Comparing search algorithms for computational protein design

David Mignon and Thomas Simonson*

†Laboratoire de Biochimie (UMR CNRS 7654), Dept. of Biology, Ecole Polytechnique, Palaiseau, France *Corresponding author. Email: thomas.simonson@polytechnique.fr

Abstract

1 Introduction

Computational protein design (CPD) has developed into an important tool for biotechnology [1–6]. Starting from a 3D structural model, CPD explores a large space of possible sequences and conformations, to identify protein variants that have certain predefined properties, such as stability or ligand binding. Conformational space is usually defined by a library of sidechain rotamers, which can be discrete or continuous, and by a finite set of backbone conformations or a specific repertoire of allowed backbone deformations. The energy function usually combines physical and empirical terms [7–9]. Both solvent and the unfolded protein state are described implicitly.

The number of amino acid positions that are allowed to mutate can vary, depending on the problem of interest, from 2 or 3 to several dozen. Thus, the combinatorial complexity can be enormous, so that speed is important, as well as accuracy. In addition, it is usually important to identify not one but several high-scoring sequences, for at least three reasons. First, if the typical error in the energy function is σ_E, we should enumerate all the possible sequences/structures within one or two σ_E of the optimal one. Second, it may be of interest to characterize the diversity of a sequence family, by enumerating sets of sequences compatible with its backbone fold (the "inverse folding problem") [10–14]. Third, we may want to compute properties that are averaged over structural and possibly sequence fluctuations at a given temperature, which requires that we explore solutions within the thermal range. An example is the calculation of ligand binding constants, following a method introduced recently [15]. Calculation of acid/base constants by constant-pH Monte Carlo can also be seen as a subproblem of CPD, where sidechain protonation state changes are treated as mutations [16–18].

Thus, the complexity and cost of a CPD calculation will depend on several factors. While energy calculations usually represent the bulk of the cost, the power and efficiency of the exploration method are also important. Several exploration methods exist that can identify exactly the global minimum energy sequence and conformation, or GMEC. These include "dead end elimination" methods, or DEE [19, 20], branch-and-bound methods [21, 22], and cost function network methods [23, 24]. While some of these methods can handle large problems, they usually cannot enumerate suboptimal solutions within a large interval σ_E above the GMEC (more than a few kcal/mol). Partly for this reason, stochastic methods remain popular, such as Monte Carlo (MC) [8, 25]. MC has two advantages:

with an appropriate setup, it samples sequences/conformations from a known, Boltzmann distribution, and it can be readily combined with enhanced sampling methods developed in the broader field of biomolecular simulations, such as Replica Exchange or umbrella sampling [26, 27].

Our goal here is to compare four exploration methods for a series of CPD problems of increasing complexity. The first method is a recently developed, exact method, based on "cost function networks", or CFN, which can identify the global minimum energy conformation, or GMEC exactly in favorable cases [23, 24]. The cost function is the energy, and the network refers to the set of interacting amino acids. The CFN method uses a depth-first branch-and-bound search through a tree of rotamer assignments, with fast integer arithmetic for the energy evaluations. It can also enumerate all the sequence/conformation combinations within a given energy range δE (not too large) above the GMEC. It is implemented in the Toulbar2 program, by Schiex and coworkers. The second method is a heuristic, stochastic method that is not guaranteed to find the GMEC but has been effective in applications [13, 14, 28]. The third method is a Monte Carlo (MC) exploration, which samples sequences/conformations from a Boltzmann distribution [26, 29, 30]. The fourth is an enhanced, multi-walker MC, which performs "replica exchange" [31–33]. Several walkers are propagated at different temperatures, and exchange conformations at regular intervals according to a MC test. We refer to it as REMC.

We use a CPD model that is rather simple but representative of a large class of applications. We use a discrete set of sidechain rotamers, a fixed backbone structure, and we assume that the energy function is pairwise additive; i.e., the energy has the form of a sum over residue pairs [34–36]. With these simplifications, all possible residue pair interactions can be computed ahead of time and stored in a lookup table [37]; exploration is then done in a second stage. Thus, the cost of energy calculations and sequence/structure exploration are well-separated. The method is implemented in the Proteus CPD package [35, 36] (except for the CFN sequence exploration, done with Toulbar2). Our MC framework is presented in some detail below; the other methods are recalled more briefly.

We considered nine proteins from three structural classes: SH3, SH2, and PDZ domains. For each one, we chose different numbers and sets of residues to mutate and we applied the different exploration methods, using several possible parameterizations for each one. To characterize the different methods, we compared their speed, their ability to identify the GMEC, and their sampling of suboptimal sequences/conformations above

the GMEC. The designed sequences were characterized by computing their similarity to natural sequences, their classification by the Superfamily fold recognition tool [38, 39], and their sequence entropies. For the few cases where there were large differences between the methods (several kcal/mol between best-scoring sequences), the 3D structural models were compared. Overall, the heuristic method is the most successful in identifying low energy solutions, while REMC is almost as successful but has the advantage of sampling from a Boltzmann distribution over a large energy range, yielding thermal averages.

2 Methods

2.1 Monte Carlo: general framework

We consider a polypeptide of n amino acids. Its sequence S is written $\mathrm{S} = t_1 t_2 \cdots t_n$, where t_i is the sidechain type of amino acid i. We assume that each amino acid i can take on a few different types t, t', ... that form a set T_i. For each sequence, there are two classes of structures: folded and unfolded. For the folded form, all the sequences S share the same, precise geometry for the polypeptide backbone; only the sidechain positions can vary. Specifically, the sidechain of each amino acid i can explore a few discrete conformations or "rotamers" r, r', ... (around 10 per type t_i). The structure of the unfolded form is not specified; the energy is assumed to be independent of the particular unfolded structure, and to have the additive form:

$$E_u(S) = \sum_{i=1}^{n} E_u(t_i) = \sum_{i=1}^{n} \left(e_u(t_i) - kT \log n_u(t_i) \right), \tag{1}$$

where $E_u(t_i)$ is a free energy associated with sidechain type t_i in the unfolded state, and the rightmost form separates it into an energy component $e_u(t_i)$ and a conformational entropy term, where kT is the thermal energy and $n_u(t_i)$ is the number of conformations or rotamers available to sidechain type t_i in the unfolded state.

