-tagged proteins allow sufficiently pure protein to be produced easily from these genes. Finally, liquid-handling robotics[28] expedites the preparation of a chemical denaturation series for

each protein in 96-well format, and the fraction of protein unfolded in each well is assayed in a plate reader measuring tryptophan fluorescence.[29] The integration of these technologies has allowed us to assess the stability of hundreds of designed protein variants with minimal experimenter intervention and limited incremental expense.

Given several design procedures to evaluate and a high-throughput experimental assay, we needed a general and rigorous method to choose a limited number of representative sequences to test from each design. Fortunately, structure-based computational protein design methods have been enlisted previously to focus high-throughput screening and selection on desirable subsets of sequence space. For example, CPD can be used to help identify positions amenable to site-saturation mutagenesis[30] and site-directed recombination.[31, 32] When a protein engineering effort is intended to help evaluate CPD procedures, as in this case, designed combinatorial mutation libraries are more appropriate because they reflect more strongly the sequence preferences of CPD. Although several useful computational protein library design methods have been developed, none reported so far takes directly into account CPD energies, allows control over library size and possible sets of amino acids, and eschews heuristics that can introduce bias into the libraries it produces.[33–36] So that our experimental results might better reflect the results of the underlying CPD calculations, we developed a new library design procedure, called Combinatorial Libraries Emphasizing And Reflecting Scored Sequences (CLEARSS), which satisfies all of these criteria.

We used standard single-state design (SSD) and MSD to redesign the core of the small, stable domain Gβ1 based on several sources of structural information, including a crystal structure, an NMR structure, and MD simulations. Our efforts were motivated by

a curiosity about the relative merits of different sources of structural data for design, and the hypothesis that use of a structural ensemble might help to correct for design failures observed in SSD. Because the imperfect nature of CPD limits the conclusions that can be drawn from a comparison of single sequences, we developed the CLEARSS algorithm to make combinatorial libraries based on the lists of scored sequences produced by CPD. We applied this algorithm to the results of our design calculations, and assayed the designed libraries using a new protocol for the expression, purification, and stability assessment of protein libraries with high throughput.

We found that all three sources of structural data resulted in designed libraries with multiple stabilized variants. The designed libraries based on an NMR ensemble were extremely similar, whether a single representative structure or all 60 ensemble members were used for modeling. The most promising results by far were achieved using a constrained 128-member MD-ensemble, which produced a designed library with no significantly destabilized and many stabilized variants. Despite the apparent success of this design, there was no correlation observed between the simulation energies and the experimental stabilities of any of these variants.

Our results suggest that the basic principles of CPD extend beyond the design of single sequences to the design of combinatorial libraries, and that the rigorous screening of such libraries can help to pinpoint sources of error in a design procedure. They show that MSD methods are applicable to large structural ensembles when used with standard rotamer libraries and energy functions, inspiring optimism about more ambitious future applications for MSD. They also hint that the use of structural ensembles could help to alleviate problems that occur when targeting a single, fixed input structure. Furthermore,

they illustrate clearly that the success of CPD does not hinge on its ability to directly correlate simulation energies with experimental measures of fitness. This surprising property of CPD may suggest a new possible direction of inquiry for the improvement of CPD.

Results and discussion

Designed libraries

To simplify the validation of our multi-state design and combinatorial library design methods, we applied them to a previously studied set of core positions (Figure 1) in the small model system Gβ1, and relied on a set of energy functions that previously found stabilized variants based on this design.[19] Given the requirements for purified protein of our stability assay, we chose to design and screen a 24-member library based on each of the following sources of structural information: a crystal structure (xtal-1), an NMR-constrained minimized average (NMR-1), an NMR ensemble (NMR-60), a constrained MD ensemble (cMD-128), and an unconstrained MD ensemble (uMD-128).

The sequence of steps used to design the combinatorial libraries we tested experimentally is depicted in Figure 2. First, the standard design procedure was applied to each structural input, and optimization was performed with SSD-FASTER or MSD-FASTER to give a list of amino acid sequences and their CPD energies for each design. The CLEARSS library design algorithm was then applied to each list of sequences to give a rank-ordered list of combinatorial mutation libraries. All amino acid sequences in each of the top 20 CLEARSS libraries were instantiated and evaluated by rotamer optimization. The CLEARSS library to test experimentally for each structural input was chosen by objective criteria based on the energies of the rescored sequences, as described in the methods section.

All five designed libraries comprise relatively conservative sets of mutations away from the wild-type sequence (Table 1). All libraries other than uMD-128 share many characteristics in common. Each of these libraries chose only the wild-type amino acid at positions A20, A26, F30, and A34. Every member of each of these four libraries contained the single-mutant Y3F, which previous experiments have shown to be well tolerated by the structure. These four libraries all allowed the wild-type amino acid at every other position, and all contain the most stable Gβ1 core variant previously characterized (Y3F+L7I+V39I).

The two NMR libraries were extremely similar to each other: both chose the amino acids FILV at position 52, and directed the remaining diversity to positions 7 and 39. In contrast, xtal-1 and cMD-128 allowed only the wild-type Phe at position 52, and instead allocated diversity towards positions 7, 39, and 54. xtal-1 differs from cMD-128 in that it gave up L7F and V39L to allow L5I. The unconstrained MD ensemble library uMD-128 was the least conservative, specifying a size reversal of two nearby residues via mutations L5A and A34F, and diversity at residue 30, a position untouched in the other libraries.

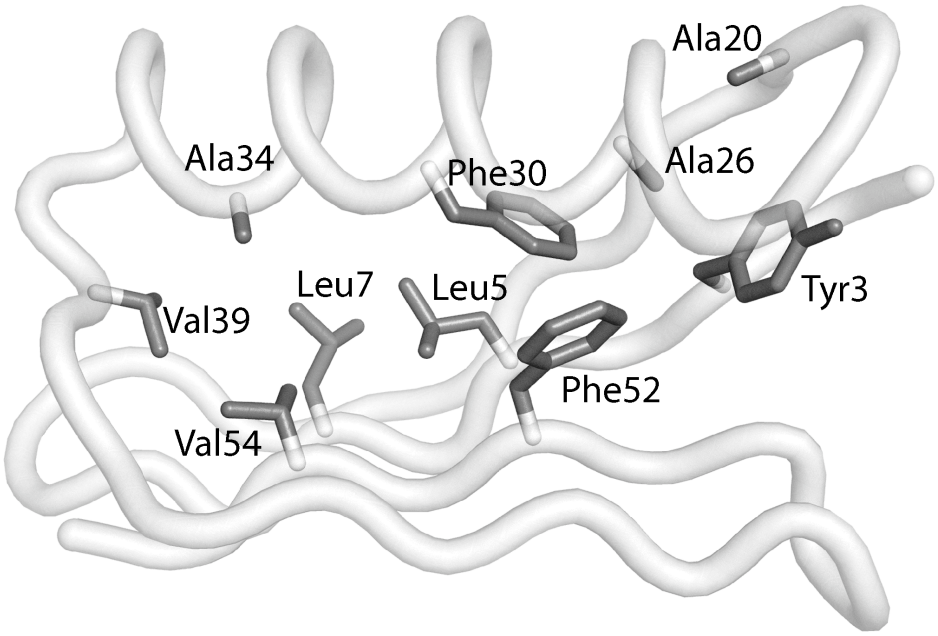

Figure 1: The core residues of Gβ1 designed in this study. Each of these positions was allowed to assume various rotamers of the hydrophobic amino acids Ala, Val, Ile, Leu, Phe, Tyr, and Trp. Position Trp43 (not shown) was additionally allowed to change rotamer but not amino acid type. All other side chains and the main chain were fixed in the input conformation for the state being modeled in each case.

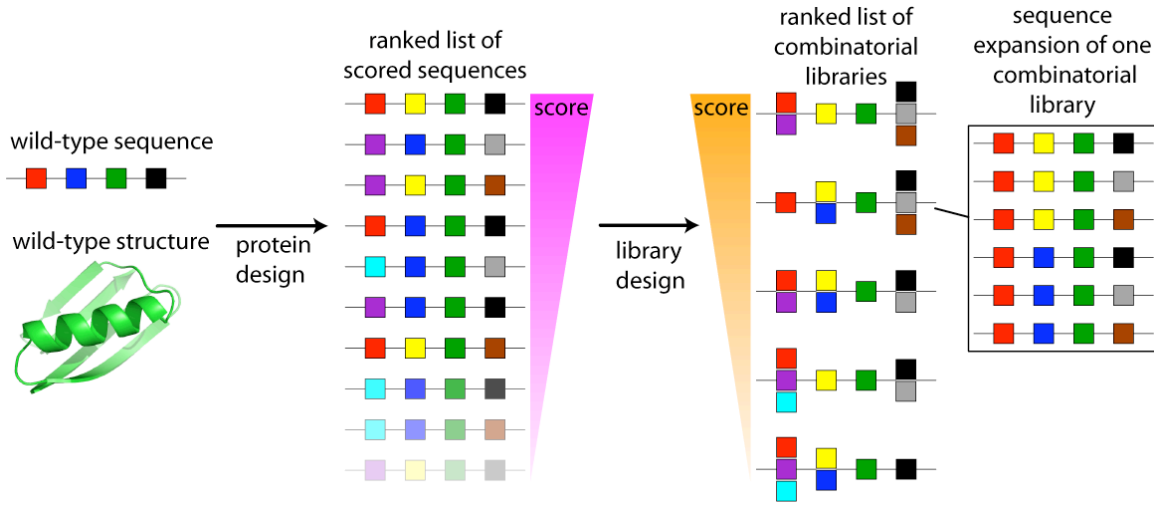

Figure 2: The general scheme used to design combinatorial mutation libraries based on computational protein design calculations. A line of boxes indicates a protein sequence; each box represents a position in the protein chain. Different colored boxes represent different amino acids. The set of sequences on the far right represent the expansion of a particular combinatorial library into the set of sequences it represents. The energies of the sequences in the expansions are used to decide which combinatorial library to test experimentally, as described in the Methods section.

Table 1: Combinatorial libraries designed from different sources of structural information. **xtal-1**: the designed library based on single-state design of the crystal structure. **NMR-1**: the library based on single-state design of the constrained minimized average NMR solution structure. **NMR-60**: the library based on multi-state design of the 60-member NMR structural ensemble. **cMD-128**: the library based on multi-state design of the constrained molecular dynamics ensemble. **uMD-128**: the library based on multi-state design of the unconstrained molecular dynamics simulation.

Residue	WT	xtal-1	NMR-1	NMR-60	cMD-128	uMD-128
3	Y	F	F	F	F	F
5	L	IL	L	L	L	A
7	L	ILV	ILV	IL	FILV	FL
20	A	A	A	A	A	A
26	A	A	A	A	A	A
30	F	F	F	F	F	FIL
34	A	A	A	A	A	F
39	V	IV	IV	ILV	ILV	IL
52	F	F	FILV	FILV	F	F
54	V	IV	V	V	IV	AV

Experimental characterization of designed libraries

Each library was constructed using a modification of the traditional gene assembly protocol[37] that minimizes oligonucleotide overlap. These changes were intended to limit oligonucleotide costs and allow degenerate nucleotides to be placed in non-overlapping regions, limiting library composition biases produced by differential annealing effects. Expensive and time-consuming oligonucleotide purification was omitted; instead, a frameshift selection plasmid pInSALect was applied to correct for errors introduced during oligonucleotide synthesis and PCR assembly.[27] Over-sequencing (4x) of a 24-member library typically gave 85% correctly inserted, non-mutated sequences (see supplemental materials), out of which ~ 80% of each desired library could be recovered. Missing library members were generated by standard quick-change mutagensis.

The libraries were then expressed, purified, and denatured as described in the methods. Control experiments verifying the accuracy and precision of the microtiter plate-based stability assay showed excellent agreement with denaturation experiments monitored by circular dichroism (see supplemental materials). Future improvements in the throughput of stability determination can come from the usage of robotics platforms for variant construction, colony picking, and protein purification. Shifting the focus from sequencing towards stability screening could quickly produce information about the best mutants, as is common in directed evolution protocols. However, since a comprehensive screening of each designed library was desired, a lower level of throughput was tolerated.

Experimental screening of the xtal-1 library (Figure 3) showed two distinct sets of variants. The 12 library members with wild-type Leu at position 5 all exhibited stabilities similar to or better than the wild-type sequence, while the 12 with Ile at position 5 were all significantly destabilized. Screening of the NMR-based libraries (Figures 4 and 5) showed a similar dichotomy. In each case, the 6 library members with the wild-type Phe at position 52 exhibited wild-type-like stability or better. The remaining 18 variants from each NMR-based library were highly destabilized, and many lacked enough of a pretransition to be fit to the two-state unfolding model.

Evaluation of the MD libraries indicated that all 24 variants from the constrained library, cMD-128, had stability similar to the wild type or better (Figure 6). In contrast, all 24 variants from the uMD-128 library failed to produce any significant change in fluorescence signal across the denaturation series, and thus may be unfolded or structurally perturbed, as discussed below. A comparison of all five experimentally characterized libraries (Figure 7) indicates clearly that the cMD-128 design successfully produced a variety of stabilized mutants, whereas every other designed library specified at least one problematic substitution that rendered many of its sequences destabilized or otherwise unlike the wild type.

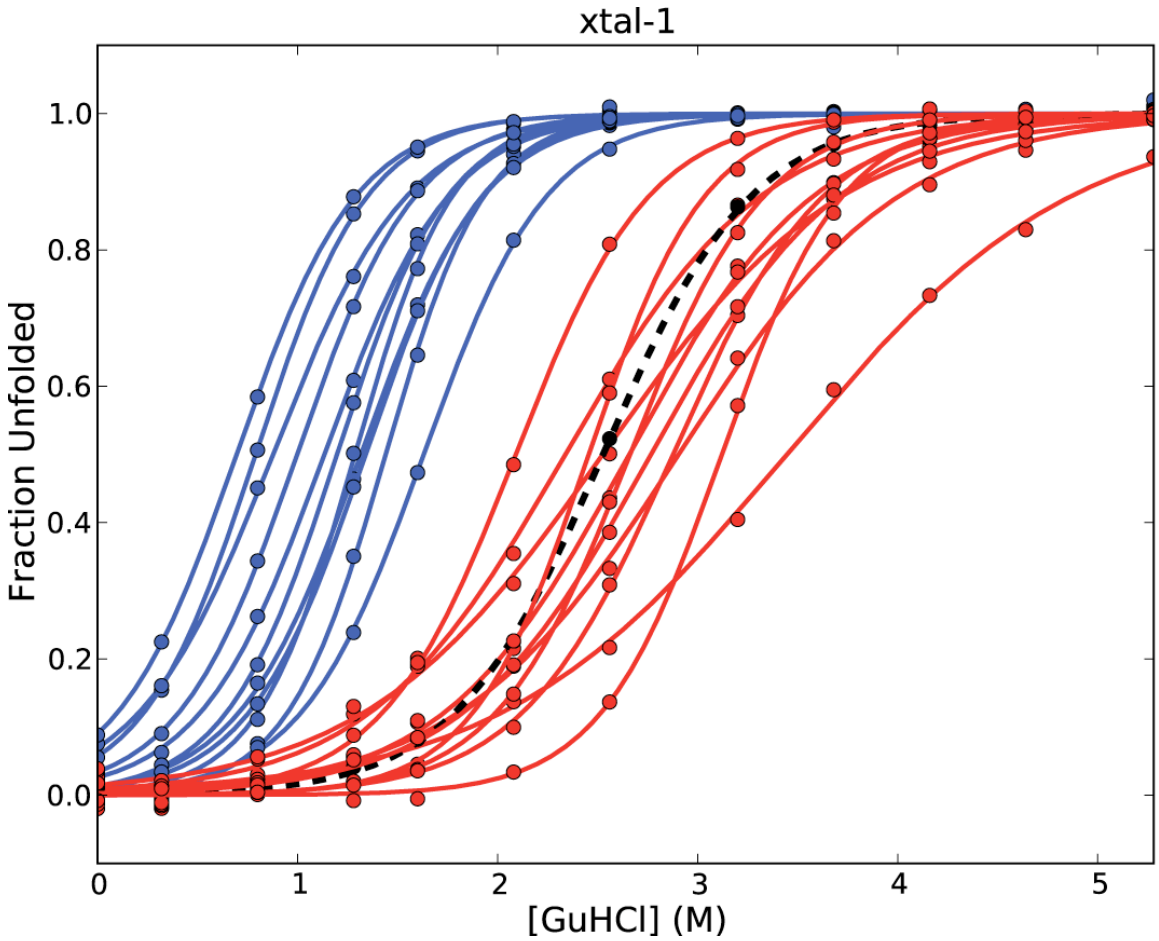

Figure 3: Fraction-unfolded curves derived from the stability determination of library xtal-1. The dashed black curve denotes variant Y3F, which is the closest library member to the wild type in terms of sequence, and which is known to have a stability very similar to the wild type. Red curves denote variants with C_m > 2.0 M, and correspond to all variants with Leu at position 5. Blue curves denote variants with C_m < 2.0 M, and correspond to variants with Ile at position 5. Not pictured: variant Y3F+L5I+L7I, which did not give a signal that could be fit to a two-state unfolding model.

