PLOS | COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

# A critical analysis of computational protein design with sparse residue interaction graphs

**Swati Jain**[1,2,3☯¤a], **Jonathan D. Jou**[2☯], **Ivelin S. Georgiev**[2¤b], **Bruce R. Donald**[2,3,4]*

**1** Computational Biology and Bioinformatics Program, Duke University, Durham, North Carolina, United States of America, **2** Department of Computer Science, Duke University, Durham, North Carolina, United States of America, **3** Department of Biochemistry, Duke University Medical Center, Durham, North Carolina, United States of America, **4** Department of Chemistry, Duke University, Durham, North Carolina, United States of America

☯ These authors contributed equally to this work.
¤a Current address: Department of Chemistry, New York University, New York, New York, United States of America
¤b Current address: Vanderbilt University School of Medicine, Nashville, Tennessee, United States of America
* brd+pcb15@cs.duke.edu





**Citation:** Jain S, Jou JD, Georgiev IS, Donald BR (2017) A critical analysis of computational protein design with sparse residue interaction graphs. PLoS Comput Biol 13(3): e1005346. https://doi.org/10.1371/journal.pcbi.1005346

**Editor:** Patrick Aloy, Institute for Research in Biomedicine, SPAIN

**Received:** December 2, 2015

**Accepted:** January 3, 2017

**Published:** March 30, 2017



**Data Availability Statement:** Code: The implementation of Sparse A* in OSPREY as used in this manuscript is available on GitHub: https://github.com/donaldlab/OSPREY_SparseAStar. We recommend that this be used to test the hypotheses and results of sparse residue interaction graphs as presented in this manuscript. In addition, our lab has also developed a novel and more efficient dynamic programming algorithm called Branch-Width Minimization* (BWM*) available at https://github.com/donaldlab/BWM for protein design with sparse residue interaction graphs. For new empirical designs, we recommend

## Abstract

Protein design algorithms enumerate a combinatorial number of candidate structures to compute the Global Minimum Energy Conformation (GMEC). To efficiently find the GMEC, protein design algorithms must methodically reduce the conformational search space. By applying distance and energy cutoffs, the protein system to be designed can thus be represented using a *sparse residue interaction graph*, where the number of interacting residue pairs is less than all pairs of mutable residues, and the corresponding GMEC is called the *sparse GMEC*. However, ignoring some pairwise residue interactions can lead to a change in the energy, conformation, or sequence of the sparse GMEC vs. the original or the *full GMEC*. Despite the widespread use of sparse residue interaction graphs in protein design, the above mentioned effects of their use have not been previously analyzed. To analyze the costs and benefits of designing with sparse residue interaction graphs, we computed the GMECs for 136 different protein design problems both with and without distance and energy cutoffs, and compared their energies, conformations, and sequences. Our analysis shows that the differences between the GMECs depend critically on whether or not the design includes core, boundary, or surface residues. Moreover, neglecting long-range interactions can alter local interactions and introduce large sequence differences, both of which can result in significant structural and functional changes. Designs on proteins with experimentally measured thermostability show it is beneficial to compute both the full and the sparse GMEC accurately and efficiently. To this end, we show that a provable, ensemble-based algorithm can efficiently compute both GMECs by enumerating a small number of conformations, usually fewer than 1000. This provides a novel way to combine sparse residue interaction graphs with provable, ensemble-based algorithms to reap the benefits of sparse residue interaction graphs while avoiding their potential inaccuracies.

using the latest version of OSPREY available at http://www.cs.duke.edu/donaldlab/osprey.php that includes continuous sidechain and backbone flexibility, ensemble-based design, and new and faster methods to calculate energy functions. Data: All the results on the 136 protein design problems and 6 retrospective protein design problems discussed in this manuscript are available from the Harvard Dataverse repository (https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/6C6IWA, doi:10.7910/DVN/6C6IWA).

**Funding:** This work is primarily supported by the following grant from the National Institutes of Health (NIH): R01 GM-78031 to BRD. In addition, SJ was supported in part by NIH grants R01 GM-73919 and R01 GM-73930 to Dr. David C. Richardson. Website: www.nih.gov. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** The authors have declared that no competing interests exist.

## Author summary

Computational structure-based protein design algorithms have successfully redesigned proteins to fold and bind target substrates *in vitro*, and even *in vivo*. Because the complexity of a computational design increases dramatically with the number of mutable residues, many design algorithms employ cutoffs (distance or energy) to neglect some pairwise residue interactions, thereby reducing the effective search space and computational cost. However, the energies neglected by such cutoffs can add up, which may have nontrivial effects on the designed sequence and its function. To study the effects of using cutoffs on protein design, we computed the optimal sequence both with and without cutoffs, and showed that neglecting long-range interactions can significantly change the computed conformation and sequence. Designs on proteins with experimentally measured thermostability showed the benefits of computing the optimal sequences (and their conformations), both with and without cutoffs, efficiently and accurately. Therefore, we also showed that a provable, ensemble-based algorithm can efficiently compute the optimal conformation and sequence, both with and without applying cutoffs, by enumerating a small number of conformations, usually fewer than 1000. This provides a novel way to combine cutoffs with provable, ensemble-based algorithms to reap the computational efficiency of cutoffs while avoiding their potential inaccuracies.

## Introduction

Computational structure-based protein design is an emerging field with many applications in basic science and biomedical research [1]. Protein sequences have been designed to fold to specific tertiary structures [2–5]. Novel biological functions have been achieved by constructing new ligand binding sites and by switching binding specificities of enzymes [6–13]. New drugs, antibodies and nanobodies have been developed for therapeutic purposes by designing protein-protein and protein-ligand interfaces [14–19]. Negative design has also been used for predicting resistance mutations for highly drug-resistant pathogens [20, 21]. The above mentioned studies are examples of the predictive power of computational protein design algorithms.

A major challenge for protein design algorithms is to efficiently explore and evaluate a vast sequence and conformational search space, which increases exponentially with the number of design positions and mutations allowed. Most design algorithms use a rigid backbone and model side-chain flexibility using a discrete set of frequently-observed, low-energy conformations called *rotamers* [22]. However, even with a pairwise energy function, rigid backbone and rotamer libraries, identification of the Global Minimum Energy Conformation (GMEC) is NP-hard [23, 24]. Moreover, modeling of additional protein flexibility (both in the backbone and side chains) for realistic biological representation increases the search space [1–3, 6, 9, 10, 12, 25–33], which in turn results in increased runtime for protein design algorithms. Due to increased complexity, numerous heuristic techniques have been used to find a locally optimal solution and generate solutions quickly [6, 7, 25–27, 34–38]. Provable algorithms, on the other hand, guarantee the quality of the solutions found relative to the input model, and can generate a gap-free list of low-energy conformations within a given energy window of the GMEC [39]. One example is dead-end elimination (DEE) [28, 30–32, 40] followed by A* search [29, 41, 42], which has been used to approximate the thermodynamic ensemble and approximate the binding constant $K_a$ [11, 43, 44]. However, in general provable algorithms require additional time and memory. With limited resources, it is important for design algorithms to systematically reduce the search space, without compromising on the quality and accuracy of design

predictions. One way to do this is to use sparse residue interaction graphs for protein design, as described below.

## Sparse residue interaction graphs and protein design

Frequently, the goal of a protein design problem is to find the lowest energy sequences or conformations. Most protein design algorithms use pairwise energy functions to score protein conformations. Any such protein design problem can be represented by a *residue interaction graph*, where the nodes represent residues, and edges represent the interaction between residues. The energy functions usually consist of distance-dependent terms to model van der Waals and electrostatic interactions between residue pairs, and the interaction energy decreases with increasing distance between residues. Therefore, it is possible to neglect interaction energies between distant residues and not add them to the overall energy of a conformation. This eliminates edges between these negligibly-interacting residues from the residue interaction graph to construct a *sparse residue interaction graph*, with the corresponding GMEC called the *sparse GMEC*. We will refer to the residue interaction graph with no edges eliminated as the *full* residue interaction graph, and the corresponding GMEC as the *full GMEC*.

Whether explicitly described or not, the concept of sparse residue interaction graphs is ubiquitous in the field of protein design. Many design algorithms apply appropriate distance cutoffs implicitly in the energy function (using different cutoffs for different kinds of energies calculated) [24, 45–52], while others develop new algorithms to take explicit advantage of the sparseness of the residue interaction graph [53–60]. By using sparse residue interaction graphs, the number of interacting residue pairs is fewer than all pairs of mutable residues, and this reduces the effective search space considerably. However, the energies omitted by deleting the edges between negligibly interacting residues can add up, causing differences in sequence (such as amino acid identity) of the GMEC returned or the rankings of the top sequences (specific examples are discussed in detail in the Sections entitled "Results" and "Discussion"). While small sequence differences might not be consequential, larger differences in energies and sequences can lead to design algorithms returning a protein sequence that may not have the desired function. Therefore there is a potential tradeoff between reducing the search space and guaranteeing the accuracy and quality of the computed GMEC. Despite the widespread use of sparse residue interaction graphs in protein design, the effects of this tradeoff have not been previously analyzed.

In this paper, we present the results of our analysis of using sparse residue interaction graphs in protein design. We implemented a variation of the A* search algorithm in the protein design software OSPREY [43], for design with sparse residue interaction graphs to return the corresponding GMEC. OSPREY has been used in many successful designs *in vitro* [11, 13–15, 17, 18, 20], and even *in vivo* [14, 15, 18, 20]. We ran computational experiments on a total of 136 protein design problems, involving core, boundary, and surface residues. We used different energy and distance cutoffs to generate the sparse residue interaction graph, and analyzed the sequence and energy differences between the different GMECs returned. Our results show that commonly used distance cutoffs can return a GMEC whose sequence is different than that of the GMEC returned without those cutoffs. The underlying assumption when using distance and energy cutoffs is that neglecting long-range interactions do not have an effect on local interactions. We show that, contrary to this assumption, neglecting long-range interactions can alter favorable local interactions. Changes to the sequence and loss of favorable interactions between residues can both result in structural and functional changes to the predicted protein.

Next, in order to study if the sequence differences between the full and the sparse GMEC lead to functional differences, we performed retrospective validation on 6 protein design

problems for which experimentally determined thermal stability data was available, and analyzed the sequences differences between the GMECs returned with and without distance cutoffs. Our analysis shows that across all 6 design problems, the sparse and full GMEC predicted different amino acid identities at 13 residues. Out of these 13 residues that have a different amino acid identity in the two GMECs, the more thermostabilizing mutation is predicted by the GMEC of the sparse residue interaction graph for 7 residues, and by the GMEC of the full residue interaction graph (without using distance cutoffs) for the remaining 6 residues. This indicates that there is no clear trend on which of the two GMECs will predict mutations with the desired function *in vitro*. Moreover, it can be difficult to correctly choose between the GMEC of the full residue interaction graph and its less computationally expensive sparse equivalent. Therefore it is beneficial to compute the GMECs for *both* the full and the sparse residue interaction graph, and to do so efficiently, while still taking advantage of the computational benefits of the reduced search space induced by the sparse residue interaction graph.

To achieve this goal, we provide a novel approach, called *Energy-bounding enumeration*, to combine sparse residue interaction graphs with provable, ensemble-based algorithms to generate both the GMECs efficiently. The gap-free list of low-energy conformations returned by an ensemble-based provable algorithm is guaranteed to contain the GMEC for the full residue interaction graph [59]. From this list, we prove that this GMEC can be found in additional $O(kn^2)$ time, where $n$ is the number of mutable residues, and $k$ is the number of conformations generated. We show that in practice, the full GMEC is almost always found within the first 1000 conformations returned. Because the number of conformations required to capture the GMEC is usually small, protein designers can henceforth combine sparse residue interaction graphs with provable, ensemble-based algorithms to exploit the reduced search space and still compute the GMECs for both the full and the sparse residue interaction graph. In short: sparse residue interaction graphs induce substantial differences in predicted sequences, conformations, and energies, with no way of telling which model will best predict the desired function. But provable, ensemble-based algorithms rescue computational protein design from these difficulties by providing a way to compute both GMECs efficiently.

In particular, this paper makes the following contributions:

1. Implementation of a variation of the A* search algorithm in the open-source protein design package OSPREY [11, 15–18, 20, 43] for protein design with sparse residue interaction graphs, and proof of the asymptotic time complexity to enumerate the GMEC from the gap-free list of conformations enumerated by this variant of A*.
2. Results showing that commonly used distance cutoffs can introduce large energy, conformation, and even sequence changes in the GMEC.
3. Examples showing that neglecting long-range interactions can alter local interactions, and an analysis of the sequence changes introduced when using sparse vs. full residue interaction graphs.
4. Retrospective designs and analysis of 6 protein design problems with experimentally measured thermostability drawn from the literature, emphasizing the benefits of computing the GMEC of both the full and the sparse residue interaction graph efficiently.
5. A novel approach, called *Energy-bounding enumeration*, to compute the GMEC of both the full and the sparse residue interaction graph in the gap-free list of conformations enumerated using the sparse residue interaction graph, and showing that the GMEC of the full residue interaction graph is usually found within the first 1000 conformations returned.

6. Examples of protein design problems in which A* with the full interaction graph failed to compute the GMEC, but using sparse residue interaction graphs allowed A* to compute the corresponding GMEC and also enumerate a gap-free list of conformations which is likely to contain the GMEC of the full graph.

## Materials and methods

### Definitions related to sparse residue interaction graphs

Each protein design problem is defined by its *input model*, namely, the input protein structure, the mutable residues, the allowed amino acids at each mutable residue, allowed side-chain conformations, and energy function.

Given this input model, the interaction energy between the mutable residues can be represented as an undirected graph, where vertices represent mutable residues, and edges represent pairwise interactions. An edge is present between two vertices when the pairwise energies between the two corresponding residues are included in the energy function. When the input energy function models all interactions between all mutable residues as pairwise energies, the *residue interaction graph* representing these interactions is the complete graph. We will refer to the complete residue interaction graph as the *full graph*. Every edge in the graph corresponds to a pairwise interaction between two mutable residues. By applying distance or energy cutoffs, the pairwise interactions between some mutable residues are omitted from the energy function. For every pair of residues (vertices) whose interactions are omitted, the corresponding edge between that pair is deleted from the residue interaction graph. This *sparse graph*, whose omitted edges correspond to the pairwise interactions omitted by the energy function, is called the *sparse residue interaction graph*. We will refer to the GMEC (which encodes both the conformation and sequence) of the full graph as the *full GMEC* and the GMEC of the sparse graph as the *sparse GMEC*. We will show that the full GMEC and sparse GMEC can be different, in both conformation and sequence. For any given conformation, we will refer to its computed energy with respect to the full graph as its *full energy*, and its energy with respect to the sparse graph as its *sparse energy*. For convenience, we will use $\delta$ and $\alpha$ to refer to distance and energy cutoffs, respectively:

- Distance cutoffs prune edges between two mutable residues whose minimum distance between any two atoms over all allowed rotamers is greater than a user-specified Euclidean distance $\delta$.
- Energy cutoffs prune edges between two mutable residues whose maximum absolute pairwise energy over all allowed rotamers is less than a user-specified energy $\alpha$.