We perform a Monte Carlo simulation [26, 29, 30] where one copy of the folded protein is explicitly represented. The unfolded state is included implicitly, by propagating the simulation with the energy function $E_M = E_f - E_u$ (the folding energy). One possible elementary MC move is to change a rotamer r_i in the current folded sequence; the energy change is $\Delta E_M = \Delta E_f = E(...t_i, r_i'...) - E(...t_i, r_i...)$. Another possible move is a mutation: we modify the sidechain type $t_i \to t_i'$ at a chosen position i in the folded protein, assigning

a particular rotamer r'_i to the new sidechain. The energy change is

$$\Delta E_M = \Delta E_f - \Delta E_u = (E_f(...t'_i, r'_i...) - E_f(...t_i, r_i...)) - (E_u(t'_i) - E_u(t_i)) \tag{2}$$

ΔE_M measures the stability change due to the mutation (for the given set of rotamers); it is as if we performed the reverse mutation $t'_i \to t_i$ in the unfolded form.

If the moves are generated and accepted with an appropriate Metropolis-like scheme, the Markov chain will visit states according to their Boltzmann probability:

$$p_M(S, c) = \frac{e^{-\beta(E_f(S,c)-E_u(S))}}{\sum_{S'} \sum_{c'} e^{-\beta(E_f(S',c')-E_u(S'))}} \tag{3}$$

where $\beta = 1/kT$ and the subscript M indicates probabilities sampled by the Markov chain. For two conformations c, c' of sequence S, the Markov probability ratio is $p_M(S, c)/p_M(S, c') = e^{-\beta(E_f(S,c)-E_f(S,c'))}$. For two sequences S, S', the probability ratio is

$$\frac{p_M(S)}{p_M(S')} = \frac{\sum_c e^{-\beta(E_f(S,c)-E_u(S))}}{\sum_{c'} e^{-\beta(E_f(S',c')-E_u(S'))}} = \frac{e^{-\beta\Delta G_{\text{fold}}(S)}}{e^{-\beta\Delta G_{\text{fold}}(S')}} \tag{4}$$

In the ratio of Markov probabilities, we recognize the ratio of Boltzmann factors for S and S' folding, so that we have the second equality, where $\Delta G_{\text{fold}}(S)$ denotes the folding free energy of sequence S (respectively, S').

Eq. (4) has a simple interpretation: the Markov chain, with the chosen energy function $E_M = E_f - E_u$ and appropriate move probabilities, leads to the same distribution of states as a macroscopic, equilibrium, physical system where all sequences S, S', ... are present at equal concentrations, and are distributed between their folded and unfolded states according to their relative stabilities. This is exactly the experimental system we want our Markov chain to mimic. In this interpretation, a Monte Carlo mutation move S $\to$ S' amounts to unfolding one copy of S and refolding one copy of S'.

It remains to specify the move generation probabilities and choose an appropriate acceptance scheme [26, 29, 30]. Let $\alpha(o \to n)$ be the probability to select a trial move between two states o and n and $acc(o \to n)$ the probability to accept it. If the simulation obeys detailed balance, we have

$$N(o)\pi(o \to n) = N(n)\pi(n \to o), \tag{5}$$

where $N(o)$, $N(n)$ are the equilibrium populations of states o and n. With "ergodic" move sets such as the one used here (see below), detailed balance is guaranteed in the limit

of a very long simulation. To produce Boltzmann statistics, we choose the acceptance probabilities [26, 29, 30]:

$$acc(o \to n) = \exp(-\beta \Delta E_M) \frac{\alpha(n \to o)}{\alpha(o \to n)} \quad \text{if } \Delta E_M > 0; 1 \text{ otherwise} \tag{6}$$

where $\Delta E_M = E_M(n) - E_M(o)$ is the o $\to$ n energy difference.

For a rotamer move at a particular position in the polypeptide chain, of type t, we define the move generation probability as $\alpha(o \to n) = \frac{1}{n_f(t)} = \alpha(o \to n)$; all possible choices for the new rotamer are equiprobable, forward and backward rotamer moves have the same generation probability, and Eq. (6) reduces to the simple Metropolis test [29].

For a mutation move at a particular position, we define $\alpha(o \to n)$ as follows:
(a) select a new type t' with equal probabilities $\alpha_t(o \to n) = \frac{1}{N}$ for all N possible types;
(b) choose a rotamer r' for the new sidechain with equal probabilities $\alpha_r(o \to n) = \frac{1}{n_f(t')}$ for all $n_f(t')$ possible folded state rotamers.
The overall probability is therefore

$$\alpha(o \to n) = \alpha_t(o \to n)\alpha_r(o \to n) = \frac{1}{N n_f(t')} \tag{7}$$

The $o \to n$ and $n \to o$ probabilities are different whenever the old and new sidechain types have different numbers of possible rotamers. With these move probabilities, the mutation acceptance probability can be written:

$$\begin{aligned} acc(t \to t') &= e^{-\beta(\Delta E_f - \Delta E_u)} \frac{n_f(t)}{n_f(t')} = e^{-\beta(\Delta E_f - \Delta e_u)} \frac{n_f(t) n_u(t')}{n_u(t) n_f(t')} \quad \text{if } \Delta E_M > 0 & (8) \\ &= 1 \quad \text{otherwise} & (9) \end{aligned}$$

If the number of rotamers in the folded and unfolded states are the same, $n_u = n_f$, the fraction on the right will cancel out. However, the rotamer numbers also appear in the energy change that determines whether the move is uphill, ΔE_M.

With REMC, several simulations ("replicas" or "walkers") are propagated in parallel, at different temperatures; periodic swaps are attempted between two walkers's conformations. The swap is accepted with probability

$$acc(t \to t') = \text{Min}\left[1, e^{(\beta_i - \beta_j)(\Delta E_i - \Delta E_j)}\right] \tag{10}$$

where β_i, β_j are the inverse temperatures of the two walkers and ΔE_i, ΔE_j are their folding energies [32, 33].