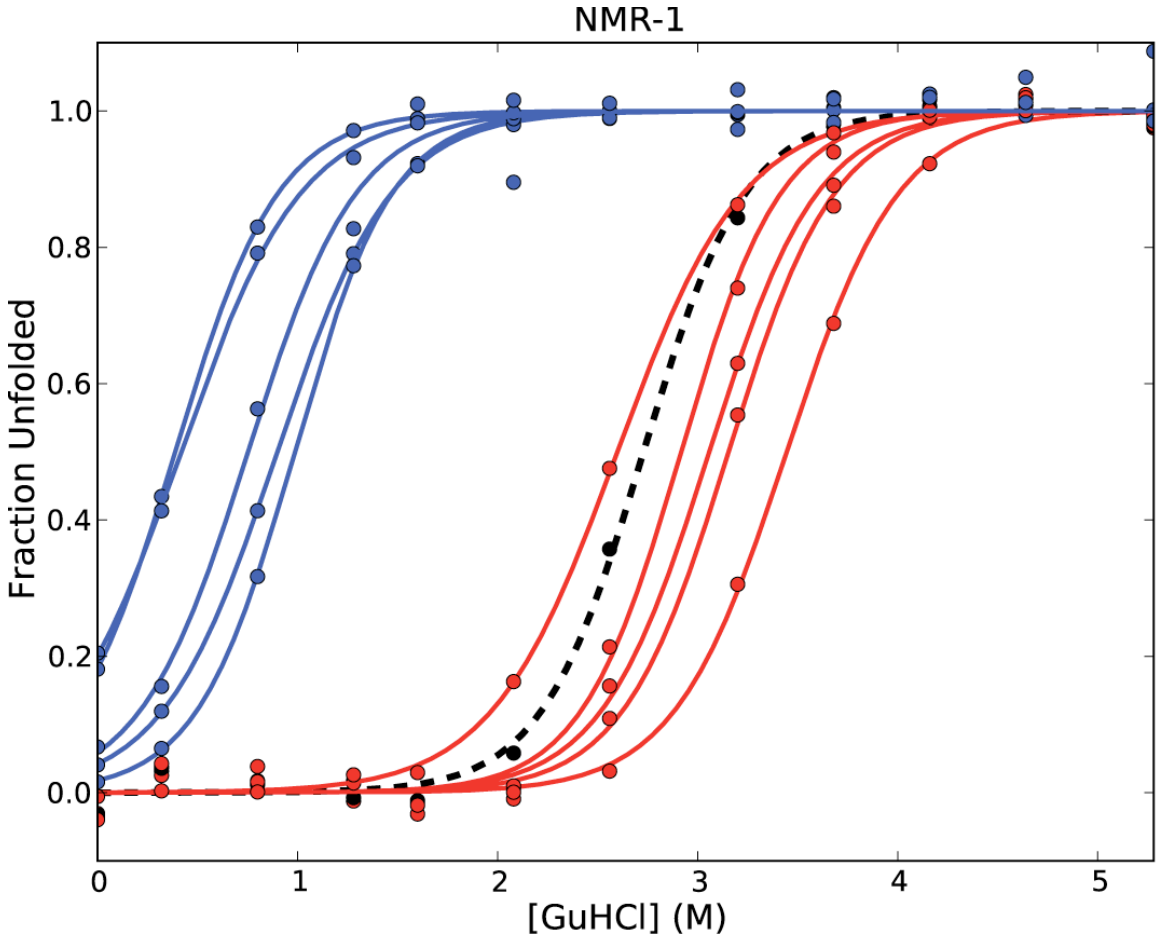

Figure 4: Fraction-unfolded curves derived from the stability determination of library NMR-1. The dashed black curve denotes variant Y3F, which is the closest library member to the wild type in terms of sequence, and which is known to have a stability very similar to the wild type. Red curves denote variants with C_m > 2.0 M, and correspond to all variants with Phe at position 52. Blue curves all represent variants with C_m < 2.0 M, which lack Phe at position 52, and have Val at position 39. Not pictured: 13 variants that lack Phe at position 52.

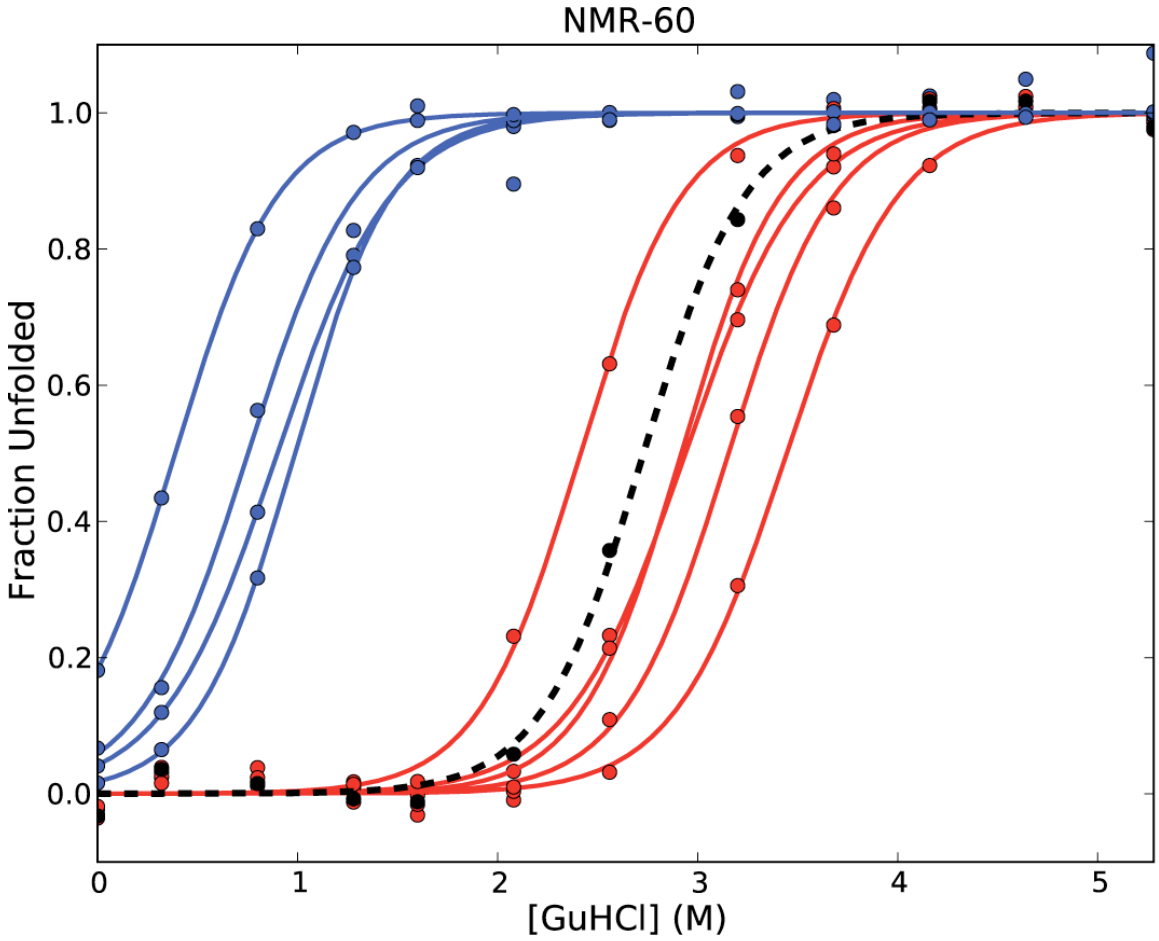

Figure 5: Fraction-unfolded curves derived from the stability determination of library NMR-60. The dashed black curve denotes variant Y3F, which is the closest library member to the wild type in terms of sequence, and which is known to have a stability very similar to the wild type. Red curves denote variants with $C_m > 2.0$ M, and correspond to all variants with Phe at position 52. Blue curves all represent variants with $C_m < 2.0$ M, which lack Phe at position 52, and have Val at position 39. Not pictured: 14 variants that lack Phe at position 52.

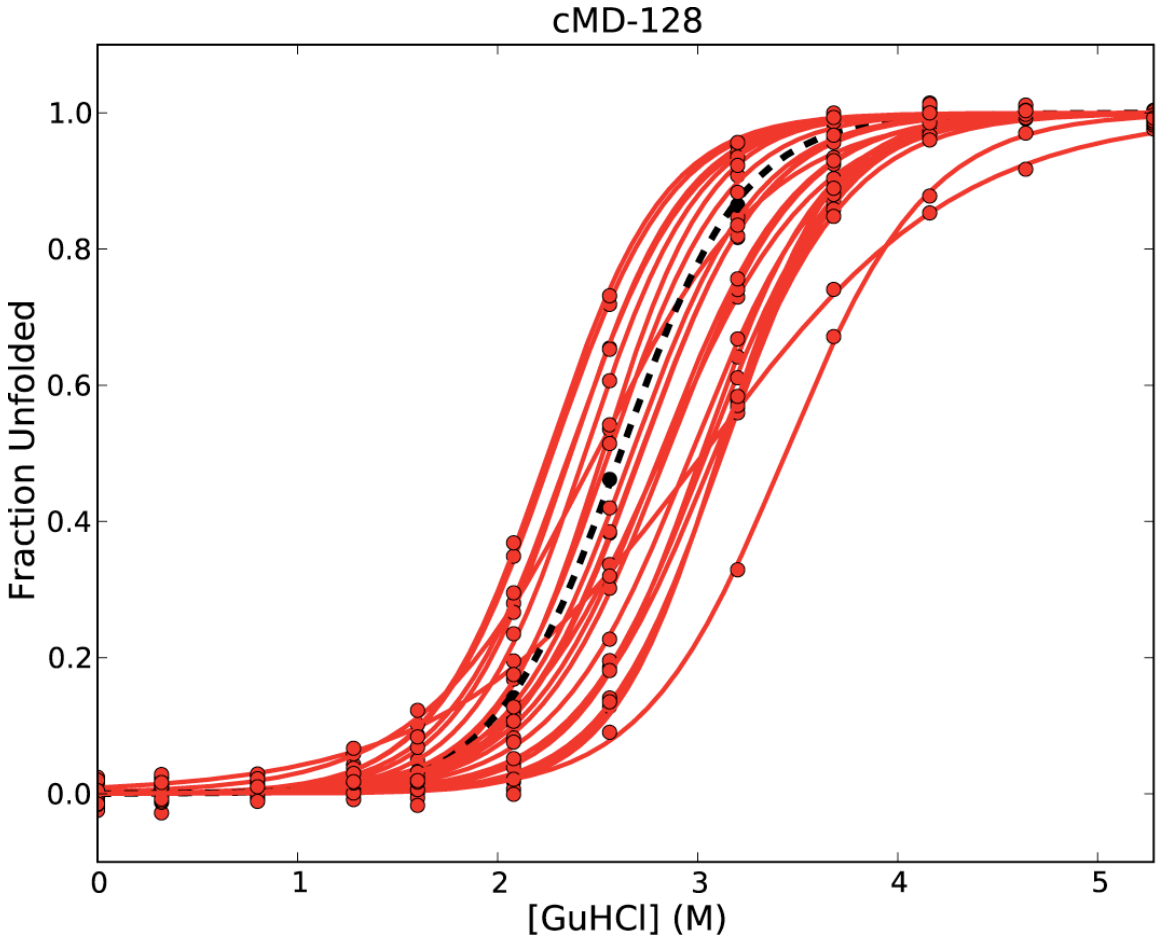

Figure 6: Fraction-unfolded curves derived from the stability determination of library cMD-128. The dashed black curve denotes variant Y3F, which is the closest library member to the wild type in terms of sequence, and which is known to have a stability very similar to the wild type. Red curves denote variants with C_m > 2.0 M, and correspond to all 24 variants in the library.

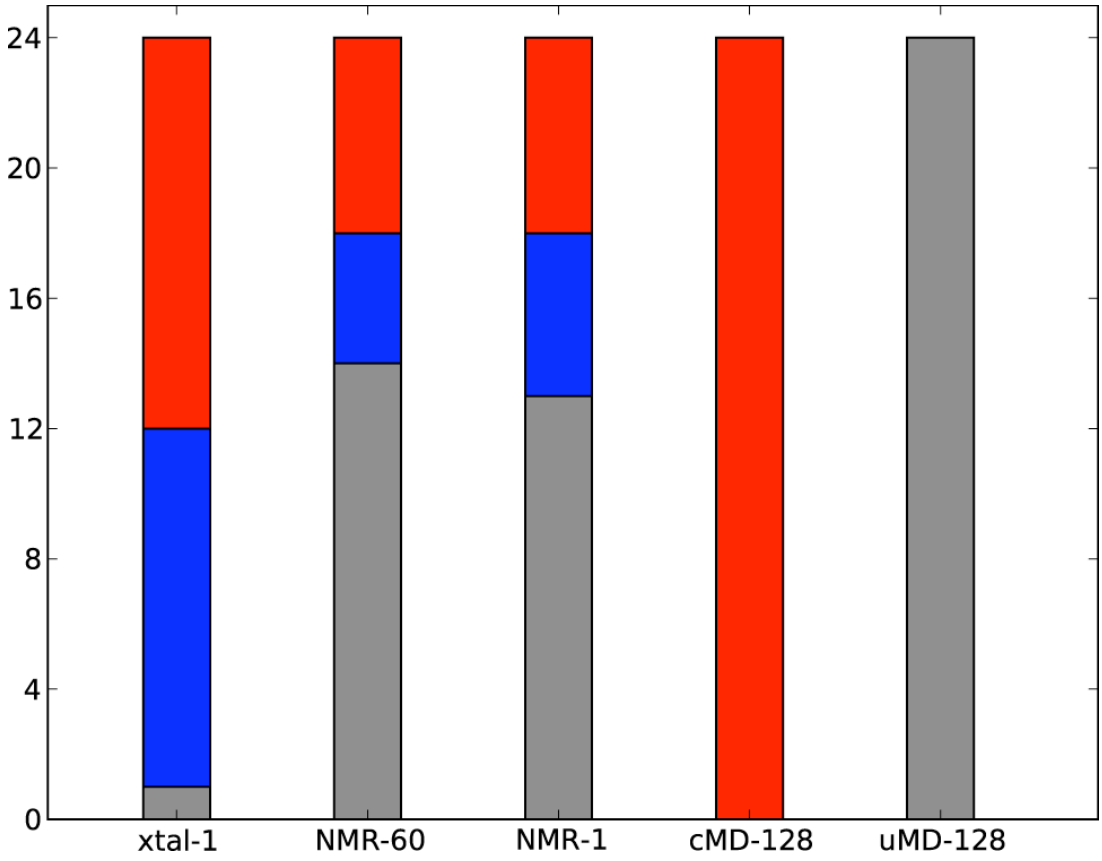

Figure 7: Each library partitioned into three stability groups. The colors match those in Figures 3–6: red (stable, $C_m > 2.0$), blue (destabilized, $C_m < 2.0$ M), grey (did not give a signal that could be fit to a 2-state model; not pictured in Figures 3–6).

Origin of destabilizing mutations

With experimental screening results in hand, we can return to the calculations that inspired them and ask why mutations such as L5I, F52ILV, and A34F were chosen by the design procedure. These mutations were all present in high-scoring sequences from the original design calculations, and thus are not artifacts introduced by the library design process.

The selection of the amino acids FILV at position Phe52 in the two NMR-based libraries resulted in three quarters of each library being significantly destabilized. In the context of the NMR structures, no Phe rotamer in the library was able to fit perfectly at position 52, encouraging the selection of smaller amino acids. If the set of rotamers at this position is supplemented with the observed rotamer in each structure, the design chooses to allocate diversity to positions 7 and 39, resulting in libraries similar to xtal-1. This result highlights how dramatically the rotameric approximation can influence the results of a design. It suggests that, at the very least, rotamers optimized for the wild-type sequence should be included when the goal is to find particular desirable sequences. In this case, we omitted the structurally observed rotamer at each position in order to limit the significant bias towards the wild-type sequence that these rotamers tend to cause. In the context of a real protein engineering project, this choice would have considerably reduced our chances of success.

The L5I mutation, which caused half of the xtal-1 library members to be destabilized relative to the wild-type sequence, may have been selected due to a failure of the softened repulsive contact potential that is used to counteract unrealistic rigidity introduced by the CPD model. The γ methyl group of Ile5 bumps into a Thr residue on

an adjacent β strand and is scored as a serious clash using unscaled van der Waals radii, but appears innocuous with the atomic radius scaling factor of $\alpha = 0.9$ that we used for the designs evaluated here. Repeating the design calculations with radii scaled by intermediate values such as 0.925 and 0.95 prevents Ile from being chosen at position 5, but also increases the frequency with which smaller residues are chosen at position Phe52. Interestingly, the recommendation of $\alpha = 0.9$ is derived from previous experiments based on the same set of Gβ1 core positions that were designed here. The earlier work drew conclusions based only on the best-scoring sequences produced by the design calculations, and found no difference between scaling atomic radii by 0.9 or 0.95.[19] Our results here indicate that the quality of sequences produced by the design procedure varies significantly with values of α between 0.9 and 0.95 when more sequences are taken into account. Given this, a more rigorous investigation of the most appropriate α value for design seems both tenable and warranted.

To analyze the uMD-128 data, it is important to note that our stability assay reports on the environment of the single Trp residue of Gβ1. Changes in packing caused by substitutions at other positions could alter the native-state environment of Trp43 enough to flip its side chain out into solution or change its fluorescence properties, crippling our ability to monitor unfolding by fluorescence. This interpretation seems unlikely for the destabilized members of the crystal structure and NMR libraries, for which a partial unfolding transition is clearly indicated by the raw data. However, the members of the uMD-128 library fail to show even a hint of such a transition, rendering the validity of our assay more suspect in this case.

Interestingly, others have investigated a 5-fold core variant of Gβ1 that bears substitutions similar to those in our uMD-128 library, including the A34F mutation. Structural characterization of this variant by NMR and X-ray crystallography indicated a domain-swapped tetrameric structure; the fluorescence emission maximum of this sequence was blue-shifted by almost 20 nm.[38] Related variants with the A34F substitution, including the A34F single mutant of the wild-type sequence, have also been shown to assume domain-swapped or side-by-side dimeric conformations in solution.[39, 40] Given these reports, the variants in our uMD-128 library, which all bear the A34F mutation, might also plausibly assume one of these oligomeric conformations. In this case, the library sequences could easily exhibit fluorescence emission spectra incompatible with our assay parameters, which were developed based on the characteristics of the wild-type sequence. Ultimately, the structural features of the uMD-128 library are unknown without additional experimental characterization. However, the published investigations of Gβ1 variants with the A34F substitution suggest that our uMD-128 library sequences are likely to assume conformations other than those modeled in our design calculations.