Fig 1 shows the full and sparse residue interaction graphs for a protein design problem with 8 mutable residues. In this section, we have given high-level intuition for sparse residue interaction graphs, and their corresponding energy functions. For the proofs of Lemma 1 and Lemma 2 in S1 Text, however, precise definitions are useful to provide a mathematical basis for our claims. Hence, in S2 Text, we provide formal definition of a residue interaction graph, the sparse residue interaction graph, and the corresponding GMECs computed using such interaction graphs. We also provide a mathematical model for the cutoff criteria used to prune pairwise interactions from a residue interaction graph.

### Computational experiments

To study the sequence differences between the full and the sparse GMEC caused due to neglecting some pairwise energies, we need to compute the sparse GMEC first. We use the

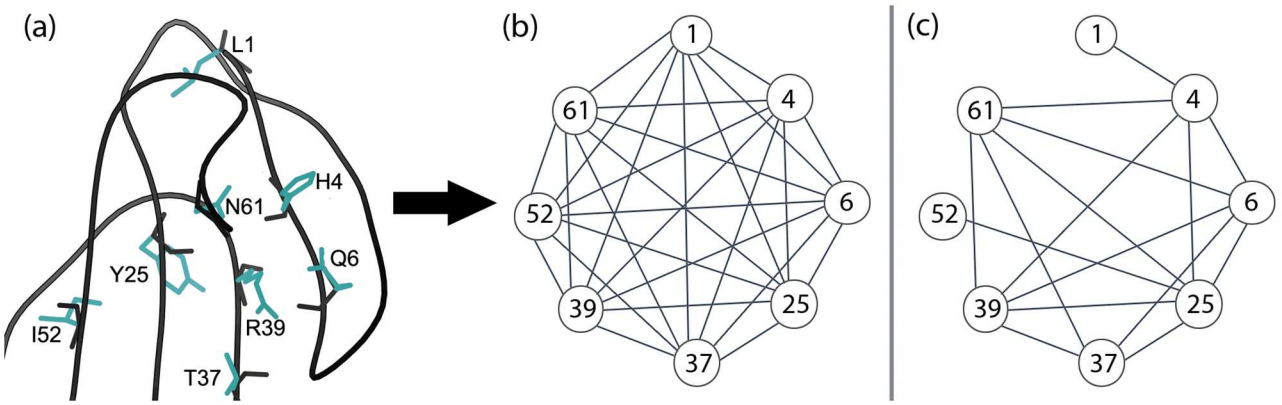


**Fig 1. Example of a sparse residue interaction graph.** (a) Cobrotoxin protein (PDB id: 1V6P) with the wild-type side chains of the 8 core mutable residues shown in cyan. (b) Design problem in (a) represented as a full residue interaction graph where all pairs of residues interact. (c) Design problem in (a) represented as a sparse residue interaction graph using a distance cutoff of $\delta = 8$ Å.

https://doi.org/10.1371/journal.pcbi.1005346.g001

protein redesign package developed by the Donald lab, OSPREY for this study [43]. This section describes the changes made to OSPREY to compute the sparse GMEC, and the computational experiments designed to study the differences between the full and the sparse GMEC.

OSPREY uses dead-end elimination (DEE) followed by the A* search algorithm [41, 42, 61, 62] to provably return the GMEC and enumerate conformations in order of increasing energy. DEE prunes a significant portion of the possible conformations, and following that, A* search explores the unpruned search space to ensure that the first conformation returned is the GMEC. After the GMEC is returned, A* continues to enumerate conformations in increasing order of energy, until either all conformations within an energy window $E_w$ of the GMEC are enumerated, or the number of conformations returned reaches a user defined number. A* guarantees that all conformations within $E_w$ of the GMEC are returned, and the list of conformations enumerated is gap-free.

To generate the sparse GMEC, we modified the A* search algorithm used by OSPREY to calculate the sparse energy of a conformation (Eq. 2 in S2 Text). We will refer to this variant of the A* search algorithm as *Sparse A**. Sparse A* evaluates conformations based on sparse energy (as opposed to full energy in traditional A* search), and can now be used for protein design with sparse residue interaction graphs. As Sparse A* retains all the guarantees provided by the A* search algorithm (because the algorithm is unmodified), the first conformation returned by Sparse A* is guaranteed to be the sparse GMEC, and it can also return a gap-free list of conformations within $E_w$ of the sparse GMEC in increasing order of sparse energy. This property of Sparse A* is used to prove a surprising result, namely, that Sparse A* can efficiently generate not only the sparse GMEC, but also the full GMEC. This will be discussed later in the Section entitled "Discussion".

Computational experiments were performed on the following design problems to generate the full and the sparse GMEC, and subsequently the energy, conformational, and sequence differences were analyzed:

- **Core designs**: 62 protein design problems, with the number of mutable residues ranging from 4-15 (each residue allowed to mutate to 5-10 amino acids) were taken from [32] and used with the same mutable residues and allowed amino acids.
- **Boundary designs**: PDB files for protein structures used in [32] were run through Naccess [63] to calculate the relative accessible surface area (RSA) for each residue. The residues with

RSA between 20-50% were classified as boundary residues. Terminal residues, residues forming disulfide bonds, and prolines were not designed. 46 design problems were chosen and at most 20 residues were designed in each case. The residues were allowed to mutate to their wild-type identities and all amino acids except proline.

- **Surface designs**: PDB files for protein structures used in [32] were run through Naccess [63] to calculate the relative accessible surface area (RSA) for each residue. The residues with RSA between 50-80% were classified as surface residues. Terminal residues, residues forming disulfide bonds, and prolines were not designed. 28 design problems were chosen and at most 20 residues were designed in each case. The residues were allowed to mutate to their wild-type identities and all polar and charged amino acids.

In all experiments, the DEE pruning stage was followed by either A* to get the full GMEC, or the following two steps to generate the sparse GMEC and gap-free list of conformations: 1) sparse residue interaction graph generation using a user-defined distance cutoff $\delta$ or energy cutoff $\alpha$, and 2) Sparse A* run to generate the sparse GMEC. For each design problem, Sparse A* was run four times using the following distance or energy cutoffs:

- $\delta$ = 8 Å;
- $\delta$ = 7 Å;
- $\alpha$ = 0.1 kcal/mol;
- $\alpha$ = 0.2 kcal/mol.

To further investigate how the differences in predicted sequence and conformation between the full and the sparse GMEC correlate with experimental measurements, we performed retrospective validation against 6 full-sequence designs from the literature, for which the designed mutants were experimentally determined to have improved thermal stability over the wild type [64–67]. Each example taken from the literature consisted of a protein redesign with an input structure together with experimentally measured melting point measurements showing a more thermostable designed mutant compared to the wild type sequence. We then performed computational redesign on the input structure. Consecutive residues of one or more adjacent secondary structures were allowed to either retain the wild-type identity or mutate to the amino acid identity of the designed, thermostabilized mutant. To compute the full GMEC, DEE pruning was followed by A* search. To compute the sparse GMEC, DEE pruning was followed by the generation of the sparse residue interaction graph with distance cutoff 7 Å, and then Sparse A* search. The number of mutable residues varied from 10-19 residues. The full and the sparse GMEC were then correlated against the measured melting point data.

The input model consists of a rigid backbone, rigid, discrete side-chain rotamers, and a pairwise energy function. All designs were done keeping the backbone fixed and modeling side-chain flexibility using the modal values of rotamers from the Penultimate rotamer library [22]. The energy function consisted of the AMBER van der Waals and electrostatic terms and the EEF1 pairwise implicit solvation model, as described in [11, 43]. Protein design formulations that consider additional side-chain flexibility [29, 32], backbone flexibility [28, 30, 31], free energy calculations [44, 61], or more accurate energy functions [68] have been developed. Nevertheless most of them call as a subroutine the simplified model discussed in this paper, which can be viewed as a core calculation common to most protein design software. Hence, the accuracy of this computation bounds the accuracy of the overall design. For more details on the input model, such as protein structures, mutable residues, and design protocol, please refer to S3 Text.

## Results

Out of the 62 core, 46 boundary, and 28 surface protein redesign problems, A* returned the full GMEC, and Sparse A* returned the sparse GMEC (with all four distance and energy cutoffs given in the Section entitled "Computational experiments") for all 62 core, 21 boundary, and 12 surface design problems, shown as green dots in Fig 2. These design problems were used to study the effects of different distance and energy cutoffs. Three kinds of differences

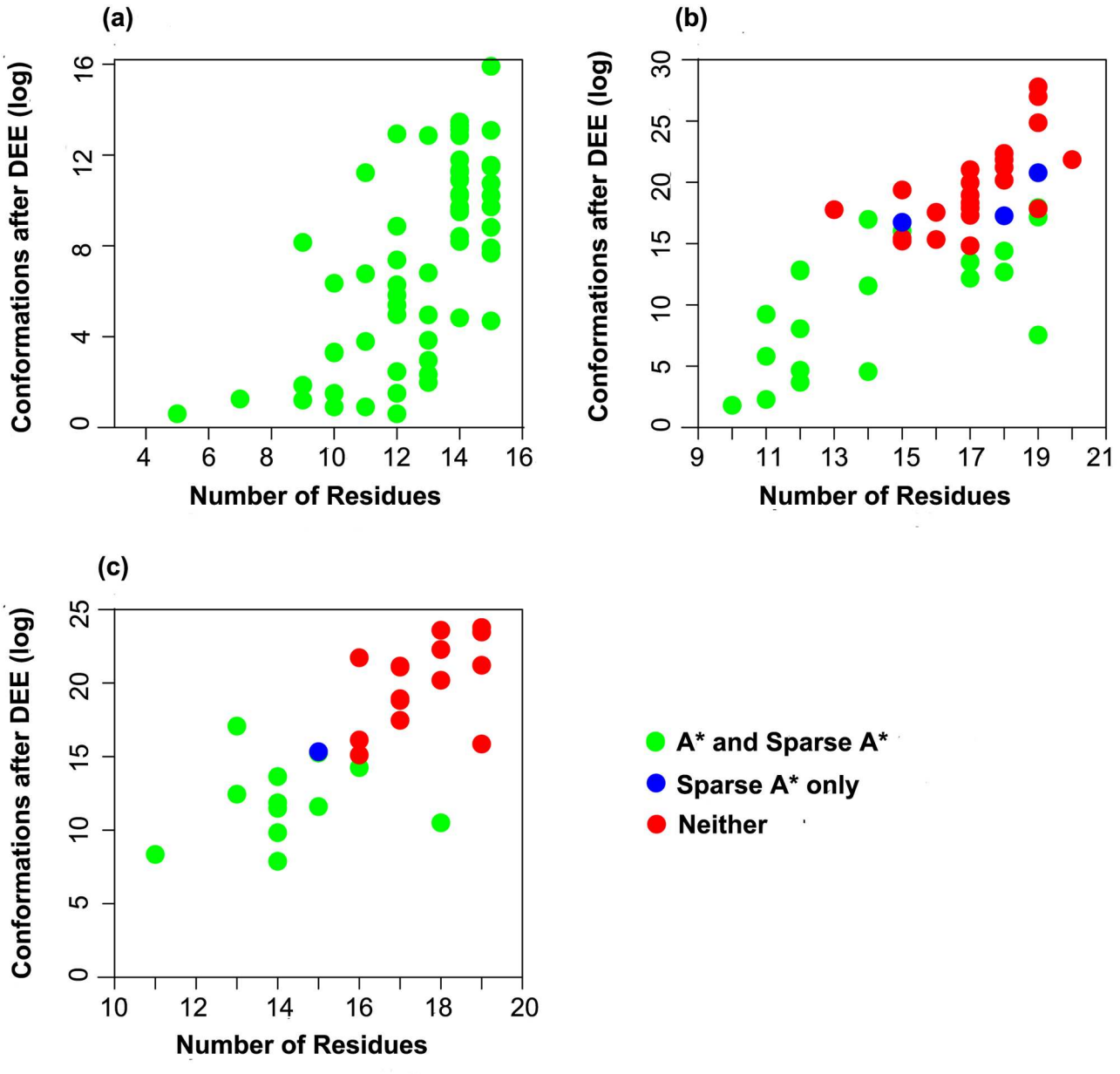


**Fig 2. Overview of the 136 protein design test problems on 62 proteins studied in this paper. Different problems required different amounts of resources**. (a) 62 core protein design problems, (b) 46 boundary design problems, and (c) 28 surface design problems. Design problems where A* returned the full GMEC and Sparse A* returned the sparse and the full GMEC are shown in green. Design problems where A* ran out of memory (30GB) before returning the full GMEC and Sparse A* returned the sparse GMEC are shown in blue. Design problems where both A* and Sparse A* ran out of memory (30GB) before returning any conformation are shown in red.

https://doi.org/10.1371/journal.pcbi.1005346.g002

were studied between the full and the sparse GMEC for each of the distance and energy cutoffs: rotamer, sequence, and energy. The following subsections (entitled "Core redesign", "Boundary redesign", and "Surface redesign") discuss the details of this analysis. The protein design problems for which only Sparse A* run finished (blue points) and for which both A* and Sparse A* ran out of memory (red points) are discussed in the Section entitled "Discussion".

## Core redesign

For 62 core designs, Sparse A* with both distance cutoffs returned the sparse GMEC identical to the full GMEC for all 62 design problems, and in 59 design problems with energy cutoff $\alpha$ = 0.1 kcal/mol, and in 58 design problems with energy cutoff $\alpha$ = 0.2 kcal/mol. For the four core problems where the full GMEC was different from the sparse GMEC, the full GMEC was the second conformation returned by Sparse A*, and the energy difference between the full and the sparse GMEC was less than the energy cutoff of 0.2 kcal/mol. Out of these four design problems, two had sequence differences between the full and the sparse GMEC: the human sulfite oxidase cytochrome b5 domain (PDB id: 1MJ4) and bacterial iron-sulfur protein (PDB id: 3A38) had single amino acid differences between the full and the sparse GMEC (residue 50 for 1MJ4 and residue 26 for 3A38). In both cases, serine in the full GMEC was replaced by alanine in the sparse GMEC. Except for these two cases, distance and energy cutoffs did not have sequence-changing effects on the GMEC returned for core designs. Interestingly, using an energy cutoff of 0.2 kcal/mol results in omitting a large fraction of residue pairs for most of the core design problems, between 45% to 80% (S1 Fig). Despite the tightly packed nature of the protein core, the energy interactions were less than 0.2 kcal/mol, which is less than the typical van der Waals interaction energy of 0.5-1 kcal/mol.