2.2 MC and REMC: implementation details

For plain MC, we use one- and two-position moves, where either rotamers or types are changed. For two-position moves, the second position is selected among those that have a significant interaction energy with the first one. For REMC, we use four or eight walkers, with temperatures T_i that range from xxx and xxx kcal/mol, and are spaced in a geometric progression: T_{i+1}/T_i = constant, following Kofke [32]. Conformation swaps are attempted at regular intervals, only between walkers at adjacent temperatures. Typical parameter setting are given in Table 1; variations are sometimes used, as detailed in Results.

2.3 Heuristic sequence optimization

The heuristic sequence optimization uses an iterative minimization [28, 35]. One "heuristic cycle" proceeds as follows: an initial amino acid sequence and set of sidechain rotamers are chosen randomly. These are improved in a stepwise way. At a given amino acid position i, the best amino acid type and rotamer are selected, with the rest of the sequence held fixed. The same is done for the following position $i + 1$, and so on, performing multiple passes over the amino acid sequence until the energy no longer improves (or a set, large number of passes is reached). The final sequence, rotamer set, and energy are output, ending the cycle. The method can be viewed as a steepest descent minimization, starting from a random sequence, and leading to a nearby, local, (folding) energy minimum. Below, we typically perform ∼100.000 heuristic cycles for each protein, thus sampling a large number of local minima on the energy surface.

2.4 Cost function network method

The CFN method is implemented in the Toulbar2 program [23, 24]. The Proteus energy matrices are converted to the Toulbar format with a perl script. With this format, all the interaction energies are approximated as positive integers, without loss of generality. We used Toulbar2 version 0.9.7.0 with a recommended parameterization (options -l=3 -m -d: -s); for the unsuccessful cases (GMEC not identified) we systematically repeated calculations with version 0.9.6.0 and a more recent and aggressive protocol (options -l=1 -dee=1 -m -d: -s). [details?] To enumerate sequence/conformation pairs that have energies higher than the GMEC, Toulbar2 is run with the "suboptimal" option and an energy

threshold. Available memory was limited to 30 gigabytes.

2.5 Energy function

The energy function has the form:

$$E = E_{\text{bonds}} + E_{\text{angles}} + E_{\text{dihe}} + E_{\text{impr}} + E_{\text{vdw}} + E_{\text{Coul}} + E_{\text{solv}} \tag{11}$$

The first six terms represent the protein internal energy. They were taken from the Charmm19 empirical energy function [40]. The last term represents the contribution of solvent. We used a "Coulomb + Accessible Surface Area", or CASA implicit solvent model [34, 41]:

$$E_{\text{solv}} = (\frac{1}{\epsilon} - 1)E_{\text{Coul}} + \sum_i \sigma_i A_i \tag{12}$$

Here, ϵ is a dielectric constant that scales the Coulomb energy; A_i is the solvent accessible surface area of atom i; σ_i is a parameter that reflects each atom's preference to be exposed or hidden from solvent. The solute atoms were divided into 4 groups with the following σ_i values (cal/mol/Å^2): unpolar (-5), aromatic (-40), polar (-8) and ionic (-10). Hydrogen atoms were assigned a surface coefficient of zero. Surface areas were computed by the Lee and Richards algorithm [42], using a 1.5 Å probe radius. Most of the calculations used a dielectric of $\epsilon = 16$ (see Results). Energy calculations are done with the Xplor program [43].

The energies $E_u(t)$ associated with the unfolded state were determined empirically to give reasonable amino acid compositions for the protein families considered here [35]; they are reported in Supplementary Material.

2.6 Test systems and preparation

We considered nine protein domains, from the SH3, SH2, and PDZ families, listed in Table 2. Each domain is known to fold stably and has an associated crystal structure used for our calculations. [details?]

2.7 Sequence characterization

Designed sequences were compared to the Pfam alignment for the corresponding family, using the Blosum40 scoring matrix and a gap penalty of -6. Each Pfam sequence was also

compared to its own Pfam alignment. For these Pfam/Pfam comparisons, if a test protein T was part of the Pfam alignment, the T/T self comparison was left out, to be more consistent with the designed/Pfam comparisons. If the test protein T was not part of the Pfam alignment, we used Blast to identify its closest Pfam homologue H and left the T/H comparison out, for consistency. More details on the SH3, SH2 and PDZ Pfam alignments are given in Results. Similarities were computed for protein core residues, defined by their near-complete burial, and listed in Results.

Designed sequences were submitted to the Superfamily library of Hidden Markov Models [38, 39], which attempts to classify sequences according to the SCOP classification [44]. Classification was based on SCOP version 1.75 and version 3.5 of the Superfamily tools. Superfamily executes the hmmscan program, which implements a Hidden Markov model for each SCOP family and superfamily; here hmmscan was executed with an E-value threshold of 10^{-10} and a maximum of 15438 family comparisons per query. [details?]

To compare the diversity in the designed sequences with the diversity in natural sequences, we used a standard, position-dependent sequence entropy [45], computed as follows:

$$S_i = -\sum_{i=1}^{6} f_j(i) ln f_j(i) \tag{13}$$

where $f_j(i)$ is the frequency of residue type j at position i, either in the designed sequences or in the natural sequences (organized into a multiple alignment). Instead of the usual, 20 amino acid types, we employ six residue types, corresponding to the following groups: {LVIMC}, {FYW}, {G}, {ASTP}, {EDNQ}, an {KRH}. This classification was obtained by a cluster analysis of the BLOSUM62 matrix [46], and also by analyzing residue-residue contact energies in proteins [47]. To get a sense of how many amino acid types appear at a specific position i, we usually report the residue entropy in its exponentiated form, $\exp(S_i)$, which ranges from 1 to 6.