Influence of the designed library selection method

At this point, it is important to address the degree to which serendipity in designed library selection might affect the conclusions we may draw from our experiments. The CLEARSS library design procedure was developed with an understanding that many different combinatorial libraries may similarly represent a given list of scored sequences. Thus, its default mode of operation is to produce a list of the

top-scoring designed combinatorial libraries that satisfy all constraints, and to let the user choose between them. In general, this choice might be influenced by chemical intuition or prior mutational data, and thus partially account for properties of the system that are not modeled during the design procedure. To make our evaluation of input structural data sources as fair as possible, we chose to ignore such influences and apply an objective strategy based on the energies of the sequences in the libraries. Nevertheless, we must ask how other reasonable libraries generated by CLEARSS would have fared in our experimental assay.

Each of the top 20 designed libraries based on the NMR ensemble, as well as each based on the single average NMR structure, assigned smaller residues than the wild-type Phe to position 52. The remaining diversity of each library was occupied by various combinations of the other mutations present in the xtal-1, NMR-1, and NMR-60 libraries we screened in this work. It seems very likely, then, that the screening of any of the top NMR-based libraries from our designs would have resulted in stability data quite similar to that shown in Figures 4 and 5. Similarly, all of the top 20 designed libraries based on the unconstrained MD ensemble contained mutations L5A and A34F, and would be expected to exhibit similar fluorescence characteristics to the library uMD-128 we tested here.

A more interesting case is provided by the designs based on the crystal structure and constrained MD ensemble. Our analysis of the libraries xtal-1 and cMD-128 produced by these designs seems to indicate that cMD-128 was more successful, since a much greater fraction of its members were shown to be highly stable. However, when the top 20 libraries from each design were inspected in aggregate, it became clear that

both designs had produced a variety of libraries with various expected properties. The xtal-1 library and the cMD-128 library were each found in the top 20 libraries produced by both designs. Furthermore, each design produced several libraries with diversity at position 52, like NMR-1 and NMR-60. It seems clear that small changes to the constrained MD ensemble or to our energy functions might have reversed any potential conclusions about the usefulness of structural ensembles compared to single structures for the purposes of CPD.

The nature of approximation in computational protein design

In addition to helping validate the use of multi-state and combinatorial library design methods for computational protein design, our experimental results also allowed some unexpected insight into protein design itself. Plots of experimental stability versus simulation energy for the cMD-128 library (Figure 8) failed to yield any correlation, despite the apparent success of this design calculation. Likewise, the design calculations for xtal-1 and the NMR libraries failed to predict the pronounced destabilizing effects of mutations L5I or F52L, even though these designs also found a variety of stabilized variants. The design problem we chose is not simply too trivial for our purposes: the uMD-128 library and many previous reports attest to the myriad ways in which this system can be broken.[19, 38–42]

With a multiplicity of approximate methods available for computing the relative stabilities of protein sequences, the difficulty of solving this problem generally and accurately is sometimes overlooked. The stability of a sequence depends on the equilibrium between a relatively well-defined ensemble of native state conformations and

a vaguely defined ensemble of competing states. Our ability to find the relevant low-energy states is constrained by the vastness of protein conformational space and the extremely rugged energy landscape produced by our energy functions. Amino acid substitutions alter this energy landscape unpredictably, limiting the utility for design of structural information gathered for individual sequences. Current approaches tend to model native states at high resolution using whatever structures happen to be available, and account for competing states implicitly using statistical and heuristic terms.

Such methods have been surprisingly effective, given the approximations they rely upon. One perspective is that a CPD method is successful only to the extent that it can accurately predict or rank the stabilities of the variants it simulates, and that improvements in designed sequences will follow from improvements in ranking ability.[43] Accordingly, several groups have taken on large-scale forcefield parameterization efforts based on thermodynamic databases.[44, 45] In our research group, a forcefield tuned to offer significantly improved correlation between simulated and experimental stability differences did not exhibit improved performance for combinatorial design methods that allow large jumps in sequence space.[45] We can infer the same about the tuned forcefield of another group, given several reports of successful designs based on iterative one-by-one design and none based on combinatorial design methods.[46–50] The ability to reproduce experimental stability rankings is apparently not sufficient for accurate combinatorial protein design, at least in the range of ranking accuracy that has been achieved so far. The results of our work here furthermore suggest that this property is not even necessary for effective design.

This perspective prompts a modified view of the factors that make structure-based protein design possible in the first place. As discussed above, protein structures relax to accommodate mutations, and the computational difficulty of simulating these relaxations accurately has so far rendered intractable the stability ranking of sequence variants with many mutations. Fortunately, this malleability also means that sequences chosen to fit into a rigid protein model, even using approximate energy functions, will likely be tolerated by whatever relaxed structure results from the mutations they contain. In this way, the soft material properties of proteins impede the development of the quantitative protein design method we seek, but also make possible the more qualitative methods we can apply today.

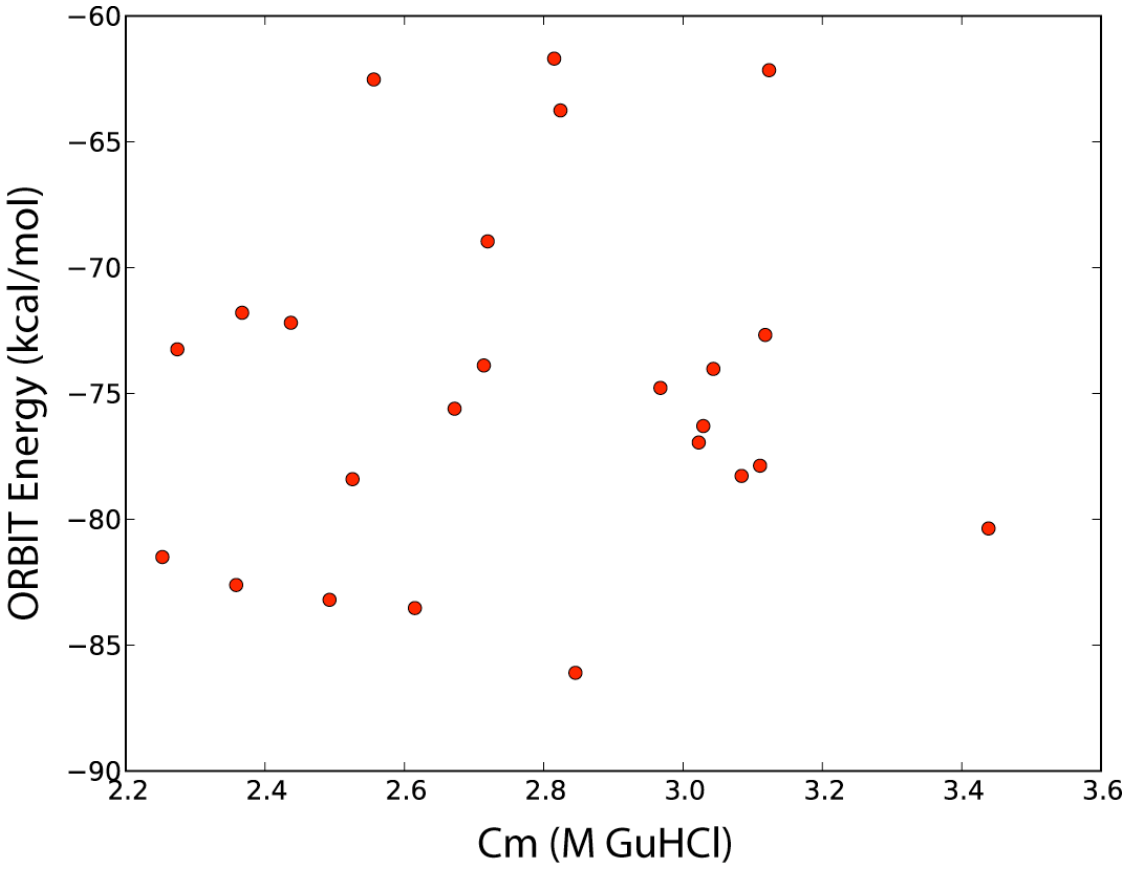

Figure 8: Correlation between simulation energy and experimental stability for the cMD-128 library. No correlation was observed between the experimentally measured fitness of the sequences and simulation energies that were used to select them for experimental screening.

Conclusions

Here, we have reported the development of new methods for the design and stability screening of combinatorial libraries based on lists of scored sequences. These methods were enlisted to test the application of multi-state design procedures to several structural ensembles, and to compare the resulting designs to those based on single structures. Designed libraries gave multiple stabilized variants when based on a crystal structure, an MD trajectory from that crystal structure, an NMR ensemble, and a single structure derived from the NMR ensemble. Our single-state and multi-state designs based on NMR data produced similar sets of libraries; likewise did those based on crystallographic data. Although an MD-based library gave superlative results, we cannot definitively conclude that the use of a structural ensemble provides any particular advantage over a single high-resolution structure for the purposes of design. Nevertheless, this initial success seems intriguing and warrants additional study. It seems clear that the energy functions and rotamer libraries developed for single-state modeling are equally applicable to the multi-state design of large structural ensembles. This result has important ramifications for future methods in CPD: even if structural ensembles fail to prove useful in the modeling of native states, they are expected to be crucial in the accurate modeling of competing states, which are undoubtedly more diverse.

In addition to validating the idea of design based on large structural ensembles, our work has provided further support in favor of rigorously screening an area of sequence space discovered by simulation, and has helped in vetting our new, general method for library design. For some designs that specified undesired destabilizing mutations, library screening suggested underlying causes for design failure that would not

have been apparent via the ad-hoc testing of individual sequences. Because our library design procedure is specifically intended to faithfully represent its input scored sequence list, and is indifferent to the origin of the list, it should be more useful for the evaluation of new design procedures than its predecessors.

Finally, the observed lack of correlation between experimental and simulated stabilities in our relatively successful sets of designed sequences may suggest a modified approach to protein design. Current design procedures seem to find stable sequences by selecting mutations that are likely to be accommodated by a relaxed version of the template structure, and not by accurately ranking the mutations relative to each other. In this view of design, finding sequences that satisfy the native state is relatively easy, while deciding which sequences satisfy it best is considerably more difficult. Given that stability is a function of nonnative states as much as native ones, the implication is that additional effort should be directed more toward eliminating sequences that can favorably assume competing states and less toward attempting to accurately predict which will best satisfy the native state. Since the relevant competing states under nondenaturing conditions likely exhibit significant residual structure, their treatment will probably require more sophisticated techniques than the composition-based heuristic terms used today. An interesting initial approach might be to perform multi-state design with an ensemble of native states as the positive design target and an ensemble of perturbed or expanded native states as the negative design target. The hypothesis is that selecting sequences to satisfy the compact native state and to not satisfy an expanded native state would tend to promote the desired specificity of a well-folded protein. Whether or not this type of strategy proves successful depends on the degree to which nonnative states

influence free energies of folding in a sequence-dependent (rather than composition-dependent) manner, and on the accuracy with which negative design can be performed against a computationally tractable set of competing states. Ultimately, techniques for native-state structural refinement will be crucial in the improvement of variant ranking; such methods may profitably be applied to produce appropriate nonnative ensembles as well. The next steps along the road to more accurate protein design thus include the development of methods for the construction and validation of useful nonnative ensembles, and the integration of structure refinement techniques with multi-state design methods. The validation provided here for our multi-state design, library design, and high-throughput stability screening methods represents a significant step towards the development of future methods that live up to the initial promise of computational protein design.

Materials and methods

Input structural data

Input atomic coordinates for the β1 domain of Streptococcal protein G (Gβ1) were taken from the 2.2 Å crystal structure 1pga,[51] the 60-member NMR structural ensemble 1gb1, and a constrained, minimized average structure generated from the ensemble 2gb1.[52] Hydrogens (if any) were stripped from each structure, and new hydrogen positions were optimized along with side-chain amide and imidazolium group flips using REDUCE.[53] Each structure was then standardized with 50 steps of conjugate gradient minimization using the DREIDING force field.[54] An unconstrained 128-member molecular dynamics (MD) ensemble was generated from the minimized crystal structure by running a 12.8 ps MD trajectory at 300 K using the DREIDING force field and saving the coordinates every 0.1 ps. The constrained MD trajectory was generated by the same procedure, using an additional harmonic point restraint with a force constant of 100 kcal/mol/Å^2 applied to keep C_α atoms near their initial positions. Each MD snapshot was standardized as described above. After standardization, the NMR, constrained MD, and unconstrained MD ensembles exhibited average pairwise main-chain RMSDs of 0.25, 0.12, and 0.84 Å, respectively.

Sequence Design Specifications and Energy Calculations

In the sequence designs, ten core positions of Gβ1 (3, 5, 7, 20, 26, 30, 34, 39, 52, and 54), were allowed to assume any of the hydrophobic amino acids A, V, L, I, F, Y,

and W. Tryptophan 43 was allowed to change conformation but not amino acid type, so that our fluorescence-based stability assay would not be compromised. Allowed side-chain conformations at the variable positions were taken from the Dunbrack backbone-dependent rotamer library with expansions of ±1 standard deviation around $\chi 1$ and $\chi 2$.[17] To avoid bias toward the wild-type sequence, this set was not supplemented with the side-chain coordinates from the input structure, except at position 43. All other side chains and the main chain were fixed in the input conformation. Pairwise energies were computed for each structure or ensemble member using energy functions described previously,[55, 56] with the polar hydrogen burial term omitted.

Sequence optimization

FASTER was used to find optimized sequences in the single-state design of the crystal structure and the NMR constrained minimized average.[57] Multi-state sequence optimization of the NMR, unconstrained MD, and constrained MD ensembles was performed using a method similar to several that have been described.[23, 25] These methods implement a combinatorial search through amino acid sequence space in which sequences are scored by performing rotamer optimization in the context of each state and these energies are combined to yield a single ensemble score. Our implementation uses FASTER for both the search through amino acid sequence space and for the rotamer optimization on each state (Chapter 3). Here, the energies of a sequence in the context of several states were combined into a single score by computing the free energy of the ensemble system at 300 K:

$$A = -kT\log(\sum_j e^{-E_j/kT})$$

where each E_j is the energy of the sequence when threaded on member j of the ensemble.

Combinatorial library design

To choose combinatorial sequence libraries for experimental screening, we used a new algorithm reported here (see supplementary material). Given a list of scored sequences, a list of allowed sets of amino acids, and a range of desired library sizes, the method evaluates all possible combinations of sets of amino acids at different positions that lead to a library with a size in the desired range. Each position in each library is scored by summing the Boltzmann weights of the sequences in the list that contain a library-specified amino acid at that position. The position scores are then summed to give an overall library score. Our algorithm is able to consider all possible libraries because it treats positions independently, and because it ignores amino acid sets that are unnecessarily large in the context of a given position. In this work, a temperature of 300 K was used in the Boltzmann weighting, and the target library size was 24. We allowed only those sets of amino acids that can be specified by degenerate codons that do not include codons observed with low frequency in *E. coli*.

After applying this algorithm to the lists of sequences produced by the computational designs, we instantiated the 20 best-scoring libraries from each design and rescored all of the amino acid sequences in each library by rotamer optimization. Each library we inspected contained the best-scoring sequence from the design it was based on, although this is not required by our method. From each design, we chose for

experimental testing the library in the top 20 with the smallest energy spread between its best-scoring and worst-scoring sequence.

Library construction, expression, and purification

Oligonucleotides (desalted, Integrated DNA Technologies) ranging from 45 to 60 bp containing ~ 18 bp overlapping segments were assembled via a modified Stemmer method[37] using KOD Hot Start Polymerase (Novagen) to generate full-length streptococcal Gβ1 with an N terminal His_6 tag. Secondary structure content and annealing temperatures were verified by NUPACK.[58, 59] The following procedure was repeated for each library constructed. Oligonucleotides containing the desired single mutation or degenerate codon were swapped into the assembly mixture to generate the diversity of each library. If a degenerate codon could not account for the desired residue diversity, equimolar ratios of applicable single mutation oligonucleotides were added to the assembly mixture. Standard subcloning techniques were performed to insert the library into a frameshift selection plasmid (pInSALect),[27] and after miniprepping the selected harvested colonies, the library was inserted into an expression plasmid (pET11a). The library was transformed into BL21 Gold DE3 cells (Stratagene) by heat shock and colonies were picked into 96-well plates for plasmid miniprepping and sequencing (Agencourt Biosciences). Missing library members were generated by standard quick-change protocols. Sequence-verified library members were pulled from replicated glycerol stocks and inoculated into 5 mL of Instant TB media (Novagen) in 24-well plates. After overnight incubation at 37ºC, cells were pelleted by centrifugation at 5,000 x *g* for 20 min. Pellets were freeze/thawed once and resuspended in lysis buffer

(50 mM $NaPO_4$, 300 mM NaCl, 1x CelLytic B (Sigma-Aldrich), 2.5 mM imidazole, pH 8) before another identical centrifugation step. Cell lysates were loaded onto an equilibrated HIS-Select filter plate (Sigma-Aldrich), washed twice and eluted with buffer containing 250 mM imidazole, pH 8.

Microtiter plate-based stability determination

Appropriate amounts of 8 M GdmCl (Sigma-Aldrich), Milli-Q water, eluted protein, and 50 mM $NaPO_4$ buffer, pH 6.5, were added to maintain a fixed volume in each well of 96-well Costar UV transparent flat bottom plates by a Freedom EVO liquid handling robot (Tecan). Mutant proteins were subjected to a 12-point GdmCl gradient across the columns of the plate where each row contained a separate denaturation experiment. Only twenty-seven 96-well plates were needed for all libraries, including duplicates. The plates were equilibrated for at least one hour and shaken at 900 rpm on a microtiter plate shaker (Heidolph).

Tryptophan fluorescence measurements were taken on a fluorescence plate reader (Tecan) with a plate stacker attachment. Ideal parameters were empirically determined for wild-type Gβ1 and later used for every library assayed. Excitation was performed at 295 nm and emission measured at 341 nm with 10 nm bandwidths. Data were fit as a two-state unfolding transition using the linear extrapolation method[60] in Pylab. The GdmCl concentration at the midpoint of denaturation, C_m, was estimated numerically based on the fraction-unfolded curve fit.