## Boundary redesign

Unlike core designs, the number of boundary design problems where the sparse GMEC was identical to the full GMEC was larger for energy cutoffs than distance cutoffs, as shown in Fig 3(a). The energy cutoff of $\alpha$ = 0.1 kcal/mol gave the best results, returning the sparse GMEC identical to the full GMEC in 19 out of the 21 boundary design problems. For problems where the full GMEC and the sparse GMEC were different, the full energy and sequence difference between the full and the sparse GMEC are larger for distance cutoffs than for energy cutoffs, (Fig 3(b) and 3(c)), with the distance cutoffs introducing sequence differences in a total of 24 residues as compared to 11 residues with energy cutoffs (S1 Table). For a single design problem, this number can be as high as 6 (C-terminal domain of the Rous Sarcoma Virus capsid protein, PDB id: 3G21), which is more than one-third of the 15 mutable residues for that design problem.

## Surface redesign

Similar to boundary design problems, energy cutoffs returned a sparse GMEC which was identical to the full GMEC in more cases than distance cutoffs did. The sparse GMEC is identical to the full GMEC in 10 out of the 12 surface design problems for energy cutoff $\alpha$ = 0.1 kcal/mol, and only for 1 out of 12 for distance cutoff $\delta$ = 7 Å (Fig 4(a)). Sequence differences between the full and the sparse GMEC occur even though the energy differences between these GMECs are small. Unlike boundary designs, where in a few cases the energy cutoffs introduced more sequence differences between the full and the sparse GMEC, the energy cutoff of $\alpha$ = 0.1 kcal/mol has a smaller or equal number of sequence differences than distance cutoffs in all 12 surface design problems, as shown in Fig 4(c). Overall, distance cutoffs

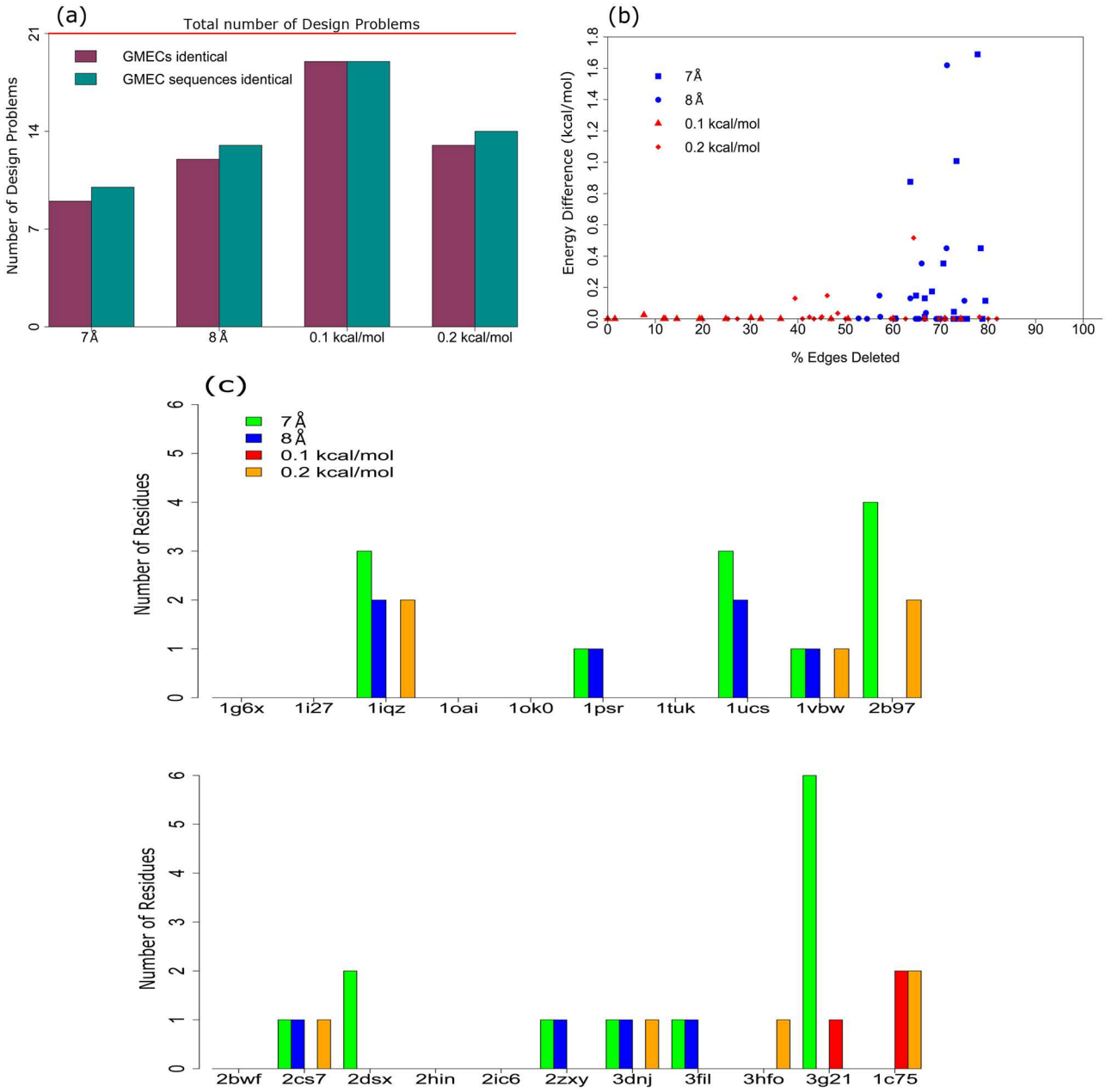


**Fig 3. Sparse residue interaction graphs introduce differences in energy, conformation, and sequence of the GMEC.** Data shown for 21 boundary design problems, for each of which Sparse A* was run with the following cutoffs: distance cutoff $\delta = 8$ Å, $\delta = 7$ Å, energy cutoff $\alpha = 0.1$ kcal/mol and $\alpha = 0.2$ kcal/mol. Number of mutable residues in each design problem ranged from 10-20. (a) Number of design problems where full GMEC and sparse GMEC are identical (purple), and where the sequences of the full GMEC and sparse GMEC are identical (cyan). The total number of boundary design problems (21) is indicated by the horizontal red line. (b) Percentage of edges deleted from the residue interaction graph vs. the full energy difference between full GMEC and sparse GMEC. (c) Number of residues with different amino acids between the full GMEC and the sparse GMEC. *y*-axis value of 0 indicates that the sequences of the full GMEC and the sparse GMEC are identical.

https://doi.org/10.1371/journal.pcbi.1005346.g003

introduced sequence differences in a total of 25 residues, as compared to 9 residues for energy cutoffs (S2 Table).

Since distances between residues on the surface of the protein are larger as compared to boundary or core regions, Sparse A* was run for the 12 surface designs problems again with a

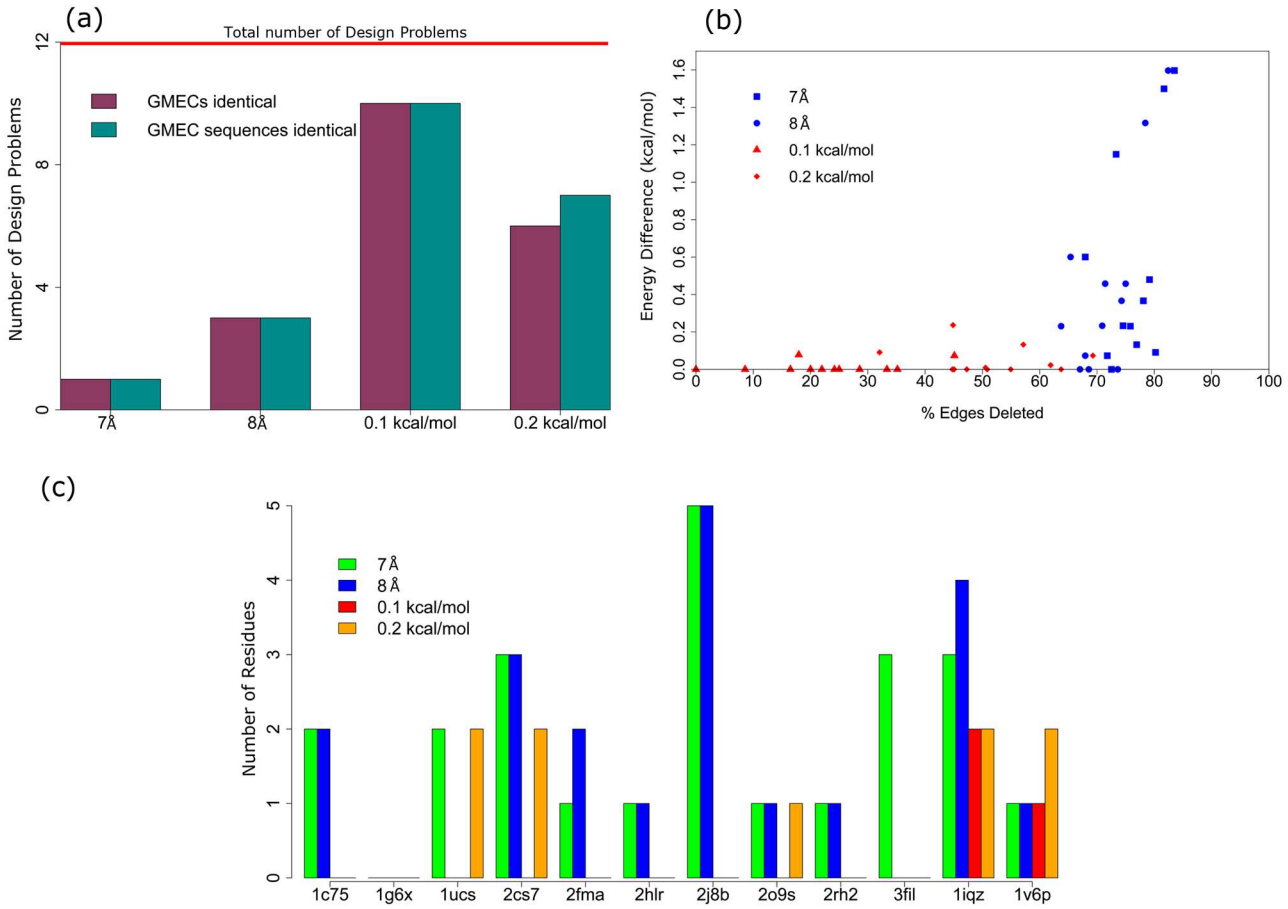


**Fig 4. Sparse residue interaction graphs introduce differences in energy, conformation, and sequence of the GMEC.** Data shown for 12 surface design problems, for each of which Sparse A* was run with the following cutoffs: distance cutoff $\delta = 8$ Å, $\delta = 7$ Å, energy cutoff $\alpha = 0.1$ kcal/mol and $\alpha = 0.2$ kcal/mol. Number of mutable residues in each design problem ranged from 11-19. (a) Number of design problems where full GMEC and sparse GMEC are identical (purple), and where the sequences of the full GMEC and sparse GMEC are identical (cyan). The total number of boundary design problems (12) is indicated by the horizontal red line. (b) Percentage of edges deleted from the residue interaction graph vs. the full energy difference between full GMEC and sparse GMEC. (c) Number of residues with different amino acids between the full GMEC and the sparse GMEC. *y*-axis value of 0 indicates that the sequences of the full GMEC and the sparse GMEC are identical.

https://doi.org/10.1371/journal.pcbi.1005346.g004

distance cutoff $\delta = 10$ Å. This resulted in the sparse GMEC being identical to the full GMEC for 5 out of 12 design problems, as compared to 3 out of the 12 design problems for $\delta = 8$ Å. For one additional case (bacterial oxidized ferredoxin protein, PDB id: 1IQZ), the number of amino acid sequence differences was reduced. However, this still only increases the number of design problems for which the full GMEC and the sparse GMEC are identical to half of the 10 when using the energy cutoff of $\alpha = 0.1$ kcal/mol. Overall, increasing the distance cutoff from 8 Å to 10 Å decreased amino acid differences in only 3 out of the 9 design problems. In human CD59 glycoprotein (PDB id: 2J8B), the number of amino acid sequence differences is 5 with distance cutoffs of 7 Å, 8 Å, and 10 Å, and increasing the distance cutoff led to no change in the sequence difference between the full and the sparse GMEC whatsoever.

## Large long-range interactions neglected by distance cutoffs

The above results suggest that using distance cutoffs can neglect long range interactions between residue pairs, and cause significant sequence differences between the GMECs

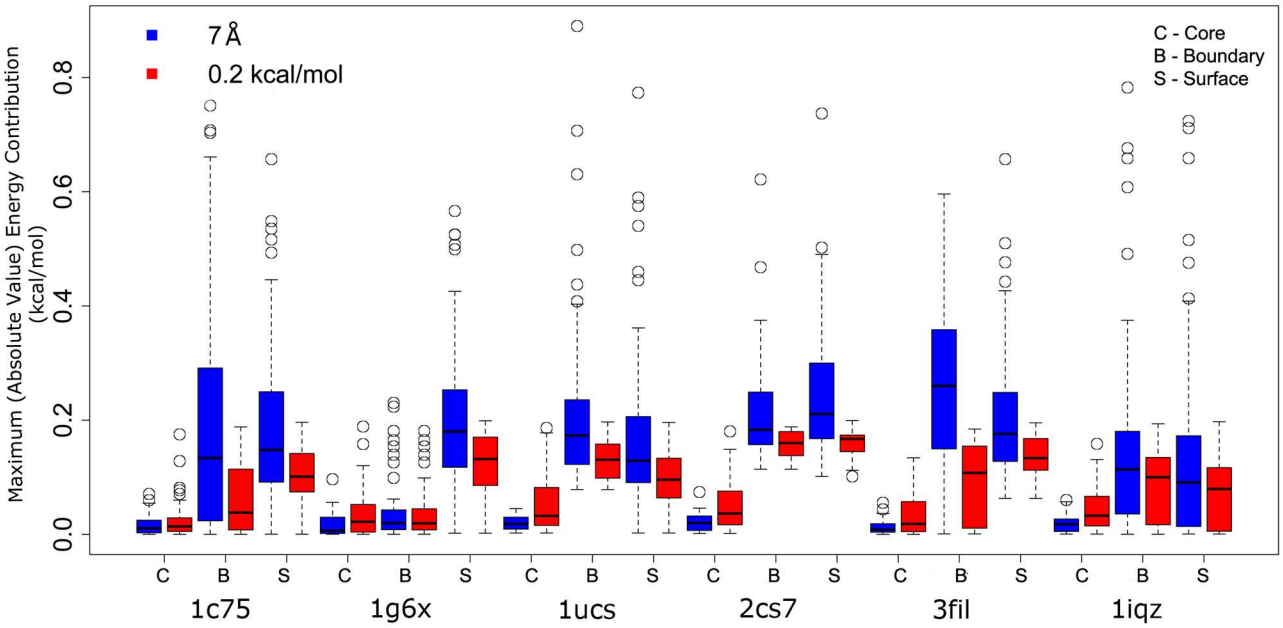


**Fig 5. Distance cutoffs can delete edges with large (almost 0.9 kcal/mol per edge) interaction energy.** The maximum (in absolute value) value of interaction energy for 18 test cases: each deleted edge with distance cutoff $\delta = 7$ Å (blue) and energy cutoff $\alpha = 0.2$ kcal/mol (red), for core (C), boundary (B), and surface (S) designs for 6 protein structures.

https://doi.org/10.1371/journal.pcbi.1005346.g005

returned (S2 Fig, S1 and S2 Tables). Using a larger distance cutoff (10 Å) did little to improve the results. Neglecting these long range interactions tends to have a larger effect on boundary and surface designs than on core designs. To investigate this further, the maximum energy (in absolute value) contributed over all rotamer pairs by each of the edges deleted using distance and energy cutoffs were analyzed. To eliminate any uncertainties, the results of core, boundary, and surface designs on the same protein structure were used for this analysis. Fig 5 shows the distribution of omitted pairwise energies of sparse graphs generated with either distance or energy cutoffs for 36 different protein design problems over 6 different structures. For the 6 protein structures shown in Fig 5, using distance cutoffs in boundary and surface designs can delete edges from the residue interaction graph with larger interaction energies, as compared to using energy cutoffs. The opposite occurs for core designs. This is consistent with the fact that both the distance cutoff $\delta = 7$ Å and energy cutoff $\alpha = 0.2$ kcal/mol had similar results for core designs, but for boundary and surface designs, the number of residues with different amino acids between the full and the sparse GMEC is larger for the distance cutoff than for the energy cutoff, except for bacterial cytochrome C-553 protein (PDB id: 1C75). By definition, energy cutoffs only delete an edge when its maximum absolute energy contribution is smaller than the specified limit. By contrast, distance cutoffs delete edges whose energy contributions can vary arbitrarily from being very small (0.05 kcal/mol) to being very large (almost 0.9 kcal/mol). As such, our results foreground a key difference between distance cutoffs and energy cutoffs: in terms of the energy contributed by each edge, precomputed energy cutoffs are more precise. While distance cutoffs omit any sufficiently distant pairwise interaction, providing limited control over the energy contributions of the omitted edges, energy cutoffs will never omit any pairwise interaction that can exceed the specified energy cutoff. Therefore energy cutoffs allow greater precision in selection of low-energy pairwise interactions.