3 Results

Our main focus is to characterize sequence/structure exploration methods and their ability to sample low energy sequences. We begin, however, by showing that the sequences we sample are similar to experimental ones. Indeed, the performance of exploration methods depends on the shape and ruggedness of the energy surface, and should be tested in sit-

uations where the energy function is sufficiently realistic, as judged by the quality of the designed sequences. After that, we compare the ability of the four exploration methods to identify low energy sequences, including the GMEC. Finally, we consider the diversity of sequence sets, or density of states sampled by each method.

3.1 Quality of the designed sequences

We first report information on the quality of our designed sequences. We use sets of REMC sequences to illustrate the main features. Results with the other exploration methods are expected to be similar. Indeed, while the methods sometimes exhibit differences of up to a few kcal/mol between their best sequences, the average sequence quality of the 100-1000 best sequences is similar between methods. Table 3 summarizes results for our 9 test proteins in design calculations where all positions were allowed to change types. All mutations were allowed, except mutations to/from Gly and Pro, since these are likely to change the backbone structure. REMC was done with 8 replicas at temperatures between 0.175 and 3 kT units. Simulation lengths were 750 million steps (per replica). The top 10000 sequence/conformation combinations were retained, corresponding to 200–400 unique sequences. For the 1A81 SH3 domain, none of these sequences was recognized by Superfamily as an SH3 family, or even superfamily member. For the other 8 proteins, all the retained sequences were recognized as members of the correct superfamily *and* family, with match lengths and E-values given in Table 3. Thus, our designed sequences are largely similar to experimental ones. Sequence identities to wildtype (including 1A81) are 31% on average (Table 3), similar to our earlier studies with the same energy function [13, 14]. Representative sequence logos for the protein core are shown in Fig. 1, illustrating the agreement between designed and experimental sequences. Similarity scores were computed between the designed sequences and experimental sequences from the Pfam database [48]. For the protein core region, the similarity is similar to that between experimental sequences, as shown in Fig. 2.

3.2 Finding the GMEC

3.2.1 CPU and memory limits for each method

The ability of an exploration method to sample low energy sequences depends on the CPU and memory ressources available, as well as on detailed parameterization choices. Here, we set somewhat arbitrary limits, to remain within a practical run situation. For CFN, we set a maximum time limit of 24 hours and a memory limit of 30 gbytes. For the heuristic method, we used 110,000 heuristic cycles, increased to 330,000 or 990,000 cycles in a few cases; even for these cases, run times did not exceed 24 hours. For MC, we ran up to 10^9 simulation steps, which corresponded to CPU times of 9 hours at most. For REMC, we ran 0.75 10^9 simulation steps per replica, with a few exceptions. We used an OpenMP, shared memory parallelization on a single processor, with one replica per core. Total CPU time per core was never more than 3 hours, for a total CPU use of less than 24 hours. For the heuristic, MC, and REMC methods, memory requirements are modest; about 2 gbytes for the largest calculations. Average run times are shown in Fig. 3 as a function of the number of designed positions, which varies from one position to the entire protein (about 90 positions). Results are averaged over the 9 test proteins; standard deviations between proteins are indicated and are small, except for CFN.

The MC and REMC methods require choices of move probabilities and temperatures, which affect the sampling in ways that vary slightly from protein to protein. Fig. 4 shows the lowest energy sampled with a small collection of protocols: one heuristic, one MC, and six REMC protocols, applied to our 9 proteins, with all positions allowed to mutate (except Gly/Pro). The lowest energy varies between protocols by up to 12 kcal/mol (compare the 1BM2 REMCa and REMCc energies or the 1CKA MC and REMCd energies). Based on these and other similar tests, we selected one specific protocol for each exploration method, which is generally good but not necessarily optimal for every situation.

3.2.2 Optimal sequences/structures with up to 10 designed positions

As a first series of tests, we did calculations for each test protein with zero, one, or five designed positions. Results are summarized in Fig. 5. With zero positions, only rotamers are optimized. With one, we systematically designed each position of each protein in turn. With 5, we picked the positions randomly, close together in the structure, in 5 different ways, for a total of 45 tests. In all these cases, CFN found the GMEC very rapidly

(seconds); the heuristic also found the GMEC, with longer run times (an hour). MC found the GMEC in all but a few cases (Fig. 5), with run times of a few minutes.

As a second series of tests, we chose randomly for each protein a set of ten positions to design; the other positions had fixed types but explored all possible rotamers. The selected positions were close by in the protein structure. For each protein, we made five separate choices of positions to design, for a total of 45 test cases. The CFN, heuristic, and MC methods were run for all 45 cases; REMC was run only when MC gave a poor result (6 cases, involving 5 proteins). Results are summarized in Fig. 5 and Table 4. 20 cases where all methods found the GMEC are not listed in the Table, leaving 25 where at least one method did not find the GMEC. CFN performed very well: only in one case did it not find the GMEC. The lowest energy was sampled in this case with the heuristic, and the best CFN energy was 5.7 kcal/mol higher (despite using the more aggressive CFN protocol).

The heuristic performed about as well as CFN for 10-position design. In one case, CFN did not find the GMEC and the heuristic gave the lowest energy (2BYG-1). In 39 cases, the heuristic found the GMEC. In 3 cases, it was within 0.15 kcal/mol of the GMEC, with no mutations (only rotamer differences). In one case (1CKA-5), it was 0.29 kcal/mol above the GMEC, with no mutations. Tripling the number of heuristic cycles allowed the GMEC to be reached (within 0.07 kcal/mol) in all these cases, with run times below 6 hours. There was only one real failure, 1M61-2, where the best heuristic solution was 3.51 kcal/mol above the GMEC, with 3 mutations relative to the GMEC. For this case, the GMEC was recovered (within 0.01 kcal/mol) if the number of cycles was increased to 990,000, for a run time of 7 hours. The heuristic structure (after 330,000 cycles) is compared to the GMEC in Fig. 6. Switching from one to the other requires concerted changes in 3 adjacent sidechain positions. This is only possible during a heuristic cycle if there is a downhill, connecting pathway made of single position changes, which is evidently very rare for this particular test. Thus, the heuristic method can only find the GMEC if it draws the right combination of types/rotamers at the very beginning of a cycle; hence the need for 990,000 cycles.