Supplementary information

Combinatorial library design

Structure-based computational protein design (CPD) methods can be harnessed to expedite the engineering of proteins by directed evolution. Several methods have been developed to allow the design of combinatorial mutation libraries to be informed by the results of CPD calculations (Figure 2). These approaches allow many specific variants chosen by CPD to be tested experimentally, and can facilitate assessment and improvement of the design procedure. Hayes et al. described a method in which a list of low-energy sequences found by CPD is used to generate a table of frequencies for each amino acid type at each position, and then a frequency cutoff is applied to limit the library to only those amino acids found more frequently than the cutoff value at each position.[33] Mena and Daugherty developed a similar procedure that produces libraries that include as many of the sequences in the CPD list as possible, while using only those sets of amino acids that can be encoded using degenerate codons.[35] This feature helps to ensure that the resulting combinatorial gene libraries can be synthesized quickly and inexpensively. Treynor et al. developed a computational library design method analogous to CPD in which interactions between sets of amino acids at various positions are scored, and this system of interactions is sampled using standard CPD optimization algorithms to find the most favorable degenerate codon sequence.[36]

In our view, a procedure that couples CPD to the design of combinatorial protein libraries should provide at least the following:

1. **Explicit consideration of CPD energies**. Methods that ignore CPD energies lead to a weaker correspondence between the final libraries and the original design calculations, limiting the predictive capability of the library design procedure and making improvement of CPD through library screening and analysis more difficult.
2. **Direct specification of the range of library sizes that should be produced**. In general, the desired library size will be a direct function of experimental screening capacity. A method that does not allow the user to specify the library size will either require repeated manual rerunning in an attempt to generate the desired library size, or will waste potentially prohibitive amounts of compute time analyzing libraries with irrelevant sizes.
3. **Control over which sets of amino acids are allowed**. Users with limited resources will usually prefer sets of amino acids that can be encoded using degenerate codons, because the resulting gene libraries can be synthesized in a single reaction with a relatively small number of inexpensive oligonucleotides. Those who can afford larger numbers of oligonucleotides and liquid-handling robots will be able to test libraries made with arbitrary sets of amino acids, which in general should more accurately reflect the sequence preferences of CPD calculations. A robust library design method must therefore handle whatever sets of amino acids the user deems appropriate.
4. **Consideration of all user-allowed sets of amino acids at each position**. Some methods use heuristics to remove from consideration particular sets of amino acids at each position. Although this process can reduce the computational cost of

the library design procedure, it can also result in the elimination of desirable libraries.

Because no previously reported algorithm that we know of satisfies all these criteria, we developed one that does. The new algorithm takes several inputs: (1) a list of scored sequences; (2) a list of allowed sets of amino acids (e.g., those that can be encoded using degenerate codons); (3) a range of preferred library sizes; (4) a simulation temperature that controls the degree of preference for sequences with better scores; and, optionally, (5) sets of amino acids that are to be required or prohibited at particular positions. Based on these inputs, the algorithm produces a list of combinatorial libraries that are ranked according to the degree to which they satisfy the input list of scored sequences.

The process used by the algorithm to produce a list of combinatorial libraries from a list of scored sequences can be conceptually separated into three steps (Figure 9).

Step **A**. Scan through the input list of scored sequences, and generate a "total diversity" library that includes, at each position, every amino acid seen in the list at that position. This library represents the list optimally but ignores the user's preferred library size and allowed sets of amino acids. If later steps indicate that the size of the problem with this total diversity is insurmountably large, the user can request that the total diversity library be constructed from a subset of the input sequence list. For example, given a list of length 10,000, the user might decide to consider only the best 1,000 sequences in the list during this step.

Step **B**. Enumerate all possible amino acid size configurations that lead to combinatorial libraries within the range of sizes specified by the user. A size configuration is simply a specific number of amino acids at each position in the protein (e.g., 3 amino acids at position 1, 4 amino acids at position 2, etc.). An amino acid set size need not be considered at a particular position if it is larger than the smallest set that includes all amino acids found at that position in the total diversity library. This greatly reduces the total number of size configurations that need to be generated in this step and scored in the next step.

Step **C**. For each size configuration, determine the best set of amino acids of the required size at each position. This is done for each position independently by computing a partition function for each amino acid set with the given size. Amino acid sets that lack user-required amino acids or contain user-prohibited amino acids can be skipped here. Given a position and an allowed set of amino acids, iterate through the list of scored sequences, and for each sequence add to a cumulative partition function the Boltzmann-weight, $\exp(-E/kT)$, where E is the score of the sequence, k is the Boltzmann constant, and T is the simulation temperature. If the amino acid at that position in the current sequence is not found in the amino acid set of interest, nothing is added to the partition function. If the simulation temperature is low, the best-scored sequences will contribute most strongly to the partition function; if the temperature is high, all sequences in the list will contribute similarly. At each position, the set of amino acids with the most favorable partition function (position library score) is chosen. This procedure produces an optimal combinatorial library for each size configuration. The optimal libraries of each possible

size configuration can then be ranked based on the sums of their position library scores across all positions.

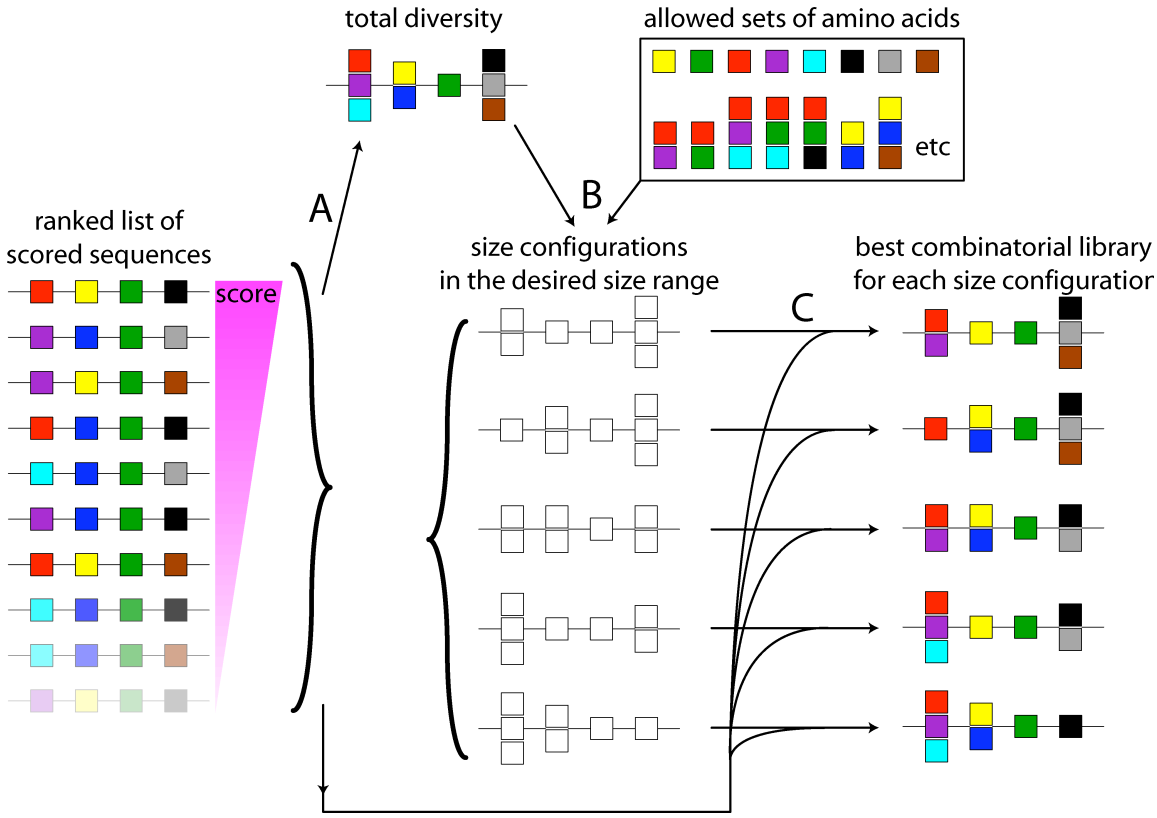

Figure 9: Detail of the library design method. (A) The list of scored sequences defines an initial "total diversity" library that is typically much larger (10^3 – 10^{15}, or even more) than the desired library size (10^2–10^6). (B) This total diversity library and the allowed sets of amino acids are used to construct a set of size configurations that lead to libraries in the desired range of sizes. The boxes in the list of size configurations are unfilled, indicating that the particular amino acids at each position have not yet been determined at this step. (C) For each size configuration generated in the previous step, the original list of scored sequences is used to find the optimal set of amino acids of the required size at each position.

Microtiter plate-based stability assay controls

The fluorescence profiles of the GdmCl gradient and the elution buffer show no effect on the shape of the unfolding transition of wild-type Gβ1 (Figure 10). Sample signal below the elution buffer was interpreted as expression failure; any data that could not be fit yet whose signal was above the elution buffer was deemed expressed but unstable/unfolded (but see discussion above). In order to test the accuracy of the microtiter plate-based denaturation assay, Gβ1 unfolding was monitored by circular dichroism (Aviv Biomedical) and tryptophan fluorescence in a fluorimeter (Photon Technology International). The denaturation profiles from these low-throughput experiments were compared to results from the fluorescence plate reader (Figure 11). The overlapping data points support the use of a two-state unfolding fit during our stability calculations and verify the accuracy of the assay. Next, the unfolding curves from several protein preparations from different concentrations confirmed the assay's precision (Figure 12). These results support some assumptions that the stability determination method described here makes in order to maintain a high level of throughput. First, we never assay for protein concentration before setting up the GdmCl gradient, relying on the fraction-unfolded plot to remove any concentration bias/effects. Second, the high concentration (250 mM) of imidazole in elution buffer is never dialyzed out of the eluted protein solution. Figures 11 and 12 show that these discrepancies in protein preparation have no significant effect on fraction unfolded plots for the wild-type protein.

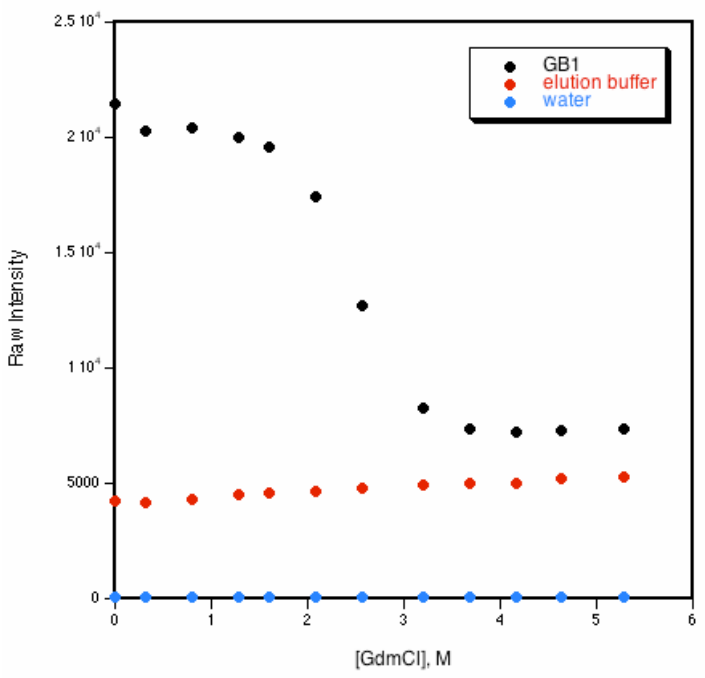

Figure 10: Denaturation gradient and elution buffer fluorescence profiles. Gβ1 (black) was expressed in a 5 mL culture, purified, and eluted with 500 μL of elution buffer (50 μM $NaPO_4$, 300 mM NaCl, 250 mM imidazole, pH 8). Since each point of the Gβ1 denaturation profile contains 35 μL of eluted protein, the elution buffer profile (red) substitutes protein with 35 μL of elution buffer. Similarly, the water profile (blue) adds 35 μL of water to make up the final volume. Each denaturation profile contains an increasing gradient of GdmCl, 50 μM $NaPO_4$ buffer at pH 6.5, and water.

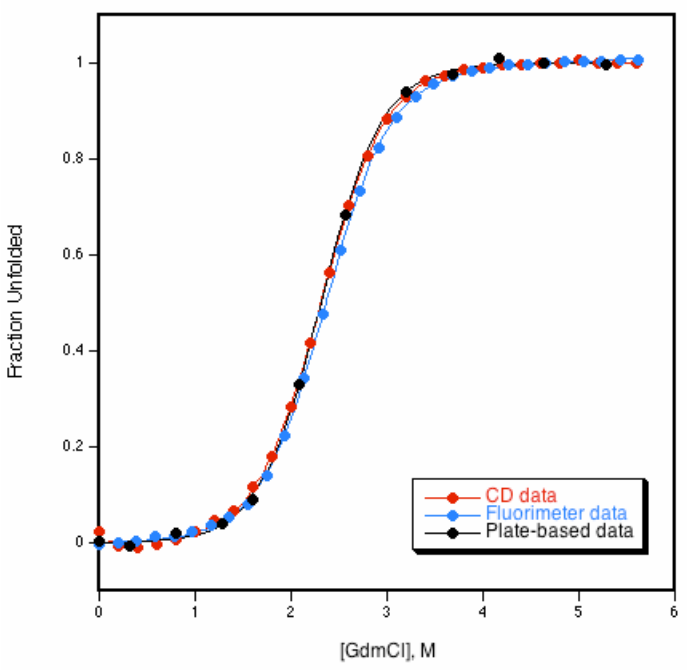

Figure 11: Fraction-unfolded profiles between different modes of detection. CD data (red) measured 5 μM Gβ1 titrated with a 5 μM Gβ1/8 M GdmCl solution in 0.2 M steps at 218 nm. Fluorimeter data (blue) measured 5μM Gβ1 titrated as in the CD experiment with excitation performed at 295 nm and emission recorded at 341 nm with 4 nm bandwidths. Plate-based data (black) measured 12 separate solutions of 10 μM Gβ1 in response to increasing amounts of 8 M GdmCl with fluorescence parameters identical to the fluorimeter data except for 10 nm bandwidths. All samples were measured at 25°C in 50 μM $NaPO_4$ buffer at pH 6.5.

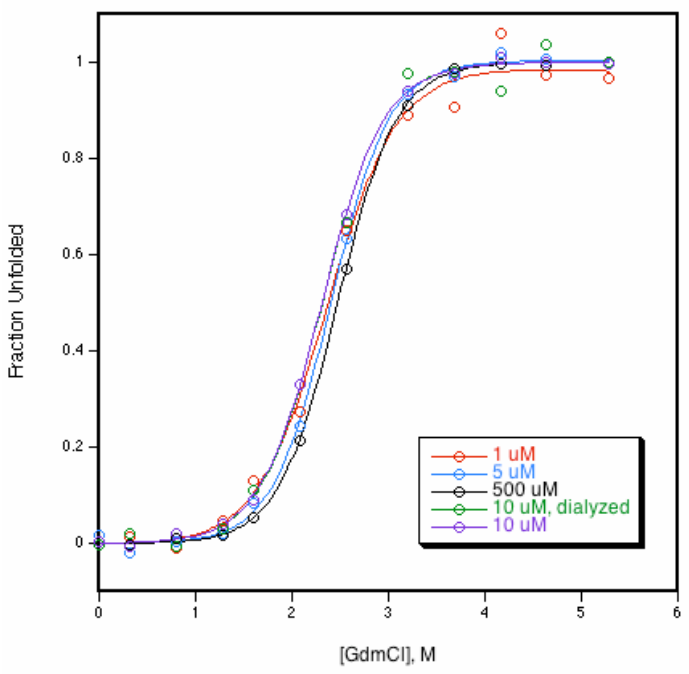

Figure 12: Fraction-unfolded profiles between different protein preparations. Gβ1 was expressed in 100 mL cultures, purified and diluted to 1, 5, 10, and 500 μM in 50 μM $NaPO_4$ buffer at pH 6.5. Another expression culture was dialyzed overnight (Pierce Biotechnology) after purification and diluted to 10 μM in the same buffer. All measurements were taken on a fluorescence plate reader as described in the text.

References

1. Arnold, F. H., Combinatorial and computational challenges for biocatalyst design. *Nature* **2001,** *409* (6817), 253–257.

2. Jackel, C.; Kast, P.; Hilvert, D., Protein design by directed evolution. *Annual Reviews of Biophysics* **2008,** *37*, 153–173.

3. Bershtein, S.; Tawfik, D. S., Advances in laboratory evolution of enzymes. *Current Opinion in Chemical Biology* **2008,** *12* (2), 151–158.

4. Alvizo, O.; Allen, B. D.; Mayo, S. L., Computational protein design promises to revolutionize protein engineering. *Biotechniques* **2007,** *42* (1), 31–35.

5. Lippow, S. M.; Tidor, B., Progress in computational protein design. *Current Opinion in Biotechnology* **2007,** *18* (4), 305–311.

6. Schueler-Furman, O.; Wang, C.; Bradley, P.; Misura, K.; Baker, D., Progress in modeling of protein structures and interactions. *Science* **2005,** *310* (5748), 638–642.

7. Bolon, D. N.; Mayo, S. L., Enzyme-like proteins by computational design. *Proceedings of the National Academy of Sciences of the United States of America* **2001,** *98* (25), 14274–14279.

8. Dahiyat, B. I.; Mayo, S. L., De novo protein design: Fully automated sequence selection. *Science* **1997,** *278* (5335), 82–87.

9. Dwyer, M. A.; Looger, L. L.; Hellinga, H. W., Computational design of a Zn2+ receptor that controls bacterial gene expression. *Proceedings of the National Academy of Sciences of the United States of America* **2003,** *100* (20), 11255–11260.

10. Jiang, L.; Althoff, E. A.; Clemente, F. R.; Doyle, L.; Rothlisberger, D.; Zanghellini, A.; Gallaher, J. L.; Betker, J. L.; Tanaka, F.; Barbas, C. F.; Hilvert, D.; Houk, K. N.; Stoddard, B. L.; Baker, D., De novo computational design of retro-aldol enzymes. *Science* **2008,** *319* (5868), 1387–1391.

