Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Current Opinion in
Structural Biology

Algorithms for protein design

Pablo Gainza[1], Hunter M Nisonoff[1] and Bruce R Donald[1,2,3]

Computational structure-based protein design programs are becoming an increasingly important tool in molecular biology. These programs compute protein sequences that are predicted to fold to a target structure and perform a desired function. The success of a program's predictions largely relies on two components: first, the input biophysical model, and second, the algorithm that computes the best sequence(s) and structure(s) according to the biophysical model. Improving both the model and the algorithm in tandem is essential to improving the success rate of current programs, and here we review recent developments in algorithms for protein design, emphasizing how novel algorithms enable the use of more accurate biophysical models. We conclude with a list of algorithmic challenges in computational protein design that we believe will be especially important for the design of therapeutic proteins and protein assemblies.

Addresses
[1] Department of Computer Science, Duke University, Durham, NC, United States
[2] Department of Biochemistry, Duke University Medical Center, Durham, NC, United States
[3] Department of Chemistry, Duke University, Durham, NC, United States

Corresponding author: Donald, Bruce R (brd+cosb15@cs.duke.edu)

Current Opinion in Structural Biology 2016, **39**:16–26

This review comes from a themed issue on **Engineering and design**

Edited by **Dan Tawfik** and **Raghavan Varadarajan**

http://dx.doi.org/10.1016/j.sbi.2016.03.006

0959-440X/© 2016 Elsevier Ltd. All rights reserved.

Introduction

Computational structure-based protein design is one of the most promising tools for engineering proteins with new functions, including the development of therapeutic proteins and protein assemblies [1–4]. Despite important successes, however, many of the current computational protein design tools often have low success rates, and designed proteins sometimes fail to achieve the functional properties of native proteins. New advances in protein design methodologies are required to improve the functional properties and success rate of computationally designed proteins.

The problem of engineering a new functional protein using computational methods is typically divided in two challenging stages. The first stage is selecting a *target* tertiary/quaternary protein *fold* that will be *designed* for a specific function. Often the selected fold is one that performs a similar function and can later be redesigned to a new one [5–8,9••]. In other cases, a protein that has a completely different function is used as a *scaffold* and repurposed for a new one [10–12]. And, increasingly, protein engineers are incorporating empirical folding and structural principles [13–18] to design proteins from scratch (*de novo* design) [12–19]. The second stage is to design a protein sequence, together with side chain rotamers and residue conformations [20•], that will adopt the overall target fold (often allowing some *backbone flexibility* [13–18,20•,21,22•,23•,24–26,27••]) and perform a desired function (e.g., binding with specificity to a target molecule). This latter stage has been historically referred to as *protein design* [28]. Many computational protein engineering protocols implement different variations of these two stages, and these have resulted in many successfully engineered new proteins [5–8,9••,10–19,29–35]. Here we focus on protein design.