Plain MC did slightly less well for 10-position design. In 21 cases, it found the GMEC. In 18 cases, its best sequence was within 0.2 kcal/mol of the GMEC, with 0–3 mutations (one on average). Notice that 0.2 kcal/mol is the thermal energy for the MC protocol employed. In 6 cases, its best sequence was between 0.9 and 4.5 kcal/mol above the GMEC, with 2–7 mutations (3 on average). For these 6 cases, REMC was run, and

sampled sequences within 0.40 kcal/mol of the GMEC, except for one case where its best sequence was 0.80 kcal/mol above the GMEC. Overall, MC or REMC reached the GMEC to within 0.40 kcal/mol in all but one case. A 0.40 kcal/mol energy difference is actually less than the average pairwise additivity errors in the energy function [34, 41, 49], and so one might consider this performance to be about as good as the CFN and heuristic methods. In terms of speed, for 10-position design, all the methods were comparable (a few hours per run on average).

3.2.3 Optimal sequences with 20 or 30 designed positions

We did similar tests with 20 designed positions, selected randomly in 5 different ways for each protein, as above. Results are given in Fig. 5 and Table 5. CFN found the GMEC in 28 out of 45 cases; in 2 others, it found the best energy of the 4 methods. For 7 of these 30 cases, the more aggressive protocol was necessary, and run times were about xxx hours on average. For 14 of the other 15 cases, the best CFN energy was 0.1–7.5 kcal/mol above the best solution found by the other methods, and 2.8 kcal/mol on average, despite using the more aggressive protocol. For the worst case, the CFN energy was 13.9 kcal/mol above the best solution.

The heuristic method found the GMEC in 22 of the 28 cases where it is known. For the other 6 cases, it was within 0.40 kcal/mol of the GMEC, with 0–4 mutations (2.7 on average). For the 17 cases where the GMEC was not identified by CFN, the heuristic produced the lowest energy of all methods, except one case (104C-1) where it was 0.35 kcal/mol above CFN. Overall, the heuristic either found the best energy of the four methods or was within 0.4 kcal/mol of the best energy.

MC converged to the best energy in 11 cases; in 25 other cases, it was within 0.50 kcal/mol of the best energy. In the other 9 cases, its best energy was at most 3.2 kcal/mol above the best energy (sampled by the heuristic and/or CFN). Finally, REMC was done for all the test cases. In 6 cases, its best energy was more than 0.50 kcal/mol from the best energy. However, the differences were notably smaller than for plain MC, with an average of just 0.8 kcal/mol for the 6 worst cases and a maximum (for 1G90-1) of 1.25 kcal/mol.

The same tests were done with 30 designed positions; see Fig. 5 and Table 5. CFN found the GMEC in just one case; in 5 others, it did not find the GMEC but gave the lowest energy overall. In 4 other cases, it was within 0.50 kcal/mol of the best energy sampled by the other methods. For the other 35 cases, its best energy was higher than the

best method, with differences of 10 kcal/mol or more in 20 cases.

The heuristic produced the lowest energy in all but 4 cases, with differences in those cases of 0.01, 0.10, 0.70, and 1.69 kcal/mol from the best energy. In the last two cases, CFN produced the best energy. Plain MC found the best energy in only 12 cases, but gave only moderate energy errors: in just 4 cases was its best sequence more than 2 kcal/mol above the overall best energy (differences of 2.2, 2.5, 2.8, and 7.7 kcal/mol). REMC was applied to the 13 cases where the MC errors were largest; in 4 of these it reduced the error to 0.6 kcal/mol or less. The largest MC error was reduced from 7.7 to 2.4 kcal/mol. Doubling the REMC trajectory length reduced the two largest remaining errors to 1.1 and 1.8 kcal/mol.

3.3 Density of states above the GMEC

The exact CFN method can enumerate exhaustively sequence/conformation states above the GMEC, up to a given energy threshold, if the threshold is not too large. Monte Carlo and REMC explore states randomly, within a typical energy range that depends on temperature. To characterize the diversity of the sequence ensembles, we focus on the CFN and REMC methods, and we consider both the sequence entropy and the total number of states.

The mean sequence entropies are reported in Table 6. The sequence entropy for the Pfam alignment is also shown. The entropy $S(E)$ is shown in Fig. 7 as a function of the energy threshold E. Exact CFN results are compared to REMC. Complete enumeration was only feasible up to an energy threshold of $E = 2$ kcal/mol (above the GMEC). REMC samples energies up to about 14 kcal/mol above the GMEC. As we consider higher energy threshold values, the entropy increases in a quasilinear way: the kT=0.592 kcal/mol replica samples the 4–10 kcal/mol range; the kT=0.888 kcal/mol replica samples the 11–14 kcal/mol range. In the CFN range, the two methods agree very well; REMC has effectively sampled sequences whose diversity, measured by $S(E)$, is about the same as the full CFN ensemble. In contrast, the number of sequences visited by REMC is much lower. Thus, while REMC samples the full diversity at each individual position, it does not sample exhaustively all the combinations of mutations (let alone rotamers).

4 Discussion

Acknowledgements

We thank Seydou Traoré for help with the Toulbar2 program and Georgios Archontis, Isabelle André, and Sophie Barbe for helpful discussions.

References

[1] Dantas, G., Kuhlman, B., Callender, D., Wong, M., and Baker, D. A large test of computational protein design: Folding and stability of nine completely redesigned globular proteins. *J. Mol. Biol. 332* (2003), 449–460.

[2] Butterfoss, G. L., and Kuhlman, B. Computer-based design of novel protein structures. *Ann. Rev. Biophys. Biomolec. Struct. 35* (2006), 49–65.

[3] Lippow, S. M., and Tidor, B. Progress in computational protein design. *Curr. Opin. Biotech. 18* (2007), 305–311.

[4] Saven, J. G. Computational protein design: engineering molecular diversity, nonnatural enzymes, nonbiological cofactor complexes, and membrane proteins. *Curr. Opin. Chem. Biol. 15* (2011), 452–457.