11. Kuhlman, B.; Dantas, G.; Ireton, G. C.; Varani, G.; Stoddard, B. L.; Baker, D., Design of a novel globular protein fold with atomic-level accuracy. *Science* **2003,** *302* (5649), 1364–1368.

12. Looger, L. L.; Dwyer, M. A.; Smith, J. J.; Hellinga, H. W., Computational design of receptor and sensor proteins with novel functions. *Nature* **2003,** *423* (6936), 185–190.

13. Malakauskas, S. M.; Mayo, S. L., Design, structure and stability of a hyperthermophilic protein variant. *Nature Structural Biology* **1998,** *5* (6), 470–475.

14. Rothlisberger, D.; Khersonsky, O.; Wollacott, A. M.; Jiang, L.; DeChancie, J.; Betker, J.; Gallaher, J. L.; Althoff, E. A.; Zanghellini, A.; Dym, O.; Albeck, S.; Houk, K.

N.; Tawfik, D. S.; Baker, D., Kemp elimination catalysts by computational enzyme design. *Nature* **2008,** *453* (7192), 190–U4.

15. Chica, R. A.; Doucet, N.; Pelletier, J. N., Semi-rational approaches to engineering enzyme activity: combining the benefits of directed evolution and rational design. *Current Opinion in Biotechnology* **2005,** *16* (4), 378–384.

16. Shortle, D., The denatured state (the other half of the folding equation) and its role in protein stability. *FASEB Journal* **1996,** *10* (1), 27–34.

17. Dunbrack, R. L.; Cohen, F. E., Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science* **1997,** *6* (8), 1661–1681.

18. Lassila, J. K.; Privett, H. K.; Allen, B. D.; Mayo, S. L., Combinatorial methods for small-molecule placement in computational enzyme design. *Proceedings of the National Academy of Sciences of the United States of America* **2006,** *103* (45), 16710–16715.

19. Dahiyat, B. I.; Mayo, S. L., Probing the role of packing specificity in protein design. *Proceedings of the National Academy of Sciences of the United States of America* **1997,** *94* (19), 10172–10177.

20. Grigoryan, G.; Ochoa, A.; Keating, A. E., Computing van der Waals energies in the context of the rotamer approximation. *Proteins* **2007,** *68* (4), 863–878.

21. Hu, X.; Wang, H.; Ke, H.; Kuhlman, B., High-resolution design of a protein loop. *Proceedings of the National Academy of Sciences of the United States of America* **2007,** *104* (45), 17668–17673.

22. Pokala, N.; Handel, T. M., Energy functions for protein design: adjustment with protein-protein complex affinities, models for the unfolded state, and negative design of solubility and specificity. *Journal of Molecular Biology* **2005,** *347* (1), 203–227.

23. Ambroggio, X. I.; Kuhlman, B., Computational design of a single amino acid sequence that can switch between two distinct protein folds. *Journal of the American Chemical Society* **2006,** *128* (4), 1154–1161.

24. Boas, F. E.; Harbury, P. B., Design of protein-ligand binding based on the molecular-mechanics energy model. *Journal of Molecular Biology* **2008,** *380* (2), 415–424.

25. Havranek, J. J.; Harbury, P. B., Automated design of specificity in molecular recognition. *Nature Structural Biology* **2003,** *10* (1), 45–52.

26. Dyson, H. J.; Wright, P. E., Intrinsically unstructured proteins and their functions. *Nature Reviews of Molecular and Cell Biology* **2005,** *6* (3), 197–208.

27. Gerth, M. L.; Patrick, W. M.; Lutz, S., A second-generation system for unbiased reading frame selection. *Protein Engineering Design & Selection* **2004,** *17* (7), 595–602.

28. Cox, J. C.; Lape, J.; Sayed, M. A.; Hellinga, H. W., Protein fabrication automation. *Protein Science* **2007,** *16* (3), 379–390.

29. Aucamp, J. P.; Cosme, A. M.; Lye, G. J.; Dalby, P. A., High-throughput measurement of protein stability in microtiter plates. *Biotechnology and Bioengineering* **2005,** *89* (5), 599–607.

30. Voigt, C. A.; Mayo, S. L.; Arnold, F. H.; Wang, Z. G., Computational method to reduce the search space for directed protein evolution. *Proceedings of the National Academy of Sciences of the United States of America* **2001,** *98* (7), 3778–3783.

31. Endelman, J. B.; Silberg, J. J.; Wang, Z. G.; Arnold, F. H., Site-directed protein recombination as a shortest-path problem. *Protein Engineering Design & Selection* **2004,** *17* (7), 589–594.

32. Voigt, C. A.; Martinez, C.; Wang, Z. G.; Mayo, S. L.; Arnold, F. H., Protein building blocks preserved by recombination. *Nature Structural Biology* **2002,** *9* (7), 553–558.

33. Hayes, R. J.; Bentzien, J.; Ary, M. L.; Hwang, M. Y.; Jacinto, J. M.; Vielmetter, J.; Kundu, A.; Dahiyat, B. I., Combining computational and experimental screening for rapid optimization of protein properties. *Proceedings of the National Academy of Sciences of the United States of America* **2002,** *99* (25), 15926–15931.

34. Kono, H.; Saven, J. G., Statistical theory for protein combinatorial libraries. Packing interactions, backbone flexibility, and the sequence variability of a main-chain structure. *Journal of Molecular Biology* **2001,** *306* (3), 607–628.

35. Mena, M. A.; Daugherty, P. S., Automated design of degenerate codon libraries. *Protein Engineering Design & Selection* **2005,** *18* (12), 559–561.

36. Treynor, T. P.; Vizcarra, C. L.; Nedelcu, D.; Mayo, S. L., Computationally designed libraries of fluorescent proteins evaluated by preservation and diversity of function. *Proceedings of the National Academy of Sciences of the United States of America* **2007,** *104* (1), 48–53.

37. Stemmer, W. P.; Crameri, A.; Ha, K. D.; Brennan, T. M.; Heyneker, H. L., Single-step assembly of a gene and entire plasmid from large numbers of oligodeoxyribonucleotides. *Gene* **1995,** *164* (1), 49–53.

38. Kirsten Frank, M.; Dyda, F.; Dobrodumov, A.; Gronenborn, A. M., Core mutations switch monomeric protein GB1 into an intertwined tetramer. *Nature Structural Biology* **2002,** *9* (11), 877–885.

39. Byeon, I. J.; Louis, J. M.; Gronenborn, A. M., A protein contortionist: core mutations of GB1 that induce dimerization and domain swapping. *Journal of Molecular Biology* **2003,** *333* (1), 141–152.

40. Jee, J.; Byeon, I. J.; Louis, J. M.; Gronenborn, A. M., The point mutation A34F causes dimerization of GB1. *Proteins* **2008,** *71* (3), 1420–1431.

41. Gronenborn, A. M.; Frank, M. K.; Clore, G. M., Core mutants of the immunoglobulin binding domain of streptococcal protein G: stability and structural integrity. *FEBS Letters* **1996,** *398* (2–3), 312–316.

42. Su, A.; Mayo, S. L., Coupling backbone flexibility and amino acid sequence selection in protein design. *Protein Science* **1997,** *6* (8), 1701–1707.

43. Mendes, J.; Guerois, R.; Serrano, L., Energy estimation in protein design. *Current Opinion in Structural Biology* **2002,** *12* (4), 441–446.

44. Guerois, R.; Nielsen, J. E.; Serrano, L., Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *Journal of Molecular Biology* **2002,** *320* (2), 369–387.

45. Zollars, E. S. Force Field Development In Protein Design. Caltech, Pasadena, CA, 2006.

46. Fajardo-Sanchez, E.; Stricher, F.; Paques, F.; Isalan, M.; Serrano, L., Computer design of obligate heterodimer meganucleases allows efficient cutting of custom DNA sequences. *Nucleic Acids Research* **2008,** *36* (7), 2163–2173.

47. Tur, V.; van der Sloot, A. M.; Reis, C. R.; Szegezdi, E.; Cool, R. H.; Samali, A.; Serrano, L.; Quax, W. J., DR4-selective tumor necrosis factor-related apoptosis-inducing ligand (TRAIL) variants obtained by structure-based design. *Journal of Biological Chemistry* **2008,** *283* (29), 20560–20568.

48. van der Sloot, A. M.; Mullally, M. M.; Fernandez–Ballester, G.; Serrano, L.; Quax, W. J., Stabilization of TRAIL, an all-beta-sheet multimeric protein, using computational redesign. *Protein Engineering Design & Selection* **2004,** *17* (9), 673–680.

49. van der Sloot, A. M.; Tur, V.; Szegezdi, E.; Mullally, M. M.; Cool, R. H.; Samali, A.; Serrano, L.; Quax, W. J., Designed tumor necrosis factor-related apoptosis-inducing ligand variants initiating apoptosis exclusively via the DR5 receptor. *Proceedings of the National Academy of Sciences of the United States of America* **2006,** *103* (23), 8634–8639.

50. Szczepek, M.; Brondani, V.; Buchel, J.; Serrano, L.; Segal, D. J.; Cathomen, T., Structure-based redesign of the dimerization interface reduces the toxicity of zinc-finger nucleases. *Nature Biotechnology* **2007,** *25* (7), 786–793.

51. Gallagher, T.; Alexander, P.; Bryan, P.; Gilliland, G. L., 2 Crystal-Structures of the B1 Immunoglobulin-Binding Domain of Streptococcal Protein-G and Comparison with Nmr. *Biochemistry* **1994,** *33* (15), 4721–4729.

52. Gronenborn, A. M.; Filpula, D. R.; Essig, N. Z.; Achari, A.; Whitlow, M.; Wingfield, P. T.; Clore, G. M., A novel, highly stable fold of the immunoglobulin binding domain of streptococcal protein G. *Science* **1991,** *253* (5020), 657–661.

53. Word, J. M.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C., Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *Journal of Molecular Biology* **1999,** *285* (4), 1735–1747.

54. Mayo, S. L.; Olafson, B. D.; Goddard, W. A., Dreiding — a Generic Force-Field for Molecular Simulations. *Journal of Physical Chemistry* **1990,** *94* (26), 8897–8909.

55. Gordon, D. B.; Hom, G. K.; Mayo, S. L.; Pierce, N. A., Exact rotamer optimization for protein design. *Journal of Computational Chemistry* **2003,** *24* (2), 232–243.

56. Gordon, D. B.; Marshall, S. A.; Mayo, S. L., Energy functions for protein design. *Current Opinion in Structural Biology* **1999,** *9* (4), 509–513.

57. Allen, B. D.; Mayo, S. L., Dramatic performance enhancements for the FASTER optimization algorithm. *Journal of Computational Chemistry* **2006,** *27* (10), 1071–1075.

58. Dirks, R. M.; Pierce, N. A., A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of Computational Chemistry* **2003,** *24* (13), 1664–1677.

59. Dirks, R. M.; Pierce, N. A., An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots. *Journal of Computational Chemistry* **2004,** *25* (10), 1295–1304.

60. Santoro, M. M.; Bolen, D. W., Unfolding free energy changes determined by the linear extrapolation method. 1. Unfolding of phenylmethanesulfonyl alpha-chymotrypsin using different denaturants. *Biochemistry* **1988,** *27* (21), 8063–8068.

Chapter 5

The Importance of Combinatorial Optimization in the Improvement of Models for Computational Protein Design

Optimization in computational protein design

Initial progress in the field of computational protein design (CPD) was accelerated by the development of mathematically rigorous optimization methods based on the dead-end elimination (DEE) theorem. The availability of these methods helped to instill confidence that provably optimal solutions could be found for astronomically combinatorial protein design problems based on the inverse-folding model. Although the utility of such methods was demonstrated by several successful designs, and many clever improvements were made to extend their applicability, their poor performance scaling soon began to limit the progress of CPD. Reliance on DEE-based optimization was especially problematic when applied in the context of more accurate sampling of side-chain conformational flexibility, the design of many positions simultaneously, or the modeling of substrates and enzymatic transition states.

In response to the limitations of DEE, stochastic optimization routines were developed based on Monte Carlo with simulated annealing (MC), FASTER, and genetic algorithms (GA). Although these methods do not guarantee the generation of optimal solutions, they can be run as long as desired to improve the quality of the solutions, and they always return a solution, regardless of the difficulty of the problem. In practice, we have found that, in contrast to the other stochastic methods, the improved FASTER procedure detailed in Chapter 2 is always able to find the DEE-derived solution when DEE can converge, and is able to converge to a single low-energy solution even for significantly more difficult problems.

Our experiences with the various types of exact and stochastic optimization techniques used in CPD strongly suggest that sampling of configuration space is not the limiting component in the application of single-state, inverse-folding models to real-world protein design problems. Even for the largest inverse-folding problems for which all possible pairwise energies between rotamers can be precomputed and stored in memory, the improved FASTER algorithm can converge to low-energy solutions that are believed (though not proven) to be optimal.

In contrast, the recent development of multi-state design (MSD) procedures has provided more fertile ground for the improvement and testing of optimization routines. MSD procedures must perform individual rotamer-optimization calculations to assess the fitness of each sequence analyzed, and therefore orders of magnitude fewer distinct sequences can be evaluated per unit time. Because scoring a sequence in MSD is so costly, efficient optimization algorithms for MSD must choose sequences to test much more carefully than would be required in single-state design (SSD) problems of equivalent combinatorial size. In Chapter 3, we saw that our implementation of MSD-FASTER significantly outperforms MSD-MC in all cases tested, often finding solutions better than the best ever found by MSD-MC. These results highlight the idea that, unlike SSD problems with precomputed pairwise energies, MSD problems can easily exceed the capabilities of existing sampling algorithms. Thus, more efficient optimization routines are expected to help generate more useful protein variants and accelerate the improvement of CPD models based on MSD.

Design protocols that compute energies on the fly have been investigated to a far lesser extent than those that rely on precomputed energy matrices. So far, the greater

computational expense of on-the-fly methods has precluded their use, despite the CPD model improvements their use enables. For example, on-the-fly methods are amenable to energy functions that cannot be expressed as sums of pairwise energies between positions, such as solvation functions that rely on exact descriptions of complete molecular surfaces. Furthermore, unlike precomputed energy methods, on-the-fly methods need not be limited to rigid main-chain structures. In on-the-fly design methodology, structure refinement and minimization moves can be applied concurrently with rotamer and amino acid changes, potentially facilitating the discovery during the design process of more appropriate scaffold conformations for evaluating the sequences of interest.

This strategy might be most useful in the context of MSD. A database of main-chain structures could be used to score individual sequences, and these structures could be refined during sequence optimization to better represent the sequences found over the course of the design. The database might include both target states and competing states for explicit negative design. Although such methods are expected to improve the predictive ability of CPD calculations, they will also be dramatically more time-consuming than the inverse-folding design calculations to which the field of CPD has become accustomed. These methods will only be rendered tractable by significant advances in computational hardware, as well as the development of conformational sampling algorithms that can handle the combinatorial explosion caused by the treatment of main-chain flexibility.

In Chapter 4, we found that CPD methods can help to predict combinatorial libraries of stable sequences, even when they cannot accurately correlate the experimental

and simulated stabilities of these sequences. Given this result, it seems worthwhile to question the utility of rigorous sampling in CPD calculations. Specifically, if the correlation between simulated and experimental fitness is low, then why bother spending additional time in an attempt to find solutions of better energy?

Characteristics of CPD as a tool for protein engineering

The high-throughput stability assessment of our designed libraries may provide insight into the level of simulation accuracy that might be required for CPD to be usefully applied in protein engineering. It is often postulated that, in order for CPD to display predictive power, it must adequately reproduce stability changes ($\Delta\Delta G$s) of mutation from experimental data sets. However, no correlation was observed between the simulation energies of the individual sequences we assayed and their experimental stabilities. Given this result, we were pleasantly surprised by the ability of our computational library design procedure to produce many well-folded and stabilized sequences based on each type of input structural data. Although it might be assumed that the sequence space of our designs contained an unusually large number of viable sequences, our own data and the reports of others soundly contradict this; we cannot reasonably conclude that the design problem we chose was serendipitously trivial.

So how can a protein design method successfully produce libraries of well-folded, stabilized variants without accurately predicting the relative stabilities of any given pair of mutants? This remarkable property of CPD may arise due to the same fundamental characteristics of proteins that make natural and directed evolution possible.

Although the ability of a protein sequence to fold to a stable and active structure is governed by a precarious balance of energetic contributions with large magnitudes and opposite signs, naturally occurring proteins are nevertheless sufficiently tolerant of substitution to enable the evolution of molecular function through mutagenesis and screening or selection. Starting with an existing functional protein, an area of sequence space enriched with active variants can be explored by iterative cycles of mutation or recombination. This process works because many substitutions can be accommodated by structural adjustments that maintain the general fold, and because the structural accuracy required for activity is not prohibitively high.

Now, we consider CPD methods in light of the biophysical properties of proteins that enable evolution. Inverse-folding design models (including those of the multi-state variety) ultimately score amino acid sequences in the context of one or more fixed scaffold conformations using molecular mechanics and heuristic energy functions. In order to rigorously assess the relative stabilities of any two sequences, a CPD procedure would need to find a representative ensemble of native and nonnative conformations for each sequence, and compute the free energy of each ensemble using a scoring function that accurately treats polar and nonpolar interactions and solvation effects. However, computational tractability requires that only a small subset of the possible conformational space be evaluated, and that approximate scoring functions which neglect explicit water and complex electrostatic effects be used. The finite set of representative structures used for a particular design will always be more appropriate for some sequences than for others. This leads to false positives, in which a sequence appears to stabilize the target ensemble but actually stabilizes alternative conformations more, and false negatives, in

which a sequence appears to destabilize the ensemble although slight adjustments to the ensemble would render it satisfactory. The unpredictability of these cases leads to the observed lack of correlation between simulation energies and experimental measures of fitness.