One reason distance cutoffs are widely used is the assumption that sufficiently distant interactions do not affect local interactions. Our results indicate that this is not always true.

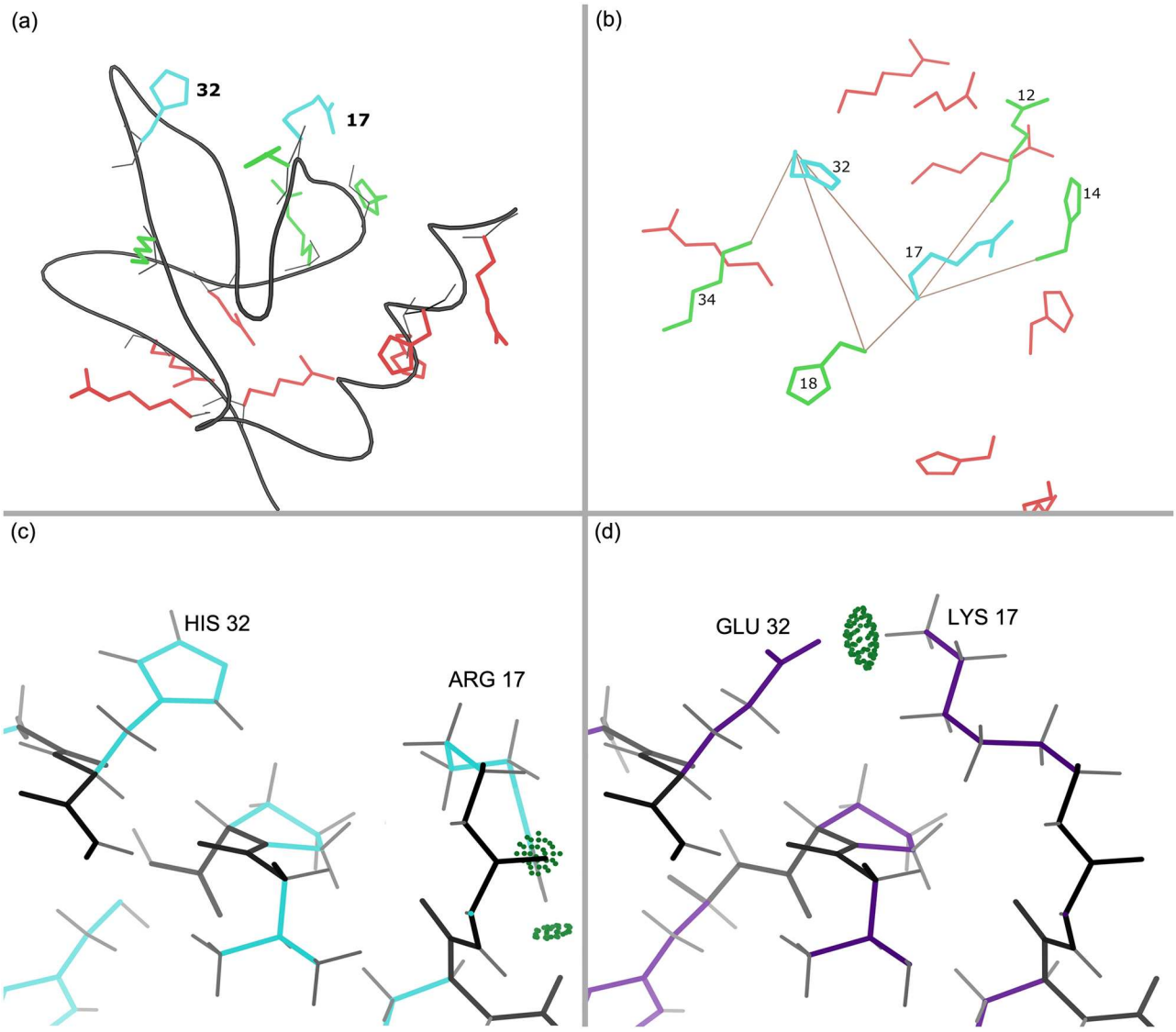


**Fig 6. Sequence differences with full vs. sparse residue interaction graphs: hydrogen bond is disrupted when long-range interactions are omitted.** Comparison between the sequences of the full and sparse GMEC for the surface design of domain of pneumococcal histidine triad A protein (PDB id: 2CS7) are shown. (a) Mutable residues of the sparse GMEC. Protein backbone is shown in black. Residues 17 and 32 are shown in cyan. With distance cutoff $\delta = 8$ Å, the interactions between red and cyan residues are eliminated in the sparse residue interaction graph. (b) Solid brown lines indicate residues interacting with the cyan residues in the sparse residue interaction graph. Amino acids at residues 17 and 32 from the sparse GMEC are shown in cyan. (c) Residues 17 and 32 of the sparse GMEC. (d) Residues 17 and 32 of the full GMEC. Note that the hydrogen bond between residues 17 and 32 (Lys:Glu) in the full GMEC (d) is lost in the sparse GMEC (c), where the side chain of residue 17 (Arg) forms hydrogen bonds with nearby backbone atoms. Hydrogen bonds are shown as green dotted-pillows that indicate the overlap between the vdW spheres of the hydrogen and the acceptor atom, generated using Probe [69].

https://doi.org/10.1371/journal.pcbi.1005346.g006

Figs 6 and 7 show the sequence differences between the full GMEC and the sparse GMEC with distance cutoff $\delta = 8$ Å in two such examples from bacterial cytochrome C-553 protein (PDB id: 1C75) and a domain of pneumococcal histidine triad A protein (PDB id: 2CS7). In both cases, neglecting the interaction between the distal residues (red) and the two proximal residues (cyan) leads to a missing hydrogen bond. In Fig 6, the amino acids of the full GMEC are replaced with entirely different amino acids. In the sparse GMEC, residue 17 is an arginine

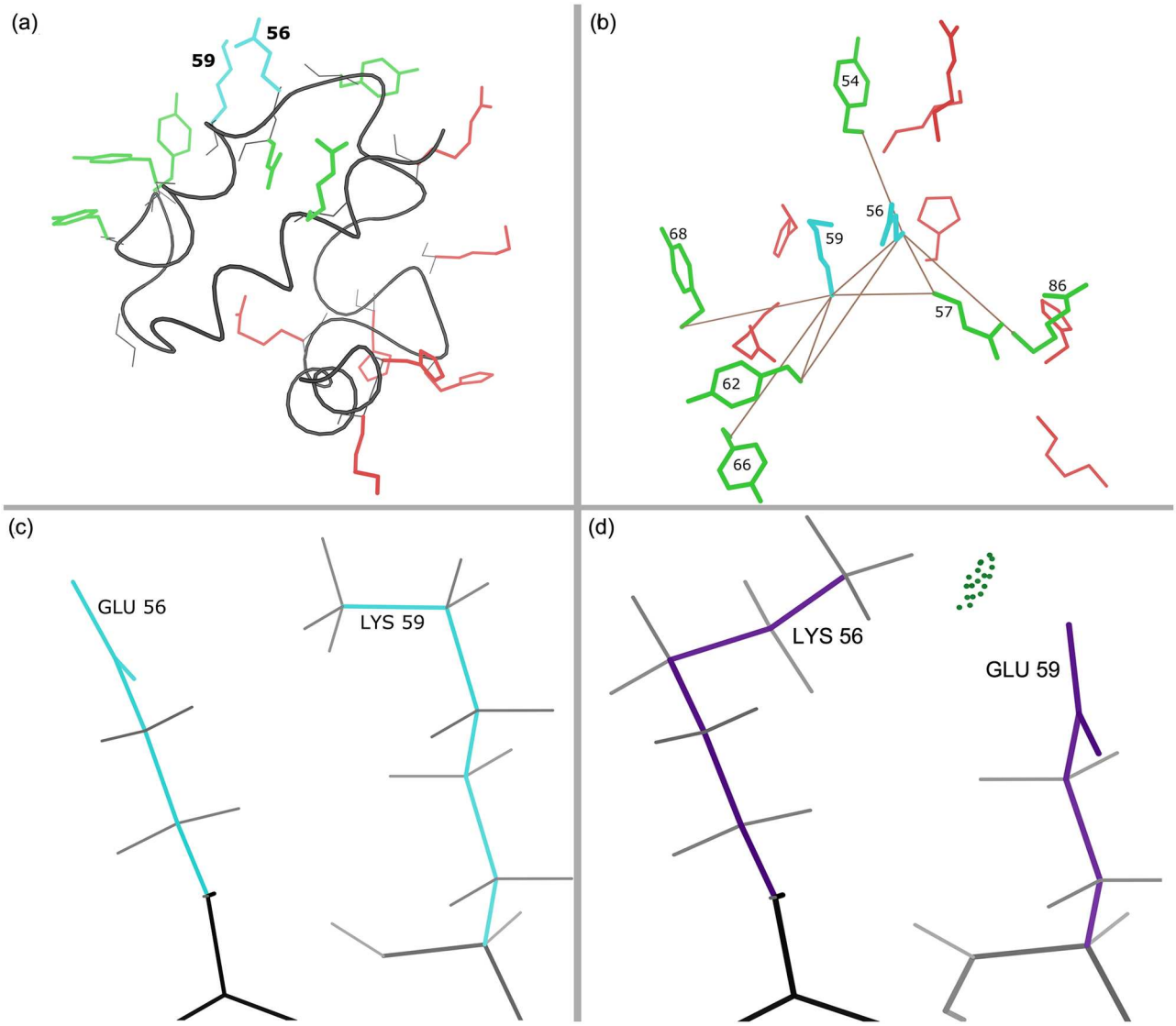


**Fig 7. Sequence differences with full vs. sparse residue interaction graphs: hydrogen bond is disrupted when long-range interactions are omitted.** Comparison between the sequences of the full and sparse GMEC for the surface design of bacterial cytochrome C-553 protein (PDB id: 1C75) are shown. (a) Mutable residues of the sparse GMEC. Protein backbone is shown in black. Residues 56 and 59 are shown in cyan. With distance cutoff $\delta = 8$ Å, the interactions between red and cyan residues are eliminated in the sparse residue interaction graph. (b) Solid brown lines indicate residues interacting with the cyan residues in the sparse residue interaction graph. Amino acids at residues 56 and 59 from the sparse GMEC are shown in cyan. (c) Residues 56 and 59 of the sparse GMEC, where the hydrogen bond is lost. (d) Residues 56 and 59 of the full GMEC, that form the hydrogen bond. Hydrogen bonds are shown as green dotted-pillows that indicate the overlap between the vdW spheres of the hydrogen and the acceptor atom, generated using Probe [69]. This hydrogen bond is lost in the sparse GMEC (c).

https://doi.org/10.1371/journal.pcbi.1005346.g007

instead of a lysine, and residue 32 is a histidine instead of a glutamic acid. While residue 17 and residue 32 of the full GMEC form a hydrogen bond, in the sparse GMEC the arginine at residue 17 forms hydrogen bonds with the backbone instead. In Fig 7, the amino acids at residues 56 and 59 are swapped between the full and the sparse GMEC. These examples illustrate two cases in which neglecting long-range interactions can disrupt favorable local interactions. In general, for the design problems analyzed in this paper, the disruption of local interactions is more common in surface designs. For 4 surface design problems, the interaction between

two mutable residues is different in the full and the sparse GMEC. In 3 of these cases a favorable local interaction is lost by using distance cutoff 7 Å.

## Retrospective validation against experimental data

To study how well the sequence differences between the full and the sparse GMEC correlate with experimental measurements, we conducted retrospective design experiments. As described in the methods section (entitled "Computational experiments"), we took examples of wild-type proteins from the literature which were computationally redesigned to improve thermostability (the designed protein had higher $T_m$ than the wild-type protein) [64–67], and redesigned a subset of each protein, allowing mutable residues to either retain their wild-type identity or mutate to the corresponding amino acid identity of the more stable designed mutant. We computed the full and the sparse GMEC for all 6 design problems and then compared the GMECs to identify differences in sequence, energy, and conformation. We correlated the difference in sequence with experimentally measured melting temperature data. Amino acid identities were restricted in the redesign procedure to ensure that any difference in sequence between the full and the sparse GMEC would correspond to either the less or the more stable protein sequence, and hence could be directly validated against the melting temperature data. The designed search space ranged from $6.65 \times 10^{15}$ to $6.13 \times 10^{25}$ conformations (see the section entitled "Rank of full GMEC in practice").