[5] Feldmeier, K., and Hoecker, B. Computational protein design of ligand binding and catalysis. *Curr. Opin. Chem. Biol. 17* (2013), 929–933.

[6] Tinberg, C. E., Khare, S. D., Dou, J., Doyle, L., Nelson, J. W., Schena, A., Jankowski, W., Kalodimos, C. G., Johnsson, K., Stoddard, B. L., and Baker, D. Computational design of ligand-binding proteins with high affinity and selectivity. *Nature 501* (2013), 212–218.

[7] Pokala, N., and Handel, T. Energy functions for protein design I: Efficient and accurate continuum electrostatics and solvation. *Prot. Sci. 13* (2004), 925–936.

[8] Samish, I., MacDermaid, C. M., Perez-Aguilar, J. M., and Saven, J. G. Theoretical and computational protein design. *Ann. Rev. Phys. Chem. 62* (2011), 129—149.

[9] Li, Z., Yang, Y., Zhan, J., Dai, L., and Zhou, Y. Energy functions in de novo protein design: Current challenges and future prospects. *Ann. Rev. Biochem 42* (2013), 315–335.

[10] Ponder, J., and Richards, F. M. Tertiary templates for proteins: Use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol. 193* (1988), 775–791.

[11] Koehl, P., and Levitt, M. Protein topology and stability define the space of allowed sequences. *Proc. Natl. Acad. Sci. USA 99* (2002), 1280–1285.

[12] Larson, S. M., England, J. E., Desjarlais, J. R., and Pande, V. S. Thoroughly sampling sequence space: Large-scale protein design of structural ensembles. *Prot. Sci. 11* (2002), 2804–2813.

[13] Schmidt am Busch, M., Mignon, D., and Simonson, T. Computational protein design as a tool for fold recognition. *Proteins 77* (2009), 139–158.

[14] Schmidt am Busch, M., Sedano, A., and Simonson, T. Computational protein design: validation and possible relevance as a tool for homology searching and fold recognition. *PLoS One 5(5)* (2010), e10410.

[15] Druart, K., Palmai, Z., Omarjee, E., and Simonson, T. Protein:ligand binding free energies: a stringent test for computational protein design. *J. Comput. Chem. in press* (2015), 0000.

[16] Aleksandrov, A., Polydorides, S., Archontis, G., and Simonson, T. Predicting the acid/base behavior of proteins: A constant-pH Monte Carlo approach with Generalized Born solvent. *J. Phys. Chem. B 114* (2010), 10634–10648.

[17] Polydorides, S., and Simonson, T. Monte Carlo simulations of proteins at constant pH with generalized Born solvent, flexible sidechains, and an effective dielectric boundary. *J. Comput. Chem. 34* (2013), 2742–2756.

[18] Kilambi, K., and Gray, J. J. Rapid calculation of protein pK_a values using Rosetta. *Biophys. J. 103* (2012), 587–595.

[19] Looger, L. L., and Hellinga, H. W. Generalized dead-end elimination algorithms make large-scale protein sidechain structure prediction tractable: Implications for protein design and structural genomics. *J. Mol. Biol. 307* (2001), 429–445.

[20] Georgiev, I., Lilien, R. H., and Donald, B. R. The minimized dead-end elimination criterion and its application to protein redesign in a hybrid scoring and search algorithm

for computing partition functions over molecular ensembles. *J. Comput. Chem. 29* (2008), 1527–1542.

[21] Gordon, D. B., and Mayo, S. L. *Structure 7* (1999), 1089.

[22] Hong, E. J., Lippow, S. M., Tidor, B., and Lozano-Perez, T. Rotamer optimization for protein design through MAP estimation and problem size reduction. *J. Comput. Chem. 30* (2009), 1923–1945.

[23] Traore, S., Allouche, D., Andr'e, I., de Givry, S., Katsirelos, G., Schiex, T., and Barbe, S. A new framework for computational protein design through cost function network optimization. *Bioinformatics 29* (2013), 2129–2136.

[24] Allouche, D., Traore, S., Andr'e, I., Barbe, S., de Givry, S., Katsirelos, G., and Schiex, T. Computational protein design as an optimization problem. *Artif. Intell. 212* (2014), 59–79.

[25] Zou, J., and Saven, J. G. Using self-consistent fields to bias Monte Carlo methods with applications to designing and sampling protein sequences. *J. Chem. Phys. 118* (2003), 3843–3854.

[26] Frenkel, D., and Smit, B. *Understanding molecular simulation.* Academic Press, New York, 1996.

[27] Chipot, C., and Pohorille, A. *Free energy calculations: theory and applications in chemistry and biology.* Springer Verlag, N.Y., 2007.

[28] Wernisch, L., Héry, S., and Wodak, S. Automatic protein design with all atom force fields by exact and heuristic optimization. *J. Mol. Biol. 301* (2000), 713–736.

[29] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys. 21* (1953), 1087–1092.

[30] Grimmet, G. R., and Stirzaker, D. R. *Probability and random processes.* Oxford University Press, 2001.

[31] Sugita, Y., and Okamoto, Y. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett. 314* (1999), 141–151.

[32] Kofke, D. A. On the acceptance probability of replica-exchange Monte Carlo trials. *J. Chem. Phys. 117* (2002), 6911–6914.

[33] Earl, D., and Deem, M. W. Parallel tempering: theory, applications, and new perspectives. *Phys. Chem. Chem. Phys. 7* (2005), 3910–3916.

[34] Schmidt am Busch, M., Lopes, A., Amara, N., Bathelt, C., and Simonson, T. Testing the Coulomb/Accessible Surface Area solvent model for protein stability, ligand binding, and protein design. *BMC Bioinformatics 9* (2008), 148–163.

[35] Schmidt am Busch, M., Lopes, A., Mignon, D., and Simonson, T. Computational protein design: software implementation, parameter optimization, and performance of a simple model. *J. Comput. Chem. 29* (2008), 1092–1102.

[36] Simonson, T., Gaillard, T., Mignon, D., Schmidt am Busch, M., Lopes, A., Amara, N., Polydorides, S., Sedano, A., Druart, K., and Archontis, G. Computational protein design: the Proteus software and selected applications. *J. Comput. Chem. 34* (2013), 2472–2484.