So, despite insufficient sampling and approximate energy functions, the forgiving nature of protein self-assembly enables CPD to find areas of sequence space likely to be compatible with a given structure and function. As described above, evolution can effectively explore sequence space because stable protein sequences are able to relax structurally and accommodate perturbing mutations. Likewise, CPD procedures are able to locate viable areas of sequence space because a sequence compatible with the simulated ensemble can also usually tolerate the minor relaxations that lead to the physically relevant conformational states that are not modeled. Since the exact nature of these relaxations, and the structures they lead to, cannot be predicted during the simulation, the energy of a sequence threaded on the ensemble does not correlate well with experimental reality. Explicit negative design provides an even greater challenge than positive design, since it demands sequences that *destabilize* an ensemble of competing conformations. Unmodelled structural relaxations are more problematic in competing states than in target states because they can transform an apparently destabilizing interaction into a stabilizing one, rendering a simulation-based fitness assessment *qualitatively* incorrect. Despite these issues, experimental validation of CPD calculations has shown that ensembles sufficiently representative of active states (and competing states, if available) can be used to identify regions of sequence space enriched with folded and functional members.

Although the structural adaptability of a protein native state renders untenable the accurate comparison of arbitrary sequences without prohibitive conformational sampling, it also enables the effective design of proteins under the same set of computational restrictions. Ultimately, we reach the surprising conclusion that accurate scoring of particular arbitrary sequences is neither necessary nor sufficient to find areas of sequence space enriched with functional variants.

In Chapter 4, we discussed how this view of current protein design methods leads to unorthodox proposals for the improvement of CPD. If the utility of CPD is derived primarily from its ability to choose variants that satisfy the native state, as it seems to, then two main avenues of inquiry arise. In the first, structural refinement, larger rotamer libraries, and better energy functions are used to improve the degree to which variants can be ranked based on their compatibility with the native state. However, the general difficulty of finding perfect structures for the evaluation of arbitrary sequences and the extreme sensitivity of molecular mechanics energy functions suggests that additional returns from this effort would diminish quickly; native state modeling is continually pushed to improve its predictive power. On the other hand, simulations of competing states have received scant attention in the context of protein design, and might represent lower-hanging fruit. Of course, the generation of appropriate structural templates for the simulation of competing states will be far from trivial.

The vastness of available conformational space will require redoubled efforts towards efficient sampling and optimization as the major simplifying approximations of CPD begin to be discarded. It seems clear that the development of more accurate design procedures must be driven by the availability of improved optimization methods and

move sets that enable protein sequences to be simulated more realistically. My intent with the projects described here was to push the boundaries of what can be attempted in CPD, to maximize the possibility of transformative breakthroughs derived from this technology. I consider it an honor to have had the opportunity to place my own small piece into this mighty puzzle.

Appendix I

Combinatorial Methods for Small Molecule Placement in Computational Enzyme Design

The text of this appendix was adapted from a manuscript coauthored with J. Kyle Lassila, Heidi K. Privett, and Stephen L. Mayo.

Lassila, J. K.; Privett, H. K.; Allen, B. D.; Mayo, S. L., Combinatorial methods for small-molecule placement in computational enzyme design. *Proceedings of the National Academy of Sciences of the United States of America* **2006,** *103* (45), 16710–16715.

Abstract

The incorporation of small molecule transition state structures into protein design calculations poses special challenges because of the need to represent the added translational, rotational, and conformational freedoms within an already difficult optimization problem. Successful approaches to computational enzyme design have focused on catalytic side-chain contacts to guide placement of small molecules in active sites. We describe a process for modeling small molecules in enzyme design calculations that extends previously described methods, allowing favorable small molecule positions and conformations to be explored simultaneously with sequence optimization. Because all current computational enzyme design methods rely heavily on sampling of possible active site geometries from discrete conformational states, we tested the effects of discretization parameters on calculation results. Rotational and translational step sizes as well as side-chain library types were varied in a series of computational tests designed to identify native-like binding contacts in three natural systems. We find that conformational parameters, especially the type of rotamer library used, significantly affect the ability of design calculations to recover native binding site geometries. We describe the construction and use of a crystallographic conformer library, and find that it more reliably captures active-site geometries than traditional rotamer libraries in the systems tested.

Introduction

As catalysts, enzymes offer advantageous properties including dramatic rate enhancements, complete control over absolute stereochemistry, and nontoxic biodegradation. Yet a fundamental limiting factor in the use of enzymes for chemical synthesis, bioremediation, therapeutics, and other applications is the availability of enzymes with the required activities, specificities, and tolerances to reaction conditions. It is therefore a major goal of computational protein design to be able to reliably create completely new protein catalysts with specific properties on demand.

A catalyst by definition must reduce the energy barrier for formation of the transition state. To design transition-state-stabilizing interactions, computational protein design groups have incorporated transition-state or high-energy intermediate state structures into design calculations. These efforts have yielded experimentally verified new catalytic proteins.[1–3] However, substantial challenges still prevent routine or reliable design of enzymes. One major challenge is in finding energy functions that are fast enough for large calculations but that still provide informative approximations of electrostatic and desolvation effects in the protein environment.[4, 5] This paper focuses on another fundamental challenge, the need to represent the large translational, rotational, and conformational freedoms of a small molecule within already astronomically large sequence design calculations.

Here we define protein design as the selection of amino acid sequences such that the resulting protein occupies a given three-dimensional fold and has desired functional properties. Earlier experiments sought to redesign full protein sequences or confer

increased thermostability,[6, 7] but newer work has successfully introduced other properties including catalytic activity, conformational specificity, ligand affinity, and even novel protein folds.[1–3, 8–10] In these examples, side-chain placement algorithms were used to select from a set of discrete, probable side-chain rotamers using energy functions tuned to produce thermostable proteins. These calculations represent difficult optimization problems[11] and they can also be large—a sample calculation performed on a typical enzyme active site yields more than 10^{65} possible sequence combinations, even when excluding movements of the small molecule.

The computational demands of sequence selection prevent ligand positioning using standard docking procedures, which often approximate or neglect side-chain flexibility.[12] Approaches developed specifically for the purpose of enzyme and binding site design have introduced other schemes to limit the calculation size. Looger et al. used stationary, inflexible ligand poses in a large number of individual protein design calculations and demonstrated experimentally that several of the resulting proteins had high ligand affinity.[9] Lilien *e*t al. reported and experimentally validated an ensemble-based method that allows ligand translation and rotation simultaneously with side-chain optimization but only permits mutation of two or three amino acid positions at a time.[13] Chakrabarti et al. described a method for sequence design that neglects conformational and positional ligand flexibility and has not been experimentally tested.[14, 15]

To design new enzyme active sites, a ligand placement method must be able to select side chains in many positions and must consider rotational, translational, and conformational freedom of the small molecule. Previously, methods for the design of catalytic proteins treated high-energy-state structures of the reacting molecules as

extensions of contacting amino acid side-chain rotamers. A two-step procedure was utilized, where ligands, anchoring side chains, and other catalytic side chains were placed through a geometric screening procedure and surrounding side chains were designed in a second step.[1, 16–18] We have developed a process for ligand placement in computational protein design calculations that expands upon previous work and that allows ligand rotation, translation, and conformational freedom to be explored combinatorially within the sequence design calculation itself. The implementation of ligand placement procedures within the context of the pairwise-decomposable protein design framework makes it possible to use a single energy function that can be parameterized as needed to reproduce experimental data.

We tested both a simple rotational and translational process for ligand placement as well as the previously used targeted ligand placement approach. A contact-based screening method is described that allows selection of ligand positions and confomations compatible with catalytic contacts. Test calculations in three systems, *E. coli* chorismate mutase, *S. cerevisiae* triosephosphate isomerase, and *S. avidinii* streptavidin, suggest that the success of ligand placement procedures can be quite sensitive to conformational sampling parameters, including rotational and translational step sizes and the types of rotamer libraries used. We evaluated the efficacy of two standard rotamer libraries and two crystallographic conformer libraries. Traditional rotamers are constructed from canonical χ angles determined by statistical analysis of the PDB,[19–21] whereas conformers have Cartesian coordinates taken directly from high-resolution structures.[22, 23] Conformer libraries may allow more accurate modeling because they are not limited to ideal geometries and their sizes can be tuned more easily and naturally.[22, 23] In our tests, a

backbone-independent conformer library recovered wild-type-like active site geometries more successfully than the other libraries, despite smaller size.

Results and discussion

We have implemented and tested a process for incorporation of small molecules into computational protein design calculations. The procedure is general and may be used to place ground-state ligands or transition-state structures. It is also amenable to multi-state design methods that seek to explicitly reflect the energy difference between reactant and transition states or between alternative ligands.

General calculation procedure

Each ligand placement calculation comprised five steps. In the first step, a large number of discrete variations of ligand coordinates was created. Initial sets of orientations were created by one of two methods, either simple rotation and translation or a targeted placement approach, both of which are discussed in more detail in subsequent sections. In the tests described here, each set of ligand variations contained 10^6–10^9 members, reflecting rotational and translational movement as well as internal conformational flexibility.

Next, the large number of substrate orientations was reduced to a manageable number (< ~ 20,000) using both a simple hard-sphere steric potential to check for backbone clashes and a set of user-defined geometric criteria for side-chain/ligand contacts. In this work, geometric criteria were defined to reflect the distances, angles, and torsions characteristic of important catalytic contacts observed in the crystal

structures (Figure 1). In designing an enzyme with no naturally existing precedent, ideal contact geometries would be based on chemical intuition and/or quantum mechanical calculations. The geometric criteria were applied as follows. For every ligand variation, each of the geometric criteria was tested for satisfaction by contacts from any possible amino acid side-chain conformation in all designed protein positions. If a ligand variation was not able to make at least one of each type of user-specified contact, that ligand variation was discarded from the set. After geometric and steric pruning, the ligand variations remaining were only those theoretically capable of making each of the user-specified contacts.

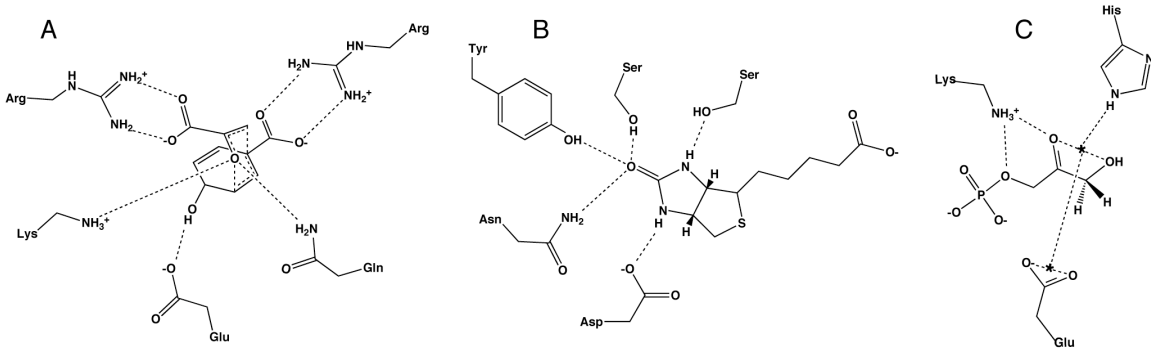

Figure 1: Contact geometries specified in small-molecule pruning step. Ranges of distances, angles, and torsions were allowed that included the crystallographic geometries. (*A*) Chorismate mutase. (*B*) Biotin in streptavidin. (*C*) Triosephosphate isomerase Michaelis complex. Asterisks indicate pseudoatoms used in geometry definitions.

In the third step, pairwise energies for all side-chain/side-chain, side-chain/backbone, backbone/ligand, and side-chain/ligand interactions were calculated using the full force field. In our work, this normally includes a scaled van der Waals[1] term,[24] hydrogen-bonding and electrostatic terms,[25] and a solvation potential.[26, 27]

The fourth step is an optional energy biasing that favors side-chain/ligand contacts deemed necessary for catalysis or binding. This energy biasing step helps to overcome the shortcomings of molecular mechanics energy functions as well as the inherent limitation of treating a multi-state design problem—*differential* stabilization of transition state relative to substrate in protein versus solution—using single-state design algorithms. As methods for modeling electrostatics and solvation and for designing over multiple states improve, the need for this biasing step should be reduced. Previous work utilized selective application of solvation energy or an additional search algorithm step[9] for the same purpose. We favor the use of adjustable bias energies that can be tailored for specific purposes and investigated as a design variable.

To implement the bias, user-specified energies were added or subtracted from pairwise side-chain/ligand interaction energies. We use the energy bias under two regimes, one for normal design calculations and another for rapid assessment of catalytic residue arrangements within a protein scaffold. In normal design calculations, a small energy benefit is simply applied to favor specified types of side-chain/ligand contacts. Alternatively, to quickly identify potential catalytic residues, exaggerated energetic benefits and penalties are applied together. A very large energy benefit is given for desired types of pairwise interactions (100 kcal/mol was used in the test cases reported here). An even larger energy penalty (10,000 kcal/mol here) is applied to all other pairwise side-chain/ligand interactions, except when the side chain is alanine or glycine. In other words, the energy penalty forces all designed side chains to alanine or glycine unless they participate in user-specified catalytic contacts with the ligand. Although this process clearly does not yield physically relevant energetics, it offers a useful tool to

investigate the catalytic conformational space within a binding pocket. The tests performed here to study the effect of sampling parameters on calculation results took advantage of this second approach. Calculations performed to demonstrate sequence selection utilized the normal design approach of applying a simple energy benefit to catalytic contacts.

Finally, in the fifth step, optimal sequences were identified using the FASTER[28, 29] or HERO[30] search methods. In the test cases described here, the result reported is the lowest-energy sequence with the maximal number of specified contacts.

Rotation-translation search

Simple rotation and translation can be used to fill the active site with an initial set of ligand variations in the first step of the process described. Because discrete steps must be used to rotate and translate the ligand, we evaluated the sensitivity of the calculation results to rotational and translational step sizes. A series of calculations was performed using an alanine-containing active-site background, as discussed in step 4 above. We first tested different rotational step sizes using the crystallographic translational starting position with three initial random rotations. Backbone-dependent and backbone-independent rotamer and conformer libraries were tested. Each side-chain library was tested with and without inclusion of the specific crystallographic side-chain rotamers from the structure under examination.

As seen in Table 1, the results of these calculations (in terms of both RMSD relative to crystallographic position and number of wild-type contacts) were strongly dependent on the both the rotational step size and the rotamer library used. In the case of

chorismate mutase, only the backbone-independent conformer library was able to find native-like geometry and contacts. Figure 2 shows results from this library with the 5° step size. When the crystallographic rotamers were included in the calculation, however, all four libraries returned native-like results. It should be noted that none of the three test case structures were included in the set of structures used to create the conformer libraries. The backbone-independent conformer library appeared the most consistently successful with the other two test cases as well, although it showed strong dependence on rotational step size in streptavidin.

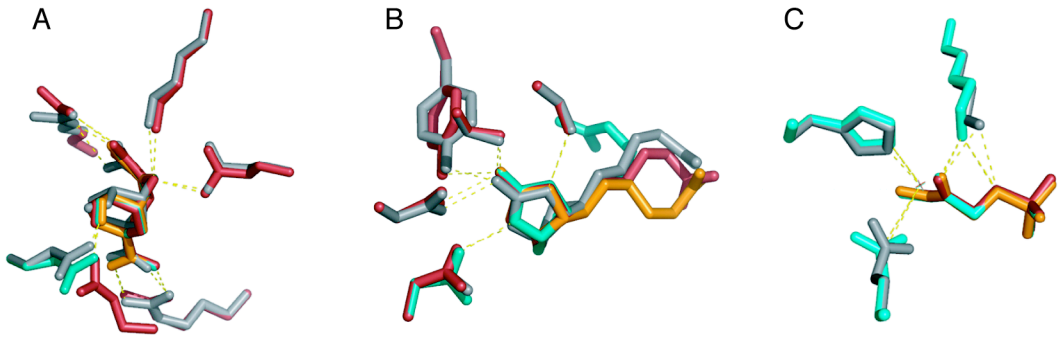

Figure 2: Sample results from test calculations presented in Table 1. Crystallographic side chains and ligands are shown in gray. Results from three trials using different initial random rotational positions are shown in red, teal, and orange. In cases where three colors are not visible, the selected rotamers from two or more calculations were identical. Results are shown from calculations with 5° rotation and the backbone-independent conformer library. (*A*) Chorismate mutase. An alternate backbone position was chosen for a glutamate-hydroxyl contact in one trial (red side chain, lower left). (*B*) Biotin in streptavidin. Note that the biotin pentanoic acid moiety samples different conformations in the calculation and the surrounding side chains were not designed. (*C*) Triosephosphate isomerase.