For 5 of the 6 protein design problems, amino acid differences between the full and sparse GMEC were found. In the sixth design problem, the sparse GMEC and full GMEC predicted the same sequence, but have different side-chain conformations. Table 1 lists the residue numbers for the 5 problems in which the full and sparse GMEC differed in sequence. For the two residues that have different amino-acid identity between the full and the sparse GMEC for the B1 domain of protein L (PDB id: 1HZ5), the full GMEC predicts the amino acid identities in the more stable designed protein for both Lys 4 and Glu 26. In contrast, for the U1 nuclear ribonucleoprotein A (PDB id: 1URN) the sparse GMEC predicts the amino acid identity in the more stable protein for both Gln 93 and Asp 97. For the designed engrailed homeodomain

**Table 1. Sequence correlation between designed mutant and wild type.** The table shows mutable residues at which the sparse and full GMEC predict different amino acid identities: one predicted the amino acid identity of the more stable designed mutant, and the other predicted the amino acid identity of the less stable wild type. The amino acid identity of the designed mutant is in bold. The wild-type amino acid identity is not in bold.

| Protein Structure | PDB id | Residue Number | Full GMEC[a] | Sparse GMEC[b] |
|---|---|---|---|---|
| Protein L [65] | 1HZ5 | 4 | **Lys** | Val |
| | | 26 | **Glu** | Phe |
| U1A [65] | 1URN | 93 | Ile | **Gln** |
| | | 97 | Met | **Asp** |
| Engrailed Homeodomain of *D. melanogaster* [64] | 1ENH | 47 | Ile | **Gln** |
| | | 55 | **Arg** | Lys |
| Symmetric protein homodimer [66] | 2MG4 | 52 | Asn | **Arg** |
| | | 54 | Arg | **Glu** |
| Acylphosphatase [65] | 2ACY | 8 | Gln | **Ser** |
| | | 10 | Lys | **Asp** |
| | | 14 | **Lys** | Phe |
| | | 16 | **Asp** | Lys |
| | | 76 | **Lys** | Asp |

[a]Full GMEC amino acid identity

[b]Sparse GMEC amino acid identity.

https://doi.org/10.1371/journal.pcbi.1005346.t001

dimer (PDB id: 2MG4), the sparse GMEC predicts the amino acid identity of the more stable protein for both Arg 52 and Glu 54. For the remaining two protein design problems, at some residues the full GMEC predicts the amino acid identity of the more stable designed protein, and for other residues the sparse GMEC predicts the amino acid at the more stable designed protein instead. Comparison of the complete sequences of the sparse and full GMEC can be found in the S3 Table. In summary, for the 5 design problems, there are 13 residues where the sparse and full GMEC predicted different amino acids. For these 13 residues, the amino acid identity of the more stable designed protein is predicted by the full GMEC for 6 residues, and by the sparse GMEC for the other 7 residues. These results suggest that when using a rigid backbone, rigid rotamer, GMEC-only input model and sparse or full pairwise energy function, it is unclear which of the two GMECs (sparse or full) will correspond to the desired protein function (in this case, improved thermostability).

We then analyzed the sparse residue interaction graph to determine the significance of the omitted edges. In particular, we identified pairwise interactions whose omission would change the sequence of the computed GMEC. Table 2 lists these omitted pairwise interactions, the minimum distance (closest Euclidean inter-residue distance between any two atoms when all rotamer combinations for the two residues are considered, see the Section entitled "Sparse residue interaction graphs and protein design") between the interacting residue pair, the total difference in energy between the full and sparse GMEC, and the difference in energy contributed by these omitted edges to the sparse and full GMEC. The omission of the pairwise interactions listed in Table 2 alone was large enough to change the sequence of the computed GMEC, and even lead to changes in experimental measurements. For example, in the case of acyl phosphatase (PDB id: 2ACY), Fig 8 shows the sequence differences and key high-energy long-range interactions omitted in the sparse residue interaction graph. In the full GMEC the minimum distance between residues Glu 12 and Lys 76 is 7.6 Å, and its high-energy long-range interaction of -0.34 kcal/mol is omitted in the sparse residue interaction graph. The minimum distance between residues Asp 10 and Arg 77 is 8.6 Å, and its high-energy long-range interaction of -0.26 kcal/mol is omitted in the sparse residue interaction graph. The corresponding pairwise energy between residues Glu 12 and Asp 76 in the sparse GMEC is 0.318 kcal/mol, and the corresponding pairwise energy between residues Glu 12 and Asp 76 in the sparse GMEC is 0.314 kcal/mol. This amounts to a total difference of 1.24 kcal/mol. The energies of the sparse and full GMEC differ by only 0.75 kcal/mol. As can be seen in panel (c) of Fig 8, the sparse GMEC neglects the energetically unfavorable interactions between both the negatively charged

**Table 2. Omitting key high-energy long-range interactions changes the sequence of the GMEC.** The table shows the one or two highest energy interactions omitted by a distance cutoff of 7 Å. The omission of these edges alone is sufficient to change the sequence of the GMEC computed using a sparse residue interaction graph.

| PDB id | High-energy Pairs[a] | Distance[b](Å) | Full Energy Difference[c] (kcal/mol) | Omitted Energy Difference[d] (kcal/mol) |
|---|---|---|---|---|
| 1HZ5 | (23, 26) | 7.1 | 0.16 | 0.393 |
| 1URN | (90, 97) | 7.4 | 0.175 | 0.462 |
| 1ENH | (43, 52), (44, 52) | 10.7, 7.86 | 0.181 | 0.231 |
| 2MG4 | (8, 54) | 14.3 | 0.201 | 0.356 |
| 2ACY | (10, 77), (12, 76) | 8.6, 7.6 | 0.75 | 1.24 |
| 1VJQ | (12, 22), (18, 24) | 9.92, 8.57 | 0.121 | 0.214 |

[a]Residue numbers of the two mutable residues in the omitted pairwise interaction

[b]Minimum distance $d_{min}(i, j)$ (see S2 Text) of the corresponding residue pair

[c]Energy difference between the sparse GMEC and full GMEC using the full energy function (Eq. 1 in S2 Text)

[d]Cumulative energy difference between the sparse and full GMEC for only the pairwise interactions *in column (a)*

https://doi.org/10.1371/journal.pcbi.1005346.t002

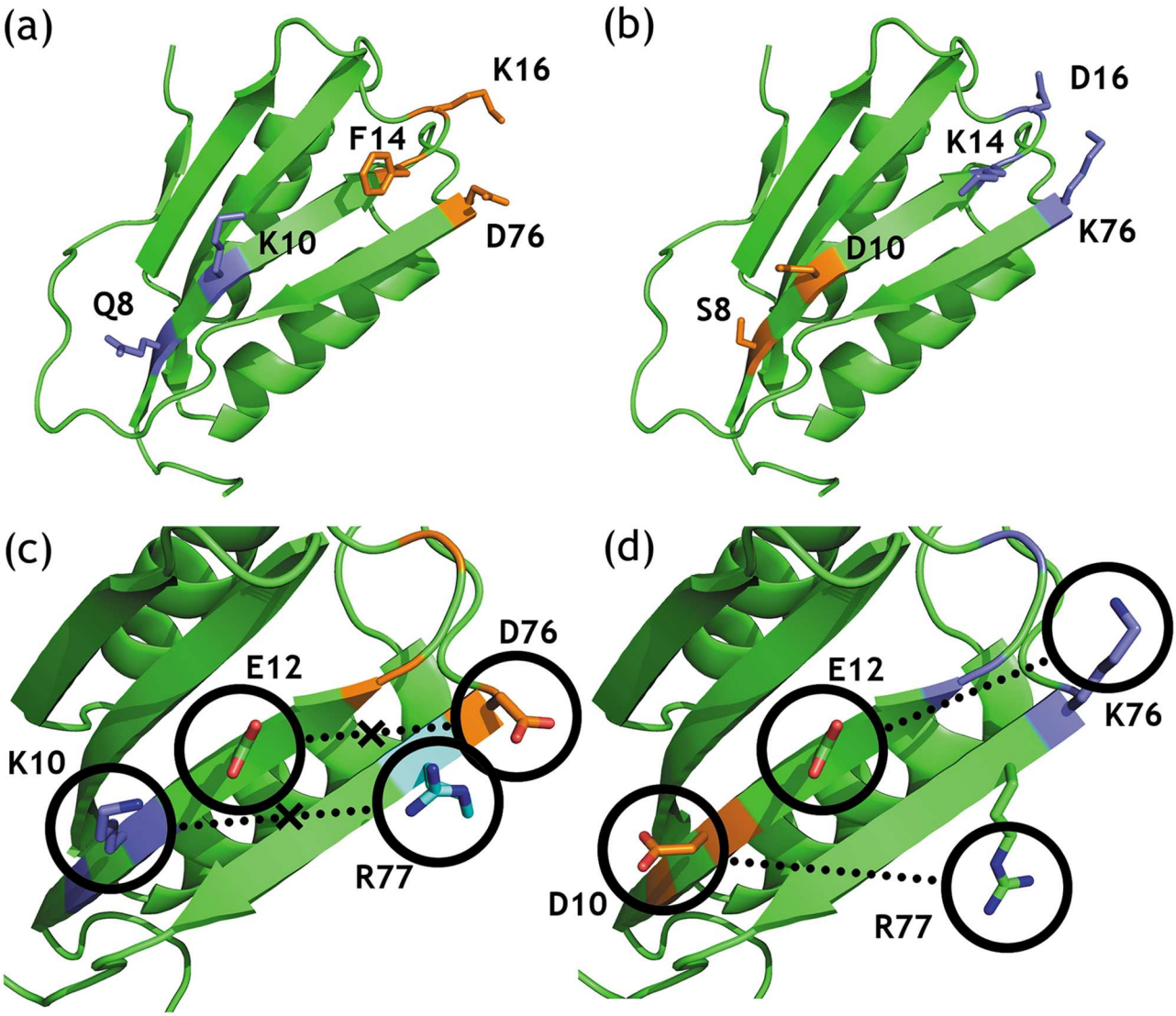


**Fig 8. Sequence differences between full vs. sparse residue interaction graphs: biophysically significant long-range interactions are omitted and change the sequence of the GMEC.** Comparison between the sequences of the full and sparse GMEC for the restrospective design of acyl phosphatase (PDB id: 2ACY) are shown. Protein backbone is shown as a green ribbon. (a) and (b) show the residues where the sparse and full GMEC have different amino acid identities: (a) shows the sparse GMEC, and (b) shows the full GMEC. Residues for which the amino acid identity is that of the thermostabilized mutant are shown in blue. Residues for which the amino acid identity is that of the wild-type are shown in orange. (c) and (d) show the omitted pairwise interactions in the sparse GMEC and their corresponding interactions in the full GMEC. Residues whose sequence and conformation are that of the full GMEC are shown in green. Arg 77 has a different side-chain conformation in the sparse GMEC, and is highlighted in cyan. Oxygen and nitrogen atoms are colored to show favorable and unfavorable interactions. (c) With distance cutoff $\delta = 7$ Å, the unfavorable interactions between Glu 12 and Asp 76 and Lys 10 and Arg 77 are omitted in the sparse residue interaction graph, and not considered when computing the sparse GMEC. (d) In the full GMEC these interactions are replaced with favorable interactions instead.

https://doi.org/10.1371/journal.pcbi.1005346.g008

glutamic acid at residue 12 and aspartic acid at residue 76, and the positively charged lysine at residue 10 and arginine at residue 77. These unfavorable long-range electrostatics are not found in the full GMEC, as seen in panel (d). Omitting these two pairwise interactions alone would change the sequence of the corresponding GMEC. Note that in this design problem, large, favorable electrostatic pairwise interactions in the full GMEC are replaced with large,

unfavorable electrostatic interactions in the sparse GMEC. The cumulative difference is greater than 1.2 kcal/mol, which is large enough to be biophysically relevant.

Even when the sparse energy function predicts amino acid identities of the more thermostable designed mutant, the difference in energy between the sparse GMEC and full GMEC can manifest as a difference in backbone coordinates and side-chain conformations. We analyzed an example of this, where the full GMEC predicts a rotamer that closely resembles the wild type while the sparse GMEC predicts a rotamer which has a $\chi_1$ angle difference of 90.6°. For human procarboxypeptidase A2 (PDB id: 1AYE), the sequences of the full and sparse GMECs were identical, but the residue conformations differed. To test if sparse residue interaction graphs could be a source of conformational difference between the predicted and experimentally observed residue conformations, we performed side-chain placement using OSPREY on the designed mutant of human procarboxypeptidase A2 (PDB id: 1VJQ). The structure of the designed mutant was used as input to rule out backbone changes as an additional source of error. We analyzed the conformations of the sparse GMEC, the full GMEC, and the crystal structure, and found the predicted conformations of sparse and full GMEC differed at two residues. At residue Glu 18, the rotamer of the full GMEC coincides closely with the crystal structure, whereas the rotamer of the sparse GMEC had a 90.6° difference in its $\chi_1$ angle, differing significantly from crystal structure. At residue Lys 22, both the sparse and the full GMEC preserve the overall direction of the charged amine group, but their $\chi$-angles differ from the crystal structure.

In the side-chain placement problem for the designed mutant of human procarboxypeptidase A2 (PDB id: 1VJQ), two pairwise interactions contributed the most to the energy difference between the sparse and full GMEC: the pairwise interactions between residues Glu 12 and Lys 22, and residues Glu 18 and Asp 24. The minimum distance between residues Glu 12 and Lys 22 is 9.9 Å, and in the full GMEC their long-range interaction of -0.42 kcal/mol is omitted in the sparse residue interaction graph. The corresponding pairwise energy between residues 12 and 22 in the sparse GMEC is -0.32 kcal/mol. The minimum distance between residues Glu 18 and Asp 24 is 8.6 Å, and in the full GMEC its long-range interaction of 0.21 kcal/mol is omitted in the sparse residue interaction graph. The corresponding pairwise energy between residues Glu 18 and Asp 24 in the sparse GMEC is 0.32 kcal/mol. The energy difference from these two edges account for a cumulative difference of 0.21 kcal/mol. The energies of the sparse and full GMEC differ by 0.16 kcal/mol. Omitting these two pairwise interactions alone would change the conformation of the computed GMEC.

## Discussion

Our results not only show how omitted pairwise interactions can change the computed sequence of the GMEC in computational protein design, but also that these differences in sequence can correlate with experimental measurements. In the examples where the full GMEC predicts mutations that correlate with the thermostable mutant but the sparse GMEC does not, key high-energy, long-range pairwise interactions are consistently omitted from the sparse residue interaction graph and this failure to account for long-range interactions changes the sequence of the sparse GMEC. Even in cases where the sparse GMEC better correlates with the designed mutant, the energy contribution of pairwise interactions that are omitted by distance cutoffs is significant.