[37] Dahiyat, B. I., and Mayo, S. L. De novo protein design: fully automated sequence selection. *Science 278* (1997), 82–87.

[38] Gough, J., Karplus, K., Hughey, R., and Chothia, C. Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. *J. Mol. Biol. 313* (2001), 903–919.

[39] Wilson, D., Madera, M., Vogel, C., Chothia, C., and Gough, J. The SUPERFAMILY database in 2007: families and functions. *Nucl. Acids Res. 35* (2007), D308—D313.

[40] Brooks, B., Brooks III, C. L., Mackerell Jr., A. D., Nilsson, L., Petrella, R. J., Roux, B., Won, Y., Archontis, G., Bartels, C., Boresch, S., Caflisch, A., Caves, L., Cui, Q., Dinner, A. R., Feig, M., Fischer, S., Gao, J., Hodoscek, M., Im, W., Kuczera, K., Lazaridis, T., Ma, J., Ovchinnikov, V., Paci, E., Pastor, R. W., Post, C. B., Pu, J. Z., Schaefer, M., Tidor, B., Venable, R. M., Woodcock, H. L., Wu, X., Yang, W., York, D. M., and Karplus, M. CHARMM: The biomolecular simulation program. *J. Comp. Chem. 30* (2009), 1545–1614.

[41] Lopes, A., Aleksandrov, A., Bathelt, C., Archontis, G., and Simonson, T. Computational sidechain placement and protein mutagenesis with implicit solvent models. *Proteins 67* (2007), 853–867.

[42] Lee, B., and Richards, F. The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol. 55* (1971), 379–400.

[43] Brünger, A. T. X-plor *version 3.1, A System for X-ray crystallography and NMR*. Yale University Press, New Haven, 1992.

[44] Andreeva, A., Howorth, D., Brenner, S. E., Hubbard, J. J., Chothia, C., and Murzin, A. G. SCOP database in 2004: refinements integrate structure and sequence family data. *Nucl. Acids Res. 32* (2004), D226–229.

[45] Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. *Biological sequence analysis*. Cambridge University Press, 2002.

[46] Murphy, L. R., Wallqvist, A., and Levy, R. M. Simplified amino acid alphabets for protein fold recognition and implications for folding. *Prot. Eng. 13* (2000), 149–152.

[47] Launay, G., Mendez, R., Wodak, S. J., and Simonson, T. Recognizing protein-protein interfaces with empirical potentials and reduced amino acid alphabets. *BMC Bioinf. 8* (2007), 270–291.

[48] Finn, R. D., Mistry, J., Schuster-Böckler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., Eddy, S. R., Sonnhammer, E. L. L., and Bateman, A. Pfam: clans, web tools and services. *Nucl. Acids Res. 34* (2006), D247–251.

[49] Gaillard, T., and Simonson, T. Pairwise decomposition of an MMGBSA energy function for computational protein design. *J. Comput. Chem. 35* (2014), 1371–1387.

Table 1: Representative MC protocols

name	kT	length/10^6	number	threshold	Probas
mc2	0.2	3	1000	10	0; 1; 0.1; 0 ;0
MC0	0.01	1000	1	10	1; 0; 0.1; 0 ;0
MCa	0.2	6000	1	10	1; 0; 0.1; 0 ;0
MCa-	0.2	1000	1	10	1; 0; 0.1; 0 ;0
MCb	0.2	6000	1	10	0; 1; 0.1; 0 ;0

Table 2: Test proteins

type	PDB	length	acronym	type	PDB	length	acronym
PDZ	1G9O	91	NHERF	SH2	1A81	108	Syk kinase
PDZ	1R6J	82	syntenin	SH2	1BM2	98	Grb2
PDZ	2BYG	97	DLH2	SH2	1M61	109	Zap70
SH3	1ABO	58	Abl	SH2	1O4C	104	Src kinase
SH3	1CKA	57	c-Crk				

Table 3: Designed sequence quality measures

Protein	Number of sequences	Match length	Superfamily E-value	Superfamily success rate	Family E-value	Family success rate	Identity % to wildtype
1A81	236	none					27
1ABO	203	51/58	4.4e-4	100%	2.8e-3	100%	32
1BM2	209	78/98	4.2e-5	100%	2.6e-3	100%	27
1CKA	416	40/57	1.1e-5	100%	3.4e-3	100%	33
1G9O	338	79/91	7.0e-7	100%	2.5e-3	100%	36
1M61	405	97/109	7.2e-7	100%	2.6e-4	100%	42
1O4C	274	95/104	2.1e-4	100%	4.5e-3	100%	21
1R6J	270	74/82	9.8e-6	100%	4.6e-3	100%	34
2BYG	426	59/97	1.4e-5	100%	7.1e-3	100%	28

Table 4: Tests with 10 designed positions

rotamers	length	Protein	CFN	Heur.	MC	REMC
3187	108	1A81 3	gmec	0.001	0.1595	
		1A81 4	gmec	0.	0.0317	
		1A81 5	gmec	0.	0.0563	
2612	58	1ABO 1	gmec	0.0675	0.9054	0.8041
		1ABO 4	gmec	0.	0.0128	
3072	98	1BM2 1	gmec	0.	0.0950	
		1BM2 5	gmec	0.	0.1082	
2600	57	1CKA 5	gmec	0.2859	3.2525	0.
2991	91	1G9O 3	gmec	0.1366	0.1366	
		1G9O 5	gmec	0.	3.9599	0.
3198	109	1M61 1	gmec	0.	0.0776	
		1M61 2	gmec	3.5105	4.5062	0.3215
		1M61 5	gmec	0.	0.0432	
3141	104	1O4C 1	gmec	0.	0.1121	
		1O4C 2	gmec	0.	0.1046	
		1O4C 3	gmec	0.	0.1519	
		1O4C 4	gmec	0.	0.1545	
		1O4C 5	gmec	0.	0.1753	
2888	82	1R6J 1	gmec	0.	2.4022	0.3986
		1R6J 2	gmec	0.	1.0398	0.3049
		1R6J 3	gmec	0.	0.0106	
		1R6J 5	gmec	0.	0.0162	
3060	97	2BYG 1	5.7485	0.	0.0337	
		2BYG 3	gmec	0.	0.0833	
		2BYG 4	gmec	0.	0.2149	