Table 1: RMSD and number of wild-type contacts as a function of rotational step size and rotamer library[a,b]

Chorismate mutase

	Rotational step size				
Rotamer Library[c]	30°	20°	15°	10°	5°
Conformer: bb-ind	-	-	0.61 ± 0.03 (4.0)	0.55 ± 0.05 (4.0)	0.47 ± 0.04 (4.7)
with xtal rotamers	-	-	0.61 ± 0.03 (4.0)	0.55 ± 0.05 (4.0)	0.47 ± 0.04 (4.7)
Rotamer: bb-ind	-	-	3.88 ± 0.37 (0.0)	2.88 ± 1.44 (0.0)	3.01 ± 1.61 (0.0)
with xtal rotamers	-	-	1.57 ± 1.70 (2.7)	0.51 ± 0.00 (4.0)	0.52 ± 0.01 (4.0)
Conformer: bb-dep	-	-	3.66 ± 0.11 (1.0)	3.59 ± 0.08 (1.0)	3.60 ± 0.09 (1.0)
with xtal rotamers	-	1.67 ± 1.78 (3.3)	1.57 ± 1.83 (3.7)	0.60 ± 0.08 (4.3)	0.54 ± 0.06 (5.0)
Rotamer: bb-dep	-	-	-	-	-
with xtal rotamers	-	-	-	0.49 ± 0.04 (4.3)	0.52 ± 0.01 (4.0)

Streptavidin-Biotin

	Rotational step size				
Rotamer Library[c]	30°	20°	15°	10°	5°
Conformer: bb-ind	-	-	-	-	0.27 ± 0.09 (5.0)
with xtal rotamers	-	0.24 ± 0.09 (5.0)	0.24 ± 0.07 (5.0)	0.26 ± 0.06 (5.0)	0.20 ± 0.13 (5.0)
Rotamer: bb-ind	-	-	0.77 ± 0.42 (2.3)	0.60 ± 0.14 (3.0)	0.60 ± 0.05 (2.7)
with xtal rotamers	0.37 ± 0.17 (5.0)	0.24 ± 0.09 (5.0)	0.24 ± 0.07 (5.0)	0.26 ± 0.06 (5.0)	0.30 ± 0.17 (5.0)
Conformer: bb-dep	-	-	-	0.25 ± 0.12 (5.0)	0.20 ± 0.07 (5.0)
with xtal rotamers	-	0.24 ± 0.09 (5.0)	0.24 ± 0.07 (5.0)	0.22 ± 0.03 (5.0)	0.29 ± 0.09 (4.0)
Rotamer: bb-dep	-	-	-	0.82 ± 0.28 (2.3)	0.66 ± 0.02 (3.0)
with xtal rotamers	-	0.24 ± 0.09 (5.0)	0.24 ± 0.07 (5.0)	0.26 ± 0.06 (5.0)	0.16 ± 0.06 (5.0)

Triosephosphate isomerase

	Rotational step size				
Rotamer Library[c]	30°	20°	15°	10°	5°
Conformer: bb-ind	-	1.87 ± 1.07 (0.7)	3.59 ± 2.28 (1.0)	0.28 ± 0.07 (3.0)	0.24 ± 0.05 (3.0)
with xtal rotamers	-	1.31 ± 0.29 (1.0)	1.95 ± 2.28 (1.3)	0.27 ± 0.06 (3.0)	0.15 ± 0.02 (3.0)
Rotamer: bb-ind	5.09 ± 0.05 (0.3)	0.60 ± 0.12 (1.7)	0.55 ± 0.25 (2.3)	0.34 ± 0.04 (2.3)	0.25 ± 0.08 (3.0)
with xtal rotamers	5.06 ± 0.05 (0.3)	0.60 ± 0.12 (2.0)	0.37 ± 0.04 (3.0)	0.25 ± 0.04 (3.0)	0.15 ± 0.02 (3.0)
Conformer: bb-dep	-	-	-	-	-
with xtal rotamers	-	-	-	-	0.15 ± 0.02 (3.0)
Rotamer: bb-dep	3.28 ± 0.73 (1.7)	0.60 ± 0.12 (1.7)	0.37 ± 0.05 (2.3)	0.31 ± 0.04 (2.3)	0.25 ± 0.08 (3.0)
with xtal rotamers	3.28 ± 0.73 (2.3)	0.60 ± 0.12 (2.3)	0.37 ± 0.05 (3.0)	0.29 ± 0.03 (3.0)	0.15 ± 0.02 (3.0)

[a] Dashes indicate that required contacts were not satisfied in at least one of three trials.
[b] Values are non-hydrogen-atom RMSD in Ångstroms relative to crystallographic ligands or bicyclic ring atom RMSD relative to crystallographic ligand for biotin (i.e., the pentanoic acid moiety was not considered in biotin RMSDs). Averages and standard deviations from three random initial positions are reported. Numbers in parentheses are the number of contacts where the amino acid position was the same as in the wild-type structure, averaged over the three trials. Maximum possible number of wild-type contacts: chorismate mutase, 5; streptavidin, 5; triosephosphate isomerase, 3
[c] bb-ind: backbone-independent, bb-dep: backbone-dependent

Next, we tested various combinations of rotational and translational step sizes starting from random initial ligand positions and using only the backbone-independent conformer library (Figure 3, Table 2). The crystallographic rotamers from the structures under investigation were not included in these calculations. The results show that, subject to the constraints imposed by the geometries defined in the pruning step and the biasing step, more than one combination of rotational and translational step size is viable for each test case and the sensitivity of the result to step size varies among the test cases.

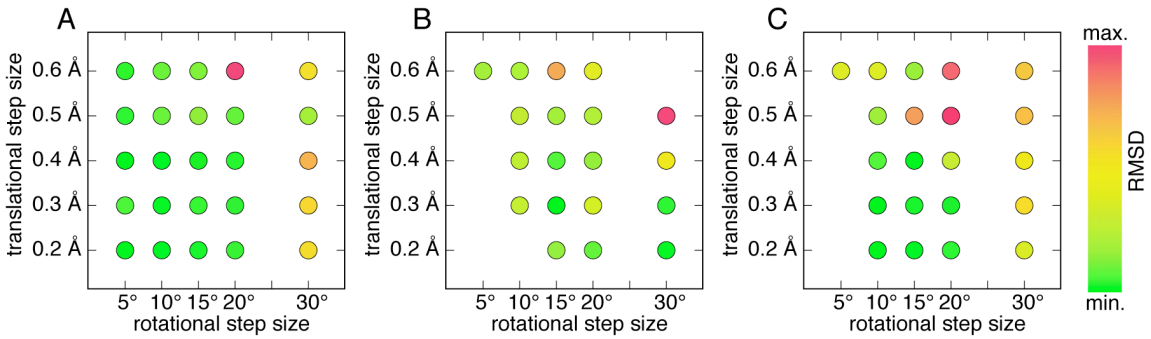

Figure 3: Effect of rotational and translational step sizes. Each spot represents the average of three trials with initial random starting positions. Missing points indicate that one or more trials could not identify wild-type-like contacts or else that the calculation was prohibitively large; no calculations were performed using a 25° rotational step size. Colors indicate non-hydrogen atom RMSD as described in the tables. (*A*) Chorismate mutase (min., 0.53 Å; max., 2.61 Å) (*B*) Streptavidin-biotin (min., 0.57 Å; max., 2.05 Å) (*C*) triosephosphate isomerase (min., 0.44 Å; max., 5.64 Å)

The rotation/translation tests were performed using three initial random starting positions for each system. The starting positions were created by randomly rotating and translating the ligand within a 1 $Å^3$ box around the ligand centroid (or the centroid of the bicyclic ring system in biotin). Using the same atom comparisons as described in the tables, the nine initial positions had RMSDs relative to crystallographic positions of

between 2.1 Å and 4.5 Å, with an average of 3.2 Å. These tests do not provide full, unbiased searches of the active sites. Full active site searches could be conducted using this method by performing separate calculations for grid points distributed evenly through the active site. Given the time required to perform these smaller calculations (Table 2), searching an entire active site using rotational and translational perturbations would be computationally expensive. For example, examining a 3.6 x 3.6 x 3.6 Å grid using the 10° and 0.3 Å step sizes would require an estimated 324 hours on a 16-processor cluster for placement of ligands and catalytic side chains in the chorismate mutase active site. Thus, for initial positioning of a ligand within an active site, rotational and translational placement is inefficient. However, the ability to adjust small molecule position and conformation simultaneously with side-chain optimization should be extremely valuable for refining an initial position identified from a coarser search method.

Table 2: RMSD and number of wild-type contacts as a function of rotational and translational step sizes[a,b]

Chorismate mutase

Translational step size (Å)	Rotational step size 30°	20°	15°	10°	5°	Time (10°, hours)[c]
0.6	1.69 ± 1.54 (2.3)	2.61 ± 1.67 (1.3)	0.77 ± 0.10 (4.3)	0.73 ± 0.02 (4.0)	0.61 ± 0.06 (4.7)	3
0.5	0.91 ± 0.20 (3.7)	0.72 ± 0.07 (4.0)	0.83 ± 0.06 (3.3)	0.74 ± 0.05 (4.0)	0.60 ± 0.13 (4.3)	10
0.4	2.02 ± 1.99 (2.3)	0.60 ± 0.04 (4.7)	0.59 ± 0.13 (4.0)	0.57 ± 0.12 (4.3)	0.53 ± 0.13 (4.3)	11
0.3	1.73 ± 1.51 (2.3)	0.61 ± 0.07 (4.3)	0.62 ± 0.15 (4.3)	0.58 ± 0.07 (4.0)	0.65 ± 0.04 (4.0)	12
0.2	1.71 ± 1.53 (2.3)	0.62 ± 0.10 (4.0)	0.60 ± 0.09 (4.0)	0.54 ± 0.07 (4.0)	0.56 ± 0.05 (4.0)	33

Streptavidin-biotin

Translational step size (Å)	Rotational step size 30°	20°	15°	10°	5°	Time (10°, hours)[c]
0.6	-	1.16 ± 0.60 (3.7)	1.67 ± 1.02 (3.7)	0.88 ± 0.44 (4.3)	0.84 ± 0.48 (4.3)	5
0.5	2.05 ± 0.59 (1.7)	0.91 ± 0.44 (5.0)	0.84 ± 0.61 (5.0)	0.99 ± 0.91 (3.7)	-	18
0.4	1.32 ± 1.39 (3.7)	0.80 ± 0.09 (5.0)	0.67 ± 0.28 (5.0)	0.96 ± 0.72 (3.7)	-	19
0.3	0.63 ± 0.16 (5.0)	1.08 ± 0.49 (5.0)	0.57 ± 0.21 (5.0)	1.03 ± 0.48 (4.3)	-	18
0.2	0.60 ± 0.32 (5.0)	0.70 ± 0.34 (5.0)	0.80 ± 0.24 (5.0)	-	-	-

Triosephosphate isomerase

Translational step size (Å)	Rotational step size 30°	20°	15°	10°	5°	Time (10°, hours)[c]
0.6	3.80 ± 2.14 (0.3)	5.22 ± 0.32 (0.0)	1.29 ± 0.91 (1.3)	2.39 ± 2.54 (1.7)	2.40 ± 2.58 (2.0)	0.4
0.5	3.92 ± 1.94 (0.0)	5.64 ± 0.45 (0.3)	4.47 ± 1.45 (0.0)	1.33 ± 1.01 (1.7)	-	2
0.4	3.13 ± 1.77 (0.3)	1.96 ± 1.05 (2.0)	0.47 ± 0.24 (1.7)	0.78 ± 0.66 (3.0)	-	2
0.3	3.44 ± 1.96 (0.3)	0.59 ± 0.18 (2.0)	0.60 ± 0.29 (2.3)	0.46 ± 0.11 (3.0)	-	2
0.2	2.33 ± 1.80 (0.7)	0.68 ± 0.10 (2.3)	0.49 ± 0.12 (3.0)	0.44 ± 0.11 (3.0)	-	5

[a] Dashes indicate that required contacts were not satisfied in at least one of three trials or that the calculation was too large to complete.

[b] Values are non-hydrogen atom RMSD in Ångstroms relative to crystallographic ligands or bicyclic atom RMSD relative to crystallographic ligand for biotin (i.e., the pentanoic acid moiety was not considered in biotin RMSDs). Averages and standard deviations from three random initial positions are reported. Numbers in parentheses are the number of contacts where the amino acid position was the same as in the wild-type structure, averaged over the three trials. Maximum possible number of wild-type contacts: chorismate mutase, 5; streptavidin, 5; triosephosphate isomerase, 3

[c] Wall clock time; calculations performed on a 16-processor cluster

Targeted ligand placement

A second approach places the small molecule with reference to a contacting side chain (Figure 4). In this approach, one or more small molecule variations are placed for every rotamer of the selected contacting side chain in every putative active-site position. This process has the advantage that ligand poses are targeted more efficiently to orientations that are able to make productive side-chain contacts. Previous computational enzyme design work utilized similar approaches.[1, 16, 17] In contrast to previous methods, however, our procedure does not maintain any association between the targeting rotamer and the small molecule—once the set of ligand conformations and orientations is constructed in step one, the ligand variations are all subjected to pruning, pairwise energy calculations, and optimization as independent entities in the calculation. An implication of this procedure is that a ligand may engage in a catalytic contact with a rotamer, amino acid, or protein position that differs from those of the side-chain rotamer that was originally used to place that ligand.

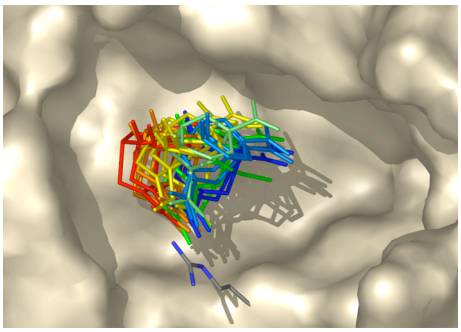

Figure 4: Targeted placement procedure. For a given side-chain rotamer, small molecule ligands are placed such that they are able to meet specified geometric criteria. This is repeated for every possible conformation of the amino acid at every designed position. Shown is a subset of orientations of a chorismate mutase transition-state structure in contact with one conformation of arginine.

We tested the effect of four types of side-chain libraries on the ability of a targeted placement process to find wild-type-like ligand positions and contacts. For the three test cases, the following side-chain contacts were used to anchor the ligand: chorismate mutase, C11 carboxylate to arginine; streptavidin, N1 to aspartate; triosephosphate isomerase, O2 and O3 to histidine. For each contact type, variations were allowed in the geometry of the contact, including the contacting atoms (NH1-NH2 vs. NE-NH1 for arginine) and variations in defined distances, angles, and dihedrals of the contact.

As with the rotational and translational search, success in achieving native active-site conformations was highly dependent on the side-chain library used (Table 3). Only the backbone-independent conformer library yielded results for all three test cases that were comparable to those with crystallographic rotamers included. Using that library, all three systems returned all wild-type contacts with low ligand RMSD relative to the crystallographic position. As with the rotation/translation search, the chorismate mutase case showed the strongest sensitivity to rotamer library. Inspection of the structures revealed that an arginine side chain (Arg 28) occupies a conformation in the inhibitor-bound, active enzyme structure that was not well approximated in the other rotamer libraries.

The targeted placement approach allowed a thorough and directed search of active-site conformational space, including between 10^6 and 10^9 small-molecule orientations and conformations spread throughout the active site. In contrast to the rotation/translation method, a full active-site search took between one and eighteen hours to complete using the backbone-independent conformer library and no initial starting

position was required. This method offers an efficient first step for defining active-site geometry in a new protein scaffold. One shortcoming is that it may be difficult to sample the many geometrical variations of a flexible hydrogen-bonding interaction. For example, the 972 variations in guanidino-carboxylate contact geometry sampled in the chorismate mutase case are probably adequate to reflect flexibility in this relatively rigid dual hydrogen-bonding interaction. A less restrained interaction, however, such as a serine hydrogen bonding with a sterically unrestricted ligand carbonyl oxygen, results in a compromise between maintaining a manageable calculation size and modeling contact flexibility. One solution is to use a targeted method to find an initial ligand position within the binding site and then, in a second calculation, optimize both active-site packing and fine rotational and translational placement of the ligand.

Table 3: Results from targeted placement procedure as a function of rotamer library

Chorismate mutase

Rotamer library[a]	log(initial ligand variations)	RMSD (Å)[b] (WT contacts)	Time (hours)[c]
Conformer: bb-ind	7.88	0.60 (5)	16
with xtal rotamers	7.88	0.68 (3)	18
Rotamer: bb-ind	8.18	3.61 (0)	51
with xtal rotamers	8.18	0.66 (4)	62
Conformer: bb-dep	7.64	3.62 (1)	8
with xtal rotamers	7.64	0.68 (4)	9
Rotamer: bb-dep	7.76	2.31 (1)	14
with xtal rotamers	7.76	0.66 (4)	16

Streptavidin-biotin

Rotamer library[a]	log(initial ligand variations)	RMSD (Å)[b] (WT contacts)	Time (hours)[c]
Conformer: bb-ind	7.07	0.64 (5)	1.4
with xtal rotamers	7.07	0.64 (5)	1.4
Rotamer: bb-ind	7.20	0.54 (4)	3.5
with xtal rotamers	7.20	0.34 (4)	3.4
Conformer: bb-dep	6.35	0.37 (5)	0.2
with xtal rotamers	6.35	0.54 (4)	0.2
Rotamer: bb-dep	7.17	3.50 (0)	2.6
with xtal rotamers	7.17	0.19 (5)	2.8

Triosephosphate isomerase

Rotamer library[a]	log(initial ligand variations)	RMSD (Å)[b] (WT contacts)	Time (hours)[c]
Conformer: bb-ind	7.31	0.49 (3)	1.3
with xtal rotamers	7.31	0.49 (3)	1.3
Rotamer: bb-ind	7.78	0.46 (3)	8.7[d]
with xtal rotamers	7.78	0.46 (3)	87[d]
Conformer: bb-dep	6.82	7.51 (0)	0.3
with xtal rotamers	6.82	0.78 (3)	0.3
Rotamer: bb-dep	7.58	0.51 (3)	4.3[d]
with xtal rotamers	7.58	0.51 (3)	4.9[d]

[a] bb-ind, backbone-independent; bb-dep, backbone-dependent

[b] RMSDs calculated as described in Table 1. Maximum possible number of wild-type contacts: chorismate mutase, 5; streptavidin, 5; triosephosphate isomerase, 3

[c] Wall clock time; calculations performed on a 16-processor cluster

[d] Calculation was performed as a series of smaller calculations.

Sequence design

The computational tests described in the previous sections were designed to evaluate the effects of calculation parameters on recovery of native enzyme geometries, and the design of active-site residues was limited to catalytic side chains. However, the general procedure described here is equally amenable to full active-site design calculations.

In previously published work, 18 active site residues of *E. coli* chorismate mutase were redesigned simultaneously with rotational and translational relaxation of the transition-state structure from the starting crystallographic position.[31] The six predicted mutations were experimentally investigated and some were found to confer increased catalytic efficiency or thermostability.[31] A detrimental mutation predicted in the study underscored the importance of continued work on energy functions. In the calculation that motivated this experimental work, the initial starting position of the small molecule was taken from the crystal structure and a limited degree of rotational and translational optimization was employed.