In this study, we imposed many modeling assumptions: the backbone is rigid, side-chain flexibility is confined to rigid rotamers, and a single conformation, namely the GMEC, is assumed to be representative of the thermodynamic ensemble. Since the GMEC (sparse or full) is the provably optimal sequence and structure predicted by the input model, and since

this model is commonly used as a subroutine in many empirical designs [1, 24, 39, 45–60], our results represent a bound on how well any algorithm can do, using either the sparse or full input models. Our results show that commonly used distance cutoffs, especially when computed without bounding the difference in energy between the sparse and full GMEC, do in fact introduce error into any computational structure-based protein design protocol. Although this error does not always negatively impact downstream experiments, the computational expedience derived from applying distance cutoffs can come with a cost: measurable loss in accuracy. To overcome this cost, algorithms that merely compute a low-energy locally optimal solution or even the GMEC of the sparse residue interaction graph are inadequate. Therefore, it will be highly beneficial if we can compute both the full and the sparse GMEC efficiently, while still harnessing the computational advantages of the reduced search space provided by sparse residue interaction graphs. In this section we describe how to efficiently compute both the full and the sparse GMEC, by a novel approach that combines sparse residue interaction graphs with ensemble-based design algorithms. The new approach, called *Energy-bounding enumeration*, is comparable in speed to sparse GMEC search on the sparse residue interaction graph.

## Full GMEC with Sparse A*

As described in the Section entitled "Definitions related to sparse residue interaction graphs", sparse residue interaction graphs are generated by omitting distant or low-energy pairwise interactions from the energy function. These sparse graphs correspond to modified energy functions. Naturally, modifying the energy function can change the GMEC. The conformation and sequence of the GMEC of the full graph may be different from the GMEC of the sparse graph. However, for any pair of mutable residues, the maximum and minimum contributions of their pairwise interactions bound their total contribution to any conformation. Furthermore, their maximum and minimum contributions are efficient to compute. Thus, the contribution of any pairwise interaction can be efficiently bounded, and the cumulative maximal and minimal contribution of all omitted pairwise interactions can be efficiently bounded as well. These cumulative maxima and minima can be mathematically combined to bound the energy difference between the sparse and full GMEC (Lemma 1, S1 Text). Although the bounds given by Lemma 1 are often loose, it is important to know that the energy difference between the full and the sparse GMEC can always be bounded. This energy bound guarantees that a gap-free list (enumerated by a provable algorithm such as Sparse A*), that contains the sparse GMEC and all conformations within that energy bound of the sparse GMEC, will contain both the sparse and full GMEC. If we simply compute the full energies of all conformations in this list, then the conformation with the lowest full energy is guaranteed to be the full GMEC. Furthermore, computing the full energies for all conformations in the list is efficient (Lemma 2, S1 Text).

The fact that Sparse A* can enumerate both the sparse and the full GMEC means that we no longer have to worry about which of the two sequences (the full or the sparse GMEC) will predict the desired functional mutations. One can generate the sparse residue interaction graph, calculate the upper bound on the energy difference, and run Sparse A* to return a gap-free list of conformations that is guaranteed to return the full GMEC. The list of conformations returned by Sparse A* can then be re-ranked based on the full energy to get the full GMEC. Once both the full and the sparse GMEC are found, they can be evaluated based on more sophisticated methods for energy calculation [68, 70], or any other method that seems pertinent to the designer. Note that both the full and the sparse GMEC are guaranteed to be found only when using provable ensemble-based algorithms that are guaranteed not to miss any

conformation within the specified energy window. The re-ranking can be done relatively quickly (Lemma 2 in S1 Text), when the number of conformations that need to be generated by Sparse A* to find the full GMEC is not large. Both the full and the sparse GMEC were found for most of the design problems used in this study, and this is discussed in the Section entitled "Rank of full GMEC in practice".

In this section, we have provided high-level intuition showing how Sparse A* can be used to compute both the sparse and full GMEC. These statements are supported by mathematical guarantees, which show that the full GMEC is contained in the gap-free list enumerated by Sparse A*, and that it is efficient to compute the full GMEC from that list. In S1 Text we provide two Lemmas and their proofs. Lemma 1 proves an upper bound on the absolute difference in sparse energy between the sparse and full GMEC, and Lemma 2 gives the time complexity to compute the full GMEC from a gap-free list guaranteed to contain the full GMEC. We then describe how these two proofs are sufficient to compute both the sparse and full GMEC using Sparse A*. Finally, we briefly describe a recent provable algorithm [59], which uses concepts from dynamic programming to exploit the optimal substructure induced by sparse graphs, and achieve asymptotic time complexity significantly better than the worst-case time complexity of any algorithm using the full graph.

## Rank of full GMEC in practice

Fig 9 plots the calculated upper bounds vs. the actual full energy difference between the full and the sparse GMEC for the core, boundary, and surface designs. It is evident from the difference in the scale of the two axes that the actual energy difference between the full and the sparse GMEC (ranges from 0.05 kcal/mol to 1.6 kcal/mol) is an order of magnitude smaller than the computed upper bound, which can be as high as 40 kcal/mol. As a result, Sparse A* returned the full GMEC relatively early, well before all conformations within the energy bound (calculated using Lemma 1 in S1 Text) of the sparse GMEC were enumerated.

The full GMEC was found within the first 20 conformations of the sparse GMEC for all 21 boundary design problems with energy cutoffs, and for 19 design problems with distance cutoffs. For the two remaining problems, ClpS protease adaptor protein (PDB id: 3DNJ) and bacterial ferredoxin protein (PDB id: 1IQZ), the full GMEC was the $168^{th}$ and $5062^{nd}$ conformation returned by Sparse A* respectively with distance cutoff $\delta = 7$ Å. The rank of the full GMEC in the gap-free list of conformations enumerated by Sparse A* for all 21 boundary design problems are given in Table 3. For the 12 surface design problems, the full GMEC is within the first 30 conformations of the sparse GMEC for energy cutoffs, but for distance cutoffs, this number is on the order of a few hundred for some of the design problems. The rank of the full GMEC in the gap-free list of conformations enumerated by Sparse A* for all 12 surface design problems are given in Table 4.

Table 5 shows the rank of the full GMEC in the gap-free, in-order list enumerated after applying a distance cutoff $\delta = 7$ Å for the six retrospective design problems discussed in the Section entitled "Retrospective validation against experimental data" (Table 1 and Fig 8), along with the search space of each design problem. Note that even after constraining the mutable residues to only two allowed amino acids, the search space size for these designs can be large. The search space of the largest retrospective design problem, a 19-residue design of the engrailed homeodomain dimer (PDB id: 2mg4), was $6.13 \times 10^{25}$ conformations. Even in this case the rank of the full GMEC was merely 19 in the gap-free, in-order list, and our provable enumeration algorithm efficiently computes both. (For these experiments, computing 20 additional conformations after computing the sparse GMEC took less than 7.5 minutes.) The search space of the design problem that required the most enumeration, an 18-residue design

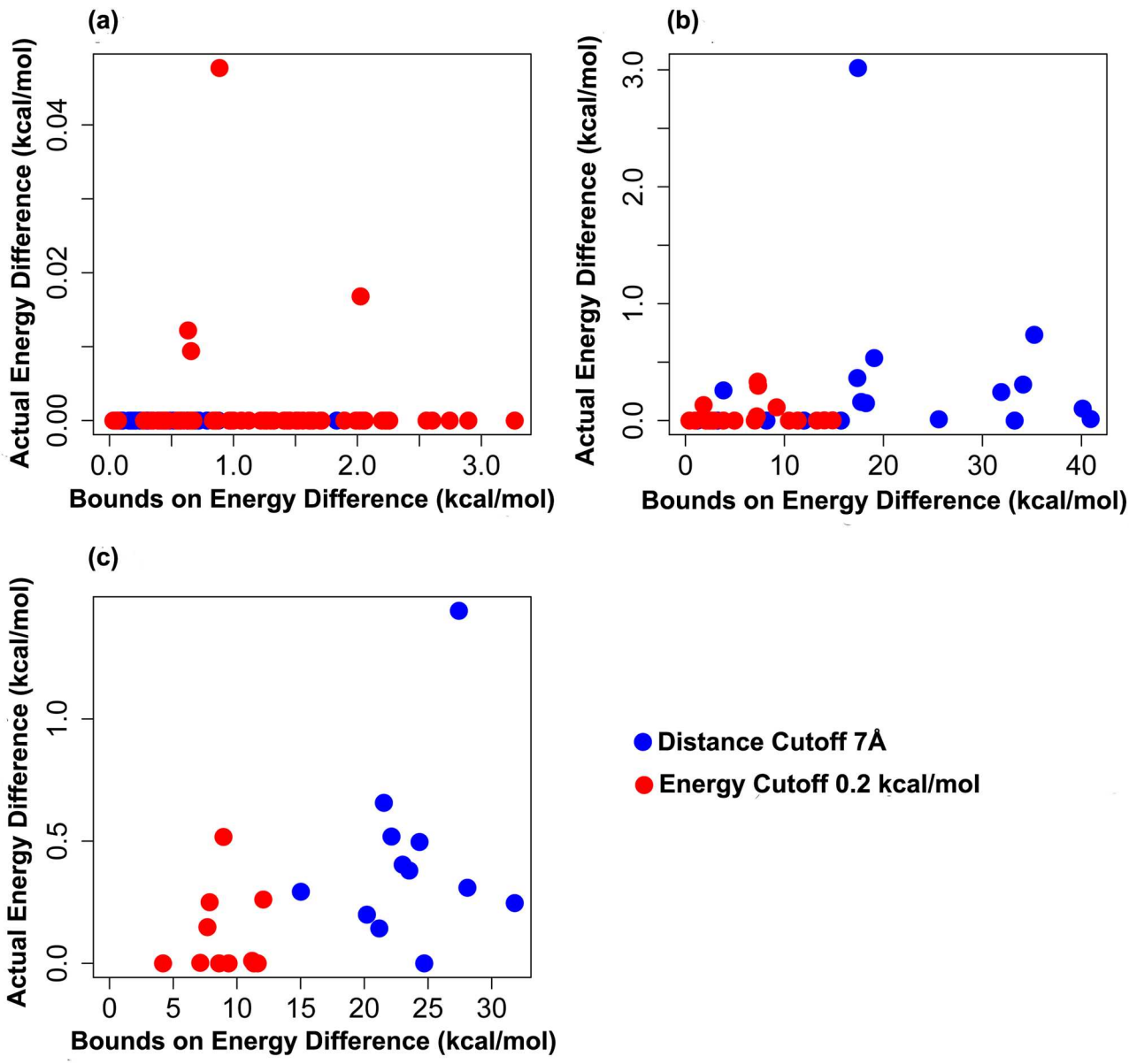


**Fig 9. Actual sparse energy difference between the full and sparse GMEC is much smaller than the theoretical energy bound.** Bounds on the sparse energy difference (as calculated by Lemma 1 in S1 Text) vs. the actual full energy difference between the full GMEC and sparse GMEC for distance cutoff $\delta = 7$ Å (blue) and energy cutoff $\alpha = 0.2$ kcal/mol (red). (a) 62 core protein design problems, (b) 21 boundary protein design problems, (c) 12 surface protein design problems.

https://doi.org/10.1371/journal.pcbi.1005346.g009

of acyl phosphatase, was $6.64 \times 10^{24}$, and enumerating 240 conformations to recover the full GMEC took less than 48 minutes. Across the six design problems, the median time to compute the sparse GMEC was 8.81 minutes, and the median time to compute the top 1000 conformations was 46.2 minutes. These times show how computing the top 1000 conformations is far less costly than computing 1000 GMECs. Therefore, using a provable algorithm to enumerate the 1000 lowest-energy conformations is in most cases very practical.

Overall, the energy difference between sparse and full GMEC was small, and therefore the rank of the full GMEC in the gap-free list enumerated by Sparse A* was low. For all but one design problem (including the core, boundary, surface, and retrospective designs), the full

**Table 3. The full GMEC is usually within 30 conformations of the sparse GMEC for boundary designs.** Rank of the full GMEC in the gap-free list of conformations generated by Sparse A* for 21 boundary protein design problems, with distance cutoffs $\delta = 7$ Å and $\delta = 8$ Å, and energy cutoffs $\alpha = 0.1$ kcal/mol and $\alpha = 0.2$ kcal/mol. Rank 1 indicates that the full and the sparse GMEC were identical.

| PDB id | $\delta = 7$ Å | $\delta = 8$ Å | $\alpha = 0.1$ kcal/mol | $\alpha = 0.2$ kcal/mol |
|---|---|---|---|---|
| 1G6X | 1 | 1 | 1 | 1 |
| 1I27 | 1 | 1 | 1 | 1 |
| 1IQZ | 5062 | 1448 | 1 | 17 |
| 1OAI | 1 | 1 | 1 | 1 |
| 1OK0 | 2 | 2 | 1 | 2 |
| 1PSR | 8 | 8 | 1 | 1 |
| 1TUK | 1 | 1 | 1 | 1 |
| 1UCS | 8 | 7 | 1 | 1 |
| 1VBW | 7 | 5 | 1 | 4 |
| 2B97 | 6 | 1 | 1 | 3 |
| 2BWF | 1 | 1 | 1 | 1 |
| 2CS7 | 11 | 11 | 1 | 8 |
| 2DSX | 4 | 1 | 1 | 1 |
| 2HIN | 1 | 1 | 1 | 1 |
| 2IC6 | 1 | 1 | 1 | 1 |
| 2ZXY | 3 | 3 | 1 | 1 |
| 3DNJ | 168 | 15 | 1 | 2 |
| 3FIL | 2 | 2 | 1 | 1 |
| 3HFO | 1 | 1 | 1 | 2 |
| 3G21 | 6 | 1 | 3 | 1 |
| 1C75 | 1 | 1 | 4 | 3 |

https://doi.org/10.1371/journal.pcbi.1005346.t003

GMEC was found by enumerating only the first 1000 conformations returned by Sparse A* (with both distance and energy cutoffs). This shows that even when limited time and memory prevent Sparse A* from provably enumerating the full GMEC (because of loose energy bounds), in practice the number of conformations that must be enumerated before Sparse A*

**Table 4. The full GMEC is usually within 1000 conformations of the sparse GMEC for surface designs.** Rank of the full GMEC in the gap-free list of conformations generated by Sparse A* for 12 surface protein design problems, with distance cutoffs $\delta = 7$ Å and $\delta = 8$ Å, and energy cutoffs $\alpha = 0.1$ kcal/mol and $\alpha = 0.2$ kcal/mol. Rank 1 indicates that the full and the sparse GMEC were identical.