Table 5: Tests with 20 and 30 designed positions

	20 positions					30 positions			
Protein	CFN	Heur.	MC	REMC	mutations	CFN	Heur.	MC	REMC
1A81 1	gmec*	0.	0.3275	0.3851	0	1.2074	0.	0.6353	
1A81 2	gmec*	0.1705	2.4355	1.0069	3	2.5520	0.	0.0578	
1A81 3	gmec	0.	0.4640	0.6186	0	43.5263	0.	2.4996	1.2025
1A81 4	gmec	0.3878	0.5748	0.6991	4	5.1300	0.	0.0305	
1A81 5	gmec	0.0068	0.5088	0.1541	4	3.2417	0.	1.9586	0.5791
1ABO 1	gmec	0.1205	1.1159	0.2153	2	44.5504	0.	0.	
1ABO 2	13.8563	0.	0.	0.	8	12.7303	0.	0.	
1ABO 3	1.2190	0.	0.	0.	9	9.3870	0.	0.2630	
1ABO 4	1.9940	0.	0.0076	0.	5	10.7691	0.	0.	
1ABO 5	3.5418	0.	0.9483	0.9483	9	4.3907	0.	0.	
1BM2 1	gmec	0.	0.0619	0.1584	0	22.5876	0.	1.7290	1.6013
1BM2 2	7.5304	0.	0.0725	0.0143	8	22.1386	0.	1.9856	1.5876
1BM2 3	gmec	0.0229	0.4762	0.2897	0	22.5410	0.	1.9990	1.1541
1BM2 4	0.1186	0.	2.5883	0.0789	2	15.2639	0.	2.2127	2.3854
1BM2 5	gmec	0.2396	0.3746	0.3746	3	15.9890	0.	2.8354	1.1937
1CKA 1	gmec*	0.	0.	0.	0	6.2700	0.	0.	
1CKA 2	gmec	0.	0.	0.	0	2.0995	0.	0.	
1CKA 3	gmec	0.	0.	0.	0	47.0217	0.	0.	
1CKA 4	4.3122	0.	0.	0.	4	44.0830	0.	0.	
1CKA 5	4.2849	0.	0.	0.	3	8.8608	0.	0.	
1G9O 1	2.0574	0.	1.2525	1.2525	5	2.0816	0.	1.5942	0.
1G9O 2	3.2106	0.	0.2177	0.1915	1	0.3270	0.	0.3126	
1G9O 3	1.9008	0.	0.4417	0.1019	1	17.7150	0.	1.5667	1.5667
1G9O 4	0.5030	0.	0.3855	0.1455	5	2.9758	0.	1.4284	1.6202
1G9O 5	0.4298	0.	0.1495	0.5114	5	0.	1.6890	7.6985	2.3857
1M61 1	gmec	0.	0.	0.	0	14.4935	0.0097	0.	0.
1M61 2	gmec	0.	0.	0.	0	5.0899	0.	1.8749	0.008
1M61 3	gmec	0.	0.	0.	0	3.5795	0.	0.0154	
1M61 4	gmec	0.	0.	0.	0	16.1511	0.	0.	
1M61 5	gmec	0.	0.2521	0.1345	0	23.0927	0.	0.	
1O4C 1	0.	0.3465	0.0690	0.0587	6	14.9064	0.	0.3435	
1O4C 2	6.4214	0.	0.1963	0.3175	4	58.1558	0.	0.0795	
1O4C 3	gmec	0.	0.3461	0.0997	0	9.9221	0.	0.1789	
1O4C 4	gmec	0.	0.3640	0.1382	0	5.7790	0.	0.0423	
1O4C 5	0.	0.	0.1131	0.2206	0	9.9221	0.	0.1789	
1R6J 1	gmec	0.	0.2604	0.2002	0	gmec*	0.	0.0246	
1R6J 2	gmec	0.	0.0071	0.0183	0	14.9800	0.	0.0957	
1R6J 3	gmec	0.	0.0537	0.0732	0	0.	0.	0.0440	
1R6J 4	gmec	0.	0.0639	0.0601	0	0.	0.	0.0957	
1R6J 5	gmec	0.	0.0735	0.0244	0	0.	0.7036	1.8823	0.0781
2BYG 1	gmec	0.	3.1878	0.0257	0	17.9752	0.	0.1592	
2BYG 2	gmec	0.	0.0524	0.0831	0	0.3832	0.	0.1502	
2BYG 3	gmec*	0.	1.3564	0.0826	0	0.1442	0.	0.1593	
2BYG 4	gmec	0.	0.1968	0.6022	0	0.	0.0958	0.0050	
2BYG 5	1.8604	0.	0.0933	0.0386	2	0.5003	0.	0.6876	

Table 6: Designed and Pfam sequence entropies

Protein	Top 10,000 structures	Top 10,000 sequences	Pfam seed	Pfam full
1A81	1.13		2.91	3.51
1ABO	1.36		2.79	3.01
1BM2	1.08		2.90	3.50
1CKA	1.20		2.84	3.03
1G9O	1.21		3.29	3.81
1M61	1.31		2.91	3.51
1O4C	1.36		2.94	3.47
1R6J	1.33		3.11	3.66
2BYG	1.57		3.31	3.67

Figure captions

1. Sequence logos for core region of two designed proteins.
2. Histogram of Blosum40 similarity scores to Pfam sequences.
3. Run times for different test calculations and search methods.
4. Comparison between selected heuristic, MC, and REMC protocols for whole protein design.
5. Lowest energies obtained with different protocols for all test calculations.
6. 3D structure of GMEC and best heuristic solution for the 1M61 protein, 1M61-2 test with ten designned positions.
7. Sequence entropy and number of states with CFN and REMC.