We performed a test calculation to demonstrate that small molecules can be placed simultaneously with full active-site side-chain optimization, without reference to any known starting position. In a sample calculation using *E. coli* chorismate mutase, the targeted placement method was used to identify 10^7 small-molecule variations. In this example, after the geometric pruning step and elimination of variants with backbone steric clashes, 155 small-molecule variations remained. These variants were evaluated combinatorially with ten different side-chain identities in twelve active-site positions.

Using FASTER for optimization, the calculation took approximately 9 hours to complete on a 16-processor cluster with about 70% of the total calculation time consumed in calculating a surface-area-based solvation term.

Conclusions

The described procedures allow the incorporation of small-molecule placement directly into sequence design calculations. The test calculations performed suggest that the results of computational enzyme design processes can be quite sensitive to calculation parameters, including the rotamer library used and the coarseness of ligand positioning. These results emphasize that the conformational space of a calculation must be explored before meaningful conclusions can be reached about energy functions.

Given that we still have much to learn about the complex relationship between protein structure and catalytic activity,[32, 33] luck and choice of system may continue to play a role in the success of *de novo* computational enzyme design efforts for some time. However, the power of computational enzyme design to stringently evaluate our understanding of the energetics of catalysis should not be overlooked. Experimental feedback gained from both successful and unsuccessful designs will make it possible to critically examine energy functions for modeling active sites. Employing quality transition-state structures derived from *ab initio* calculations and experimental evidence will help computational design experiments to provide more meaningful information about the effectiveness of energy functions. The use of large side-chain structural libraries and fine movements of transition-state structures will help to reduce errors from conformational sampling. Backbone relaxation and multi-state design will offer other

important tools to improve the value of design calculations. Finally, the construction of gene libraries or large numbers of computationally designed variants has great potential for overcoming the shortcomings of enzyme design models,[34] but results from these experiments will be most useful for furthering our understanding of catalysis and design if both active and inactive variants are reported. By critically evaluating current methods for computational enzyme design, we will move closer to a deeper and more practically useful understanding of the sequence determinants of enzyme activity in the future.

Methods

Structures and charges

PDB files were used without minimization (*E. coli* chorismate mutase, 1ecm;[35] *S. avidinii* streptavidin, 1mk5;[36] *S. cerevisiae* triosephosphate isomerase, 1ney).[37] Hydrogens were added with Reduce.[38]

A library of ligand internal conformations was created for each system as follows. Chorismate mutase: An HF/6-31G* *ab initio* transition-state structure[39] was used with only one variation—the O4 hydroxyl proton was allowed to occupy three positions, 60°, 180°, and -35°, defined by the H-C-O-H dihedral angle. The minima in a torsional profile at the HF/6-31G* level were at approximately 180° and -35°, and 60° was included as an option because hydrogen-bonding patterns in chorismate mutases from other species suggested population of that region of torsional space. Streptavidin: Four rotatable bonds in biotin were allowed to occupy three positions each (60°, -60°, 180° for sp3-sp3 bonds and 30°, 90°, 150° for the symmetric carboxylate group). Thirty-four conformations were excluded because of high internal energy calculated using the van

der Waals component of the DREIDING force field.[40] Triosephosphate isomerase: The pdb structure used was the Michaelis complex with the substrate dihydroxyacetone phosphate. In ground-state dihydroxyacetone phosphate, two rotatable bonds (defined by the P-O-C-C and C-C-O-H dihedral angle) were allowed to occupy three positions each (60°, -60°, 180°). Three conformations were excluded because of high internal DREIDING van der Waals energy.

Ligand atomic charges were obtained by fitting charges to electrostatic potential from HF/6-31G* single-point energy calculations using[19] the transition-state structure (chorismate mutase) or crystallographic ground-state structure (biotin, dihydroxyacetone phosphate). *Ab initio* calculations and charge determinations were performed using Spartan (Wavefunction, Inc.) or Jaguar (Schrödinger, Inc.).

Side-chain rotamer libraries

Standard backbone-dependent and backbone-independent rotamer libraries were used with expansion by one standard deviation about $\chi 1$ and $\chi 2$.

Crystallographic conformer libraries were prepared using coordinates from 149,813 side chains selected from 1011 unique structures. A clustering algorithm was developed based on ideas described by Shetty et al.[22] and is described briefly here. Every side-chain conformation from the raw data set is assigned to exactly one cluster. Each cluster is represented by the centroid, which is the member with coordinates closest to the average coordinates of all cluster members. A conformer library consists of a list of all of the cluster representatives and their coordinates. In our clustering algorithm, clusters are assigned through discrete clustering moves: *Switch* allows a single raw conformer to

leave one cluster and join another; *Merge* combines two clusters into one; *Split* allows a raw conformer to start a new cluster on its own. These moves are depicted in Figure 5.

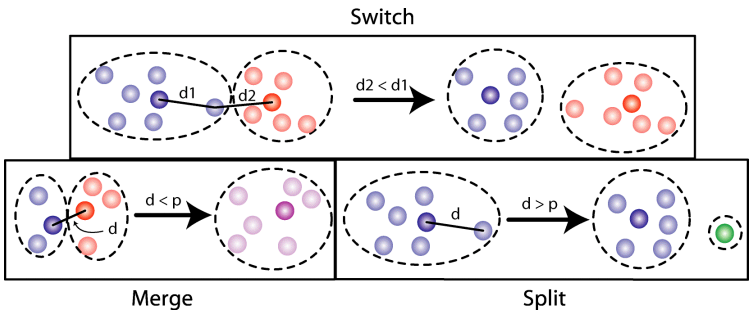

Figure 5: **The three clustering moves are illustrated by showing the state of a sample system before and after the move is performed.** Each dot represents a single side-chain conformation taken from the PDB. Distances represent side-chain RMSDs between pairs of conformers. Dots sequestered together by a dashed line and colored the same are members of the same cluster. Darker-colored dots denote cluster representatives.

RMSDs between pairs of conformers are compared to determine whether or not to apply a particular move. *Switch* is applied so that each raw conformer is a member of the cluster whose centroid is closest to it. *Merge* and *Split* are applied based on the value of the clustering parameter *p*: two clusters are merged if their centroids are within *p* of each other, whereas a conformer splits off and starts a new cluster if the closest centroid of any existing cluster is farther than *p* from it. The clustering moves are applied as follows until the number of clusters converges:

1. Start with a small number of clusters (1 was used in this work), and randomly assign a single raw conformer to each as the sole member and cluster representative.
2. Assign each raw conformer in the data set to the cluster whose centroid is closest.
3. While the number of clusters is not converged:
    a. Iteratively attempt to *Merge* pairs of clusters until no cluster can be further merged.
    b. For each conformer C:
        i. Measure the distance *d* between C and the centroid of every existing cluster.
        ii. If the distance *d* to the closest cluster centroid is greater than *p*, *Split* C off as its own cluster.
        iii. Else, *Switch* C to the closest cluster.
        iv. Recompute the centroid for every cluster that has changed membership.

The algorithm allows the construction of both backbone-dependent and backbone-independent libraries to custom sizes by using clustering factor *p* to define the desired degree of similarity between independent conformers. In this work, clustering factors of 0.3 Å and 1.0 Å were used for backbone-dependent and backbone-independent rotamer libraries, respectively.

For all calculation types, conformer libraries were smaller than the standard rotamer libraries. As an example, the number of side-chain conformations for the chorismate mutase calculations described in Table 3 were as follows: backbone-independent rotamer, 14229; backbone-independent conformer, 5955; backbone-dependent rotamer, 7945; and backbone-dependent conformer, 5539.

Calculation parameters

All non-Gly, non-Pro residues reasonably within the natural active sites were included in calculations. Residues with any atom within a 5 Å radius from any atom in the crystallographic ligands were included, less those residues separated from the natural ligand by backbone elements and plus a few adjacent residues not within the 5 Å cutoff. The positions designed were (all in chain A unless otherwise designated): chorismate mutase, 28, 32, 35, 39, 46, 47, 48, 51, 52, 55, 81, 84, 85, 88, 7B, 11B, 14B, 18B; streptavidin, 23, 24, 25, 27, 43, 45, 46, 47, 49, 50, 79, 86, 88, 90, 92, 108, 110, 112, 128, 130; and triosephosphate isomerase, 10, 12, 95, 97, 165, 170, 211, 230.

In ligand placement test cases, designed residues were restricted to ligand-contacting residues or alanine as follows: Arg, Lys, Gln, Glu, or Ala in chorismate mutase; Ser, Asn, Tyr, Asp, or Ala in streptavidin; and Glu, His, Lys, or Ala in triosephosphate isomerase. Four calculations on triosephosphate isomerase were run as smaller component calculations, as indicated in Table 2, because of prohibitive size as a single calculation.

Energy functions and optimization

Energy functions included scaled van der Waals,[24] hydrogen-bonding, and electrostatic terms.[25] A surface-area-based solvation potential[27] was used in sequence design calculations but not for ligand placement, where solvation energy would have been heavily outweighed by geometric considerations. Sequences were optimized with respect to the energy function using FASTER[28, 29] or HERO.[30] On occasion, a top-ranked sequence contained more than one instance of a given specified geometric contact, owing to the energy benefit applied for these contacts. In these cases, Monte Carlo[41, 42] was used to sample around the global minimum energy sequence, and the top-ranked sequence with a single instance of each geometric contact was reported.

References

1. Bolon, D. N.; Mayo, S. L., Enzyme-like proteins by computational design. *Proceedings of the National Academy of Sciences of the United States of America* **2001,** *98* (25), 14274–14279.

2. Jiang, L.; Althoff, E. A.; Clemente, F. R.; Doyle, L.; Rothlisberger, D.; Zanghellini, A.; Gallaher, J. L.; Betker, J. L.; Tanaka, F.; Barbas, C. F.; Hilvert, D.; Houk, K. N.; Stoddard, B. L.; Baker, D., De novo computational design of retro-aldol enzymes. *Science* **2008,** *319* (5868), 1387–1391.

3. Rothlisberger, D.; Khersonsky, O.; Wollacott, A. M.; Jiang, L.; DeChancie, J.; Betker, J.; Gallaher, J. L.; Althoff, E. A.; Zanghellini, A.; Dym, O.; Albeck, S.; Houk, K. N.; Tawfik, D. S.; Baker, D., Kemp elimination catalysts by computational enzyme design. *Nature* **2008,** *453* (7192), 190–U4.

4. Mendes, J.; Guerois, R.; Serrano, L., Energy estimation in protein design. *Current Opinion in Structural Biology* **2002,** *12* (4), 441–446.

5. Vizcarra, C. L.; Mayo, S. L., Electrostatics in computational protein design. *Current Opinion in Chemical Biology* **2005,** *9* (6), 622–626.

6. Dahiyat, B. I.; Mayo, S. L., De novo protein design: Fully automated sequence selection. *Science* **1997,** *278* (5335), 82–87.

7. Malakauskas, S. M.; Mayo, S. L., Design, structure and stability of a hyperthermophilic protein variant. *Nature Structural Biology* **1998,** *5* (6), 470–475.

8. Kuhlman, B.; Dantas, G.; Ireton, G. C.; Varani, G.; Stoddard, B. L.; Baker, D., Design of a novel globular protein fold with atomic-level accuracy. *Science* **2003,** *302* (5649), 1364–1368.

9. Looger, L. L.; Dwyer, M. A.; Smith, J. J.; Hellinga, H. W., Computational design of receptor and sensor proteins with novel functions. *Nature* **2003,** *423* (6936), 185–190.

10. Shimaoka, M.; Shifman, J. M.; Jing, H.; Takagi, L.; Mayo, S. L.; Springer, T. A., Computational design of an integrin I domain stabilized in the open high affinity conformation. *Nature Structural Biology* **2000,** *7* (8), 674–678.

11. Pierce, N. A.; Winfree, E., Protein design is NP-hard. *Protein Engineering* **2002,** *15* (10), 779–782.

12. Taylor, R. D.; Jewsbury, P. J.; Essex, J. W., A review of protein-small molecule docking methods. *Journal of Computer-Aided Molecular Design* **2002,** *16* (3), 151–166.

13. Lilien, R. H.; Stevens, B. W.; Anderson, A. C.; Donald, B. R., A novel ensemble-based scoring and search algorithm for protein redesign and its application to modify the

substrate specificity of the gramicidin synthetase a phenylalanine adenylation enzyme. *Journal of Computational Biology* **2005,** *12* (6), 740–761.

14. Chakrabarti, R.; Klibanov, A. M.; Friesner, R. A., Sequence optimization and designability of enzyme active sites. *Proceedings of the National Academy of Sciences of the United States of America* **2005,** *102* (34), 12035–12040.

15. Chakrabarti, R.; Klibanov, A. M.; Friesner, R. A., Computational prediction of native protein ligand-binding and enzyme active site sequences. *Proceedings of the National Academy of Sciences of the United States of America* **2005,** *102* (29), 10153–10158.

16. Dwyer, M. A.; Looger, L. L.; Hellinga, H. W., Computational design of a biologically active enzyme. (Retracted article: See vol. 319, p. 569, 2008). *Science* **2004,** *304* (5679), 1967–1971.

17. Dwyer, M. A.; Looger, L. L.; Hellinga, H. W., Computational design of a biologically active enzyme. (Retraction of vol. 304, p. 1967, 2004). *Science* **2008,** *319* (5863), 569–569.

18. Hellinga, H. W.; Richards, F. M., Construction of New Ligand-Binding Sites in Proteins of Known Structure .1. Computer-Aided Modeling of Sites with Predefined Geometry. *Journal of Molecular Biology* **1991,** *222* (3), 763–785.

19. Dunbrack, R. L.; Cohen, F. E., Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science* **1997,** *6* (8), 1661–1681.

20. Lovell, S. C.; Word, J. M.; Richardson, J. S.; Richardson, D. C., The penultimate rotamer library. *Proteins—Structure Function and Genetics* **2000,** *40* (3), 389–408.

21. Ponder, J. W.; Richards, F. M., Tertiary Templates for Proteins - Use of Packing Criteria in the Enumeration of Allowed Sequences for Different Structural Classes. *Journal of Molecular Biology* **1987,** *193* (4), 775–791.

22. Shetty, R. P.; de Bakker, P. I. W.; DePristo, M. A.; Blundell, T. L., Advantages of fine-grained side chain conformer libraries. *Protein Engineering* **2003,** *16* (12), 963–969.

23. Xiang, Z. X.; Honig, B., Extending the accuracy limits of prediction for side-chain conformations. *Journal of Molecular Biology* **2001,** *311* (2), 421–430.

24. Dahiyat, B. I.; Mayo, S. L., Probing the role of packing specificity in protein design. *Proceedings of the National Academy of Sciences of the United States of America* **1997,** *94* (19), 10172–10177.

25. Dahiyat, B. I.; Gordon, D. B.; Mayo, S. L., Automated design of the surface positions of protein helices. *Protein Science* **1997,** *6* (6), 1333–1337.

26. Lazaridis, T.; Karplus, M., Effective energy function for proteins in solution. *Proteins—Structure Function and Genetics* **1999,** *35* (2), 133–152.

27. Street, A. G.; Mayo, S. L., Pairwise calculation of protein solvent-accessible surface areas. *Folding & Design* **1998,** *3* (4), 253–258.

28. Allen, B. D.; Mayo, S. L., Dramatic performance enhancements for the FASTER optimization algorithm. *Journal of Computational Chemistry* **2006,** *27* (10), 1071–1075.

29. Desmet, J.; Spriet, J.; Lasters, I., Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins* **2002,** *48* (1), 31–43.

30. Gordon, D. B.; Hom, G. K.; Mayo, S. L.; Pierce, N. A., Exact rotamer optimization for protein design. *Journal of Computational Chemistry* **2003,** *24* (2), 232–243.

31. Lassila, J. K.; Keeffe, J. R.; Oelschlaeger, P.; Mayo, S. L., Computationally designed variants of Escherichia coli chorismate mutase show altered catalytic activity. *Protein Engineering Design & Selection* **2005,** *18* (4), 161–163.

32. Benkovic, S. J.; Hammes-Schiffer, S., A perspective on enzyme catalysis. *Science* **2003,** *301* (5637), 1196–1202.

33. Kraut, D. A.; Carroll, K. S.; Herschlag, D., Challenges in enzyme mechanism and energetics. *Annual Review of Biochemistry* **2003,** *72*, 517–571.

34. Bolon, D. N.; Voigt, C. A.; Mayo, S. L., De novo design of biocatalysts. *Current Opinion in Chemical Biology* **2002,** *6* (2), 125–129.

35. Lee, A. Y.; Karplus, P. A.; Ganem, B.; Clardy, J., Atomic-Structure of the Buried Catalytic Pocket of Escherichia-Coli Chorismate Mutase. *Journal of the American Chemical Society* **1995,** *117* (12), 3627–3628.

36. Hyre, D. E.; Le Trong, I.; Merritt, E. A.; Eccleston, J. F.; Green, N. M.; Stenkamp, R. E.; Stayton, P. S., Cooperative hydrogen bond interactions in the streptavidin-biotin system. *Protein Science* **2006,** *15* (3), 459–467.

37. Jogl, G.; Rozovsky, S.; McDermott, A. E.; Tong, L., Optimal alignment for enzymatic proton transfer: Structure of the Michaelis complex of triosephosphate isomerase at 1.2-angstrom resolution. *Proceedings of the National Academy of Sciences of the United States of America* **2003,** *100* (1), 50–55.

38. Word, J. M.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C., Asparagine and glutamine: Using hydrogen atom contacts in the choice of side-chain amide orientation. *Journal of Molecular Biology* **1999,** *285* (4), 1735–1747.

39. Wiest, O.; Houk, K. N., On the Transition-State of the Chorismate-Prephenate Rearrangement. *Journal of Organic Chemistry* **1994,** *59* (25), 7582–7584.

40. Mayo, S. L.; Olafson, B. D.; Goddard, W. A., Dreiding — a Generic Force-Field for Molecular Simulations. *Journal of Physical Chemistry* **1990,** *94* (26), 8897–8909.

41. Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E., Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics* **1953,** *21* (6), 1087–1092.

42. Voigt, C. A.; Gordon, D. B.; Mayo, S. L., Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *Journal of Molecular Biology* **2000,** *299* (3), 789–803.