| PDB id | $\delta = 7$ Å | $\delta = 8$ Å | $\alpha = 0.1$ kcal/mol | $\alpha = 0.2$ kcal/mol |
|---|---|---|---|---|
| 1C75 | 50 | 97 | 1 | 1 |
| 1G6X | 1 | 1 | 1 | 1 |
| 1UCS | 3 | 1 | 1 | 16 |
| 2CS7 | 37 | 64 | 1 | 4 |
| 2FMA | 18 | 2 | 1 | 1 |
| 2HLR | 6 | 6 | 1 | 1 |
| 2J8B | 286 | 286 | 1 | 1 |
| 2O9S | 840 | 943 | 1 | 2 |
| 2RH2 | 46 | 6 | 1 | 2 |
| 3FIL | 57 | 1 | 1 | 1 |
| 1IQZ | 2231 | 213 | 2 | 26 |
| 1V6P | 14 | 12 | 5 | 22 |

https://doi.org/10.1371/journal.pcbi.1005346.t004

**Table 5. The rank of the full GMEC is small for retrospective design problems, and the 1000 lowest-energy conformations can be enumerated quickly.**

| PDB id | Number of Residues | Number of Mutable Residues | Full GMEC Rank[a] | Search Space Size[b] (conformations) | Sparse GMEC Time[c] (minutes) | Full GMEC Time[d] (minutes) | Time to 1000 Conformations[e] (minutes) |
|---|---|---|---|---|---|---|---|
| 1hz5 | 61 | 15 | 10 | $2.07 \times 10^{19}$ | 21.0 | 25.1 | 60.7 |
| 1urn | 64 | 16 | 2 | $1.87 \times 10^{20}$ | 50.6 | 52.4 | 191 |
| 1enh | 66 | 12 | 2 | $6.65 \times 10^{15}$ | 0.230 | 0.254 | 20.3 |
| 2mg4 | 66 | 19 | 19 | $6.13 \times 10^{25}$ | 1.01 | 1.52 | 19.2 |
| 2acy | 98 | 18 | 240 | $6.64 \times 10^{24}$ | 16.6 | 48.0 | 119 |
| 1vjq | 73 | 26 | 3 | $4.10 \times 10^{18}$ | 0.251 | 0.346 | 31.8 |

[a]Rank of the full GMEC in the gap-free list enumerated by A* when applying the distance cutoff $\delta = 7$ Å

[b]Total number of possible conformations

[c]Time to compute the sparse GMEC

[d]Time to compute a gap-free list containing the full GMEC

[e]Time to compute a gap-free list of the 1000 lowest-energy conformations

https://doi.org/10.1371/journal.pcbi.1005346.t005

returns the full GMEC can be small. This provides useful information that can be used to compute the full GMEC using Sparse A* for design problems where A* fails. This is highlighted by the three boundary and one surface design problem for which Sparse A* with distance cutoff $\delta = 7$ Å (the cutoff which deleted the most of edges) returned the sparse GMEC, whereas A* ran out of 30 GB of memory before returning the full GMEC (orange points in Fig 2). Sparse A* returned the sparse GMEC along with a gap-free list of conformations for three boundary designs (heterogeneous nuclear ribonucleoprotein K (PDB id: 1ZZK), Beta-elicitin cinnamomin (PDB id: 2AIB), Dihydrofolate reductase type 2 (PDB id: 2RH2), and for one surface design of scorpion toxin protein (PDB id: 1AHO)). The number of conformations enumerated by Sparse A* was 47 for 1ZZK, 3029 for 2AIB, 46 for 2RH2, and 10,000 for 1AHO. Given the results that with distance cutoff $\delta = 7$ Å the full GMEC can be found almost always within the first 30 conformations for boundary designs, and within 1000 conformations for surface designs, the gap-free list computed by Sparse A* for the above four protein design problems almost certainly contains the full GMEC. Because the number of conformations that must be enumerated by Sparse A* to find the full GMEC is usually small, the GMECs of both the full and the sparse residue interaction graphs can be computed by enumerating a gap-free, in-order list of conformations.

Note that this study relies critically on provable algorithms that are guaranteed to enumerate the GMEC followed by a gap-free list of conformations in order of increasing energy. Without these algorithms it would be difficult and perhaps even unsound to compare the results of computational protein design with and without sparse residue interaction graphs, since differences induced by the sparse model can not be deconvolved from differences stemming from undersampling or inadequate stochastic optimization. Moreover, the provable guarantees of Lemma 2 (S1 Text) would not be possible if the enumeration algorithm missed any low-energy conformations within the calculated energy window of the sparse GMEC.

It has been previously argued that crucial improvements to the energy function and input model (e.g. side-chain flexibility, backbone flexibility, and entropy) should, for reasons of computational complexity, be accompanied by novel algorithmic enhancements [39]. Hence, it is also important to distinguish design algorithms that only apply distance cutoffs to the energy function [24, 45–52] vs. algorithms that exploit the optimal substructure induced by sparse residue interaction graphs (via techniques such as dynamic programming) [53–60].

While algorithms that only modify the energy function and algorithms that effectively exploit the optimal substructure both benefit from the reduced effective search space of sparse residue interaction graphs, significant large-scale gains in computational efficiency (including reduced asymptotic time complexity) are not achieved by the former, whereas they are *guaranteed* by the latter. By developing new, efficient methodologies and algorithms (such as Energy-bounding enumeration, used in the Section entitled "Discussion"), larger, harder problems built on more sophisticated biophysical input models become tractable without any increase in hardware capability. So even though physical hardware power may not improve quickly enough to relieve the computational costs of protein design, algorithmic improvements can reduce previously difficult or even intractable tasks in protein design to well-understood problems for which a wealth of efficient algorithms already exist. One example is a recent paper [70], which reduces the problem of protein design with continuous side-chain flexibility to the well-understood problem of protein design with discrete rotamers, enabling designers to perform designs with continuous side-chain flexibility as efficiently as they could previously perform discrete rigid rotamer designs.

## Conclusion

In this paper, we implemented a variant of the A* search algorithm in our lab's open source protein design package OSPREY, for protein design with sparse residue interaction graphs. We ran A* and Sparse A* on 136 different protein design problems involving core, boundary, and surface residues and analyzed the effects of using various distance and energy cutoffs. We compared the energies and sequences of the full GMEC returned by A* vs. the sparse GMEC returned by Sparse A*, and found that distance cutoffs, especially in surface and boundary designs, can lead to significant sequence differences between the full and the sparse GMEC. Our analysis indicates that the effects of distance cutoffs range from introducing no sequence differences in core designs, to sequence differences in almost all surface designs. By comparison, the effects of energy cutoffs are similar across core, boundary, and surface designs. In addition, we show examples of protein design problems in which neglecting long-range interactions alters local interactions. Furthermore, we performed retrospective designs for proteins with experimentally measured data, and our analysis of sequence differences between the full and the sparse GMEC indicates that it is not readily apparent if the sparse or full GMEC predicts mutations that perform better *in vitro*.

The sequence differences between the full and the sparse GMEC occur even though the energy differences between these GMECs are small. While these errors are severe, we provided a way to overcome these discrepancies. We used a provable, ensemble-based algorithm and showed that the full GMEC was found within the first 1000 conformations returned for all but one of our design problems. Because the number of conformations that must be enumerated to find the full GMEC is usually small, we can take advantage of the reduced search space provided by sparse residue interaction graphs and still efficiently compute both the full and the sparse GMEC. To do this, we compute a gap-free, in-order list of conformations. The gap-free list of low-energy conformations returned by Sparse A* is guaranteed to contain the full GMEC, and we show it takes only polynomial additional time to find the full GMEC (Lemma 2 in S1 Text). For 3 boundary and 1 surface design problem where A* failed to return even a single conformation, Sparse A* not only computed the sparse GMEC, but also enumerated a gap-free list of conformations that almost certainly contains the full GMEC. This provides a novel way for provable, ensemble-based algorithms and sparse residue interaction graphs to compute not only the sparse GMEC, but also the full GMEC for previously intractable design problems.

Previous studies have found that computational structure-based protein design protocols are often susceptible to forcefield inaccuracies (particularly hydrogen bonding and electrostatics) [71]. Distance cutoffs are widely used because interaction energy decreases with distance, which is relatively inexpensive (compared with energy cutoffs) to calculate. The underlying assumption is that beyond a certain distance, the interaction energy is negligible. Our results show that this is, in fact, not always true: these seemingly negligible interaction energies can add up, leading to significant sequence differences and changes in local residue interactions that can alter not only the structure, but also the function of the predicted protein. Therefore by using distance cutoffs, protein designers have been neglecting significant pairwise interactions, which may compromise the accuracy of their predictions. Our paper is the first large scale study of the magnitude and consequences of distance cutoffs and their effects. On the positive side, we showed that by combining sparse residue interaction graphs with provable, ensemble-based algorithms, we provide a way to overcome this inaccuracy. Therefore, by using provable algorithms in the manner we described, protein designers can continue to reap the benefits of distance cutoffs without worrying about loss in accuracy. The gap-free list of conformations generated will include the sequence of both the sparse and full GMEC, allowing both sequences to be inspected and tested. We believe this is a notable improvement over traditional protocols, which require designers to commit beforehand to either the sparse or full interaction model. Our work simultaneously exposes potentially significant experimental inaccuracies in the input model and provides novel methodology to address these inaccuracies directly.

## Supporting information

**S1 Text. Supplementary theory: Proofs for Lemma 1 and Lemma 2.**
(PDF)

**S2 Text. Supplementary theory and methods: Sparse residue interaction graphs.**
(PDF)

**S3 Text. Full details of computational experiments performed.**
(PDF)

**S1 Fig. Distance cutoffs prune a larger percentage of edges than energy cutoffs in boundary and surface design problems.** Number of unpruned conformations left after DEE vs. the percentage of edges deleted from the residue interaction graph. Two data points are plotted for each design problem: with distance cutoff $\delta = 7$ Å (blue), and energy cutoff $\alpha = 0.2$ kcal/mol (red). (a) 62 core design problems, (b) 46 boundary design problems, and (c) 28 surface design problems.
(TIF)

**S2 Fig. Distance cutoffs introduce amino acid changes in more residues than energy cutoffs.** Amino acid identities of residues in the full GMEC which were mutated in the sparse GMEC, for boundary and surface protein design problems with distance cutoff $\delta = 7$ Å and energy cutoff $\alpha = 0.2$ kcal/mol. The number in parenthesis indicates the cumulative number of residues across all design problems for which that amino acid was different in the sparse GMEC.
(TIF)

**S1 Table. Sequence differences between GMECs for boundary designs.** Sequence differences between the full and the sparse GMEC for boundary design problems, for distance cutoff $\delta = 7$ Å and energy cutoff $\alpha = 0.2$ kcal/mol. A dash ("–") indicates that the amino acid identity was the same in the full and the sparse GMEC.
(PDF)

**S2 Table. Sequence differences between GMECs for surface designs.** Sequence differences between the full and the sparse GMEC for surface design problems, for distance cutoff $\delta = 7$ Å and energy cutoff $\alpha = 0.2$ kcal/mol. A dash ("–") indicates that the amino acid identity was the same in the full and the sparse GMEC.
(PDF)

**S3 Table. Sequence correlation between designed mutant and wild type.** The table shows the sequences of the sparse and full GMEC. Residues at which the sparse or full GMEC has the same amino acid identity as the thermostabilized mutant are in bold. Residues at which the sparse or full GMEC has the same amino acid identity as the less stable wild-type are not in bold.
(PDF)

## Acknowledgments

The authors thank all members of the Donald lab, and Prof. Jane S. Richardson and Prof. David. C. Richardson for helpful discussion and comments.

## Author Contributions

**Conceptualization:** SJ JDJ ISG BRD.

**Data curation:** SJ JDJ ISG BRD.

**Formal analysis:** SJ JDJ ISG BRD.

**Funding acquisition:** BRD.

**Investigation:** SJ JDJ ISG BRD.

**Methodology:** SJ JDJ ISG BRD.

**Project administration:** BRD.

**Resources:** SJ JDJ ISG BRD.

**Software:** SJ JDJ ISG BRD.

**Supervision:** BRD.

**Validation:** SJ JDJ ISG BRD.

**Visualization:** SJ JDJ ISG BRD.

**Writing – original draft:** SJ JDJ ISG BRD.

**Writing – review & editing:** SJ JDJ ISG BRD.

## References

1. Donald BR. Algorithms in Structural Molecular Biology. The MIT Press (Cambridge); 2011.
2. Offredi F, Dubail F, Kischel P, Sarinski K, Stern AS, Van de Weerdt C, et al. De novo backbone and sequence design of an idealized alpha/beta-barrel protein: evidence of stable tertiary structure. Journal of molecular biology. 2003 Jan; 325(1):163–174. https://doi.org/10.1016/S0022-2836(02)01206-8 PMID: 12473459
3. Keating AE, Malashkevich VN, Tidor B, Kim PS. Side-chain repacking calculations for predicting structures and stabilities of heterodimeric coiled coils. Proceedings of the National Academy of Sciences of the United States of America. 2001 Dec; 98(26):14825–14830. https://doi.org/10.1073/pnas.261563398 PMID: 11752430

4. Dahiyat BI, Sarisky CA, Mayo SL. De Novo protein design: towards fully automated sequence selection. Journal of molecular biology. 1997 Nov; 273(4):789–796. https://doi.org/10.1006/jmbi.1997.1341 PMID: 9367772
5. Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, Baker D. Design of a novel globular protein fold with atomic-level accuracy. Science. 2003 Nov; 302(5649):1364–1368. https://doi.org/10.1126/science.1089427 PMID: 14631033
6. Hellinga HW, Richards FM. Construction of new ligand binding sites in proteins of known structure. I. Computer-aided modeling of sites with pre-defined geometry. Journal of molecular biology. 1991 Dec; 222(3):763–785. https://doi.org/10.1016/0022-2836(91)90510-D PMID: 1749000
7. Marvin JS, Hellinga HW. Conversion of a maltose receptor into a zinc biosensor by computational design. Proceedings of the National Academy of Sciences of the United States of America. 2001 Apr; 98(9):4955–4960. https://doi.org/10.1073/pnas.091083898 PMID: 11320244
8. Lippow SM, Tidor B. Progress in computational protein design. Current opinion in biotechnology. 2007 Aug; 18(4):305–311. https://doi.org/10.1016/j.copbio.2007.04.009 PMID: 17644370
9. Shifman JM, Mayo SL. Modulating calmodulin binding specificity through computational protein design. Journal of molecular biology. 2002 Oct; 323(3):417–423. https://doi.org/10.1016/S0022-2836(02)00881-1 PMID: 12381298
10. Looger LL, Dwyer MA, Smith JJ, Hellinga HW. Computational design of receptor and sensor proteins with novel functions. Nature. 2003 May; 423(6936):185–190. https://doi.org/10.1038/nature01556 PMID: 12736688
11. Chen CY, Georgiev I, Anderson AC, Donald BR. Computational structure-based redesign of enzyme activity. Proceedings of the National Academy of Sciences. 2009 Mar; 106(10):3764–3769. https://doi.org/10.1073/pnas.0900266106
12. Bolon DN, Mayo SL. Enzyme-like proteins by computational design. Proceedings of the National Academy of Sciences of the United States of America. 2001 Dec; 98(25):14274–14279. https://doi.org/10.1073/pnas.251555398 PMID: 11724958
13. Stevens BW, Lilien RH, Georgiev I, Donald BR, Anderson AC. Redesigning the PheA domain of gramicidin synthetase leads to a new understanding of the enzyme's mechanism and selectivity. Biochemistry. 2006 Dec; 45(51):15495–15504. https://doi.org/10.1021/bi061788m PMID: 17176071
14. Gorczynski MJ, Grembecka J, Zhou Y, Kong Y, Roudaia L, Douvas MG, et al. Allosteric inhibition of the protein-protein interaction between the leukemia-associated proteins Runx1 and CBFbeta. Chemistry & Biology. 2007 Oct; 14(10):1186–1197. https://doi.org/10.1016/j.chembiol.2007.09.006
15. Rudicell RS, Kwon YD, Ko SY, Pegu A, Louder MK, Georgiev IS, et al. Enhanced Potency of a Broadly Neutralizing HIV-1 Antibody In Vitro Improves Protection against Lentiviral Infection In Vivo. Journal of Virology. 2014 Nov; 88(21):12669–12682. https://doi.org/10.1128/JVI.02213-14 PMID: 25142607
16. Georgiev IS, Rudicell RS, Saunders KO, Shi W, Kirys T, McKee K, et al. Antibodies VRC01 and 10E8 neutralize HIV-1 with high breadth and potency even with Ig-framework regions substantially reverted to germline. Journal of Immunology. 2014 Feb; 192(3):1100–1106. https://doi.org/10.4049/jimmunol.1302515
17. Georgiev I, Acharya P, Schmidt SD, Li Y, Wycuff D, Ofek G, et al. Design of epitope-specific probes for sera analysis and antibody isolation. Retrovirology. 2012; 9(Suppl 2):P50. https://doi.org/10.1186/1742-4690-9-S2-P50
18. Roberts KE, Cushing PR, Boisguerin P, Madden DR, Donald BR. Computational design of a PDZ domain peptide inhibitor that rescues CFTR activity. PLoS computational biology. 2012; 8(4):e1002477. https://doi.org/10.1371/journal.pcbi.1002477 PMID: 22532795
19. King C, Garza EN, Mazor R, Linehan JL, Pastan I, Pepper M, et al. Removing T-cell epitopes with computational protein design. Proceedings of the National Academy of Sciences. 2014 Jun; 111(23):8577–8582. https://doi.org/10.1073/pnas.1321126111
20. Frey KM, Georgiev I, Donald BR, Anderson AC. Predicting resistance mutations using protein design algorithms. Proceedings of the National Academy of Sciences. 2010 Aug; 107(31):13707–13712. https://doi.org/10.1073/pnas.1002162107
21. Reeve SM, Gainza P, Frey KM, Georgiev I, Donald BR, Anderson AC. Protein design algorithms predict viable resistance to an experimental antifolate. Proceedings of the National Academy of Sciences. 2015 Jan; 112(3):749–754. https://doi.org/10.1073/pnas.1411548112
22. Lovell SC, Word JM, Richardson JS, Richardson DC. The penultimate rotamer library. Proteins: Structure, Function, and Bioinformatics. 2000; 40(3):389–408. https://doi.org/10.1002/1097-0134(20000815)40:3%3C389::AID-PROT50%3E3.0.CO;2-2
23. Pierce NA, Winfree E. Protein design is NP-hard. Protein Engineering. 2002 Oct; 15(10):779–782. https://doi.org/10.1093/protein/15.10.779 PMID: 12468711

24. Kingsford CL, Chazelle B, Singh M. Solving and analyzing side-chain positioning problems using linear and integer programming. Bioinformatics. 2005 Apr; 21(7):1028–1039. https://doi.org/10.1093/bioinformatics/bti144 PMID: 15546935

25. Street AG, Mayo SL. Computational protein design. Structure. 1999 May; 7(5):R105–9. https://doi.org/10.1016/S0969-2126(99)80062-8 PMID: 10378265

26. Jaramillo A, Wernisch L, Héry S, Wodak SJ. Automatic procedures for protein design. Combinatorial Chemistry & High Throughput Screening. 2001 Dec; 4(8):643–659. https://doi.org/10.2174/1386207013330724

27. Jin W, Kambara O, Sasakawa H, Tamura A, Takada S. De novo design of foldable proteins with smooth folding funnel: automated negative design and experimental verification. Structure. 2003 May; 11 (5):581–590. https://doi.org/10.1016/S0969-2126(03)00075-3 PMID: 12737823

28. Georgiev I, Keedy D, Richardson JS, Richardson DC, Donald BR. Algorithm for backrub motions in protein design. Bioinformatics. 2008 Jul; 24(13):i196–204. https://doi.org/10.1093/bioinformatics/btn169 PMID: 18586714

29. Georgiev I, Lilien RH, Donald BR. 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. Journal of Computational Chemistry. 2008 Jul; 29(10):1527–1542. https://doi.org/10.1002/jcc.20909 PMID: 18293294

30. Georgiev I, Donald BR. Dead-end elimination with backbone flexibility. Bioinformatics. 2007 Jul; 23(13): i185–94. https://doi.org/10.1093/bioinformatics/btm197 PMID: 17646295

31. Hallen MA, Keedy DA, Donald BR. Dead-end elimination with perturbations (DEEPer): A provable protein design algorithm with continuous sidechain and backbone flexibility. Proteins: Structure, Function, and Bioinformatics. 2013 Jan; 81(1):18–39. https://doi.org/10.1002/prot.24150

32. Gainza P, Roberts KE, Donald BR. Protein Design Using Continuous Rotamers. PLoS computational biology. 2012 Jan; 8(1):e1002335. https://doi.org/10.1371/journal.pcbi.1002335 PMID: 22279426

33. Silver NW, King BM, Nalam MNL, Cao H, Ali A, Kiran Kumar Reddy GS, et al. Efficient Computation of Small-Molecule Configurational Binding Entropy and Free Energy Changes by Ensemble Enumeration. Journal of chemical theory and computation. 2013 Nov; 9(11):5098–5115. https://doi.org/10.1021/ct400383v PMID: 24250277

34. Yanover C, Fromer M, Shifman JM. Dead-end elimination for multistate protein design. Journal of Computational Chemistry. 2007 Oct; 28(13):2122–2129. https://doi.org/10.1002/jcc.20661 PMID: 17471460

35. Fromer M, Yanover C. A computational framework to empower probabilistic protein design. Bioinformatics. 2008 Jul; 24(13):i214–i222. https://doi.org/10.1093/bioinformatics/btn168 PMID: 18586717

36. Fromer M, Yanover C. Accurate prediction for atomic-level protein design and its application in diversifying the near-optimal sequence space. Proteins: Structure, Function, and Bioinformatics. 2009 May; 75 (3):682–705. https://doi.org/10.1002/prot.22280

37. Kamisetty H, Xing EP, Langmead CJ. Free Energy Estimates of All-Atom Protein Structures Using Generalized Belief Propagation. dxdoiorg. 2008 Sep; 15(7):755–766.

38. Kuhlman B, Baker D. Native protein sequences are close to optimal for their structures. Proceedings of the National Academy of Sciences of the United States of America. 2000 Sep; 97(19):10383–10388. https://doi.org/10.1073/pnas.97.19.10383 PMID: 10984534

39. Gainza P, Nisonoff HM, Donald BR. Algorithms for protein design. Current Opinion in Structural Biology. 2016; 39:16–26. https://doi.org/10.1016/j.sbi.2016.03.006 PMID: 27086078

40. Gordon DB, Mayo SL. Branch-and-Terminate: a combinatorial optimization algorithm for protein design. Structure. 1999 Sep; 7(9):1089–1098. https://doi.org/10.1016/S0969-2126(99)80176-2 PMID: 10508778

41. Leach AR, Lemon AP. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. Proteins: Structure, Function, and Bioinformatics. 1998 Nov; 33(2):227–239. https://doi.org/10.1002/(SICI)1097-0134(19981101)33:2%3C227::AID-PROT7%3E3.0.CO;2-F

42. Georgiev I, Lilien RH, Donald BR. Improved Pruning algorithms and Divide-and-Conquer strategies for Dead-End Elimination, with application to protein design. Bioinformatics. 2006 Jul; 22(14):e174–e183. https://doi.org/10.1093/bioinformatics/btl220 PMID: 16873469

43. Gainza P, Roberts KE, Georgiev I, Lilien RH, Keedy DA, Chen CY, et al. OSPREY: protein design with ensembles, flexibility, and provable algorithms. Methods in enzymology. 2013; 523:87–107. https://doi.org/10.1016/B978-0-12-394292-0.00005-9 PMID: 23422427

44. Lilien RH, Stevens BW, Anderson AC, Donald BR. 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 Jul; 12 (6):740–761. https://doi.org/10.1089/cmb.2005.12.740 PMID: 16108714

45. Jones DT. De novo protein design using pairwise potentials and a genetic algorithm. Protein Science. 1994 Apr; 3(4):567–574. https://doi.org/10.1002/pro.5560030405 PMID: 8003975

46. 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 Jun; 239 (2):249–275. https://doi.org/10.1006/jmbi.1994.1366 PMID: 8196057

47. Desjarlais JR, Handel TM. De novo design of the hydrophobic cores of proteins. Protein Science. 1995 Oct; 4(10):2006–2018. https://doi.org/10.1002/pro.5560041006 PMID: 8535237

48. Jiang X, Farid H, Pistor E, Farid RS. A new approach to the design of uniquely folded thermally stable proteins. Protein Science. 2000 Feb; 9(02):403–416. https://doi.org/10.1110/ps.9.2.403 PMID: 10716193

49. 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 Bioinformatics. 2002 Jul; 48(1):31–43. https://doi.org/10.1002/prot.10131

50. Kortemme T, Morozov AV, 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 Feb; 326(4):1239–1259. https://doi.org/10.1016/S0022-2836(03)00021-4 PMID: 12589766

51. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, Jacak R, et al. ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. In: Methods in Enzymology. Methods in Enzymology; 2011. p. 540–574.

52. Privett HK, Kiss G, Lee TM, Blomberg R, Chica RA, Thomas LM, et al. Iterative approach to computational enzyme design. Proceedings of the National Academy of Sciences. 2012 Mar; 109(10):3790–3795. https://doi.org/10.1073/pnas.1118082108

53. Canutescu AA, Shelenkov AA, Dunbrack RL. A graph-theory algorithm for rapid protein side-chain prediction. Protein Science. 2003 Sep; 12(9):2001–2014. https://doi.org/10.1110/ps.03154503 PMID: 12930999

54. Leaver-Fay A, Kuhlman B, Snoeyink J. An adaptive dynamic programming algorithm for the side chain placement problem. Pacific Symposium on Biocomputing. 2005; 10:16–27.

55. Xu J, Berger B. Fast and accurate algorithms for protein side-chain packing. Journal of the ACM (JACM). 2006 Jul; 53(4):533–557. https://doi.org/10.1145/1162349.1162350

56. Peng J, Hosur R, Berger B, Xu J. iTreePack: Protein Complex Side-Chain Packing by Dual Decomposition. arXiv:150405467 [q-bioBM]. 2015; Available from: https://arxiv.org/abs/1504.05467.

57. Krivov GG, Shapovalov MV, Dunbrack RL. Improved prediction of protein side-chain conformations with SCWRL4. Proteins: Structure, Function, and Bioinformatics. 2009 Dec; 77(4):778–795. https://doi.org/10.1002/prot.22488

58. Xu J. Rapid Protein Side-Chain Packing via Tree Decomposition. In: Research in Computational Molecular Biology, Lecture Notes in Computer Science. vol. 3500. Proceedings of the Annual International Conference on Research in Computational Molecular Biology (RECOMB), Cambridge, May 14-18, 2005. Springer-Verlag (Berlin); 2005. p. 423–439.

59. Jou JD, Jain S, Georgiev IS, Donald BR. BWM*: A Novel, Provable, Ensemble-based Dynamic Programming Algorithm for Sparse Approximations of Computational Protein Design. J Comput Biol. 2016; 23(6):413–424. https://doi.org/10.1089/cmb.2015.0194 PMID: 26744898

60. Zhou Y, Wu Y, Zeng J. Computational Protein Design Using AND/OR Branch-and-Bound Search. J Comput Biol. 2016; 23(6):439–451. https://doi.org/10.1089/cmb.2015.0212 PMID: 27167301

61. Georgiev I, Lilien RH, Donald BR. 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. Journal of Computational Chemistry. 2008 Jul; 29(10):1527–1542. https://doi.org/10.1002/jcc.20909 PMID: 18293294

62. Roberts KE, Gainza P, Hallen MA, Donald BR. Fast gap-free enumeration of conformations and sequences for protein design. Proteins: Structure, Function, and Bioinformatics. 2015 Oct; 83 (10):1859–1877. https://doi.org/10.1002/prot.24870

63. Hubbard SJ, Thornton JM. NACCESS: Computer Program, Department of Biochemistry and Molecular Biology, University College London. 1993.

64. Shah PS, Hom GK, Ross SA, Lassila JK, Crowhurst KA, Mayo SL. Full-sequence Computational Design and Solution Structure of a Thermostable Protein Variant. Journal of Molecular Biology. 2007; 372(1):1–6. Available from: http://dx.doi.org/10.1016/j.jmb.2007.06.032. PMID: 17628593

65. Dantas G, Kuhlman B, Callender D, Wong M, Baker D. A large scale test of computational protein design: folding and stability of nine completely redesigned globular proteins. Journal of molecular biology. 2003; 332(2):449–460. https://doi.org/10.1016/S0022-2836(03)00888-X PMID: 12948494

66. Mou Y, Huang PS, Hsu FC, Huang SJ, Mayo SL. Computational design and experimental verification of a symmetric protein homodimer. Proceedings of the National Academy of Sciences of the United States of America. 2015; 112(34):10714–10719. Available from: http://dx.doi.org/10.1073/pnas.1505072112. PMID: 26269568

67. Dantas G, Corrent C, Reichow SL, Havranek JJ, Eletr ZM, Isern NG, et al. High-resolution Structural and Thermodynamic Analysis of Extreme Stabilization of Human Procarboxypeptidase by Computational Protein Design. Journal of Molecular Biology. 2007; 366(4):1209–1221. Available from: http://dx.doi.org/10.1016/j.jmb.2006.11.080. PMID: 17196978

68. Hallen MA, Gainza P, Donald BR. Compact Representation of Continuous Energy Surfaces for More Efficient Protein Design. Journal of Chemical Theory and Computation. 2015; 11(5):2292–2306. https://doi.org/10.1021/ct501031m PMID: 26089744

69. Word JM, Lovell SC, LaBean TH, Taylor HC, Zalis ME, Presley BK, et al. Visualizing and quantifying molecular goodness-of-fit: small-probe contact dots with explicit hydrogen atoms. Journal of molecular biology. 1999 Jan; 285(4):1711–1733. https://doi.org/10.1006/jmbi.1998.2400 PMID: 9917407

70. Hallen MA, Jou JD, Donald BR. LUTE (Local Unpruned Tuple Expansion): Accurate Continuously Flexible Protein Design with General Energy Functions and Rigid Rotamer-Like Efficiency. J Comput Biol. 2016; Epub ahead of print. https://doi.org/10.1089/cmb.2016.0136

71. Fleishman S, Baker D. Role of the Biomolecular Energy Gap in Protein Design, Structure, and Evolution. Cell. 2012; 149(2). https://doi.org/10.1016/j.cell.2012.03.016 PMID: 22500796