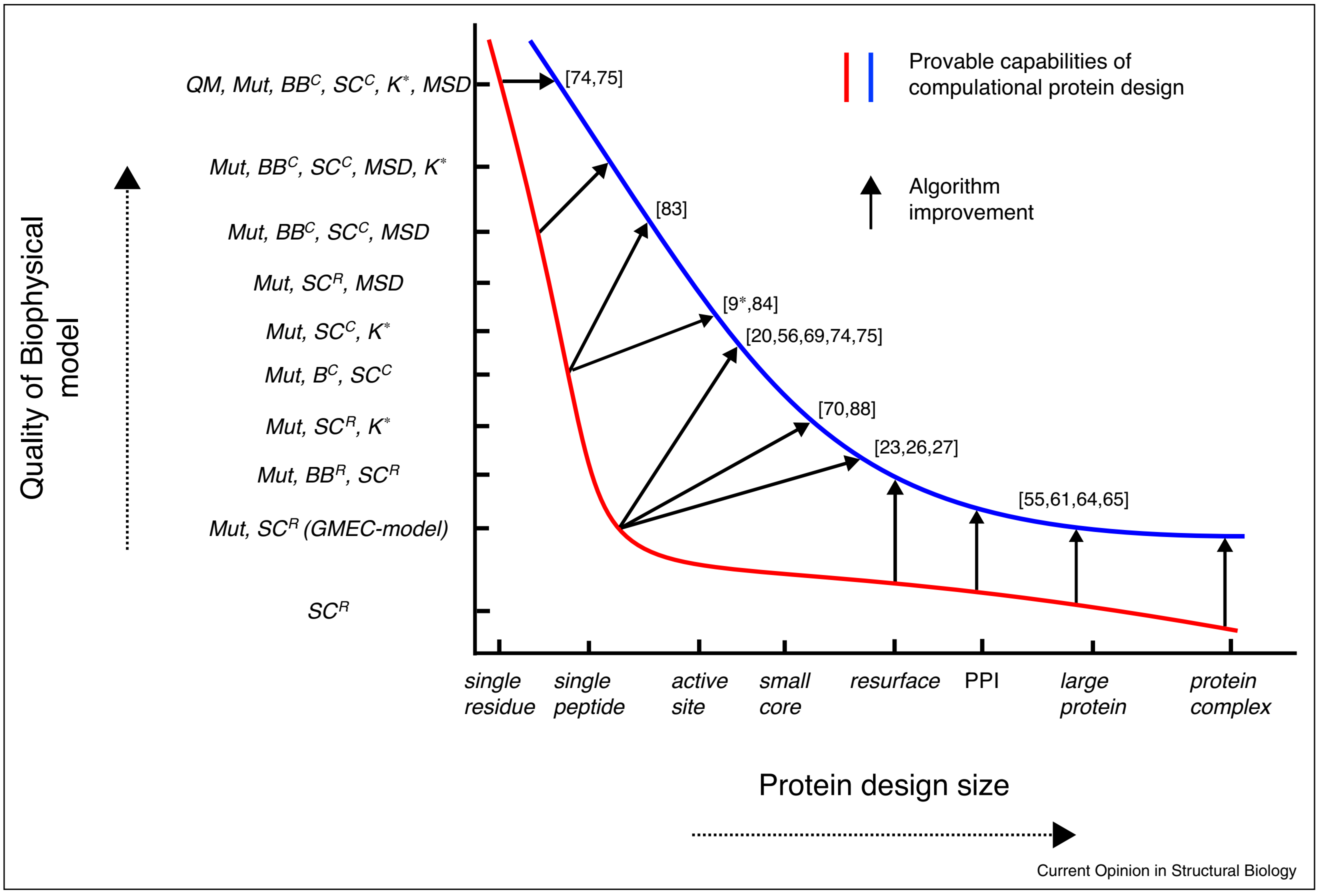
Protein design can be formulated as a well-defined computational problem by reducing it to an optimization over a family of parameterized structure-based protein redesign problems. In this well-posed version, an optimization *algorithm* (also known as a search algorithm) computes and outputs the best protein amino acid sequence(s) and structure(s) in a space defined by a *biophysical input model*. This biophysical model defines the sequence and structural search space (e.g., template input structure, the allowed flexibility, the amino acid sequences allowed, etc.), the optimization objective (e.g., single state, multi-state, ensemble-based, etc.), and the scoring potential for protein energetics (i.e., the energy function [36,37]). To our knowledge, all structure-based protein design programs conform to this formulation [13,38–42]. For example, one of the most frequently used biophysical models for backbone flexibility in Rosetta [13] consists of a target structure, an ensemble of allowed backbone moves (e.g., backbone dihedral changes), a rotamer library, the energy function, and a predefined sequence space [13,43]. This biophysical model describes a space which is then searched using Rosetta's *iterative relaxation/design* algorithm [13]. Iterative relaxation/design iteratively intercalates two steps: first, a design step, where the backbone is held constant while the conformations and amino acid identities of the side chains are optimized; and second, a relaxation step, where the sequence is held constant, while the backbone and side chains are optimized using a hybrid stochastic/gradient descent optimization [13,44,45].

The accuracy of a computational protein design relies largely on the biophysical model and it is thus highly desirable to improve this model. Biophysical model improvements, however, often come at the cost of exponentially increasing the computational complexity of the search problem. Since computational hardware cannot grow at the same rate, the only practical solution to search more complex biophysical models is through novel algorithms. Therefore, substantial improvement in computational protein design necessitates the development of novel algorithms (see Figure 1). For this reason, we focus on algorithms for protein design, and review those that we believe represent new algorithmic breakthroughs and that have potential for the design of therapeutic proteins and protein assemblies. We focus on developments since 2010 (foundational and earlier algorithms are discussed in [37,46,47•]) in four areas: optimization algorithms for protein design, algorithms to search improved flexibility models, multi-state design, and ensemble-based design. Because of constraints on the length of this survey, we exclude related algorithms that are important for therapeutic and assembly protein design that have also been highly productive recently, such as docking algorithms (for a review see [48]), scaffold search algorithms (e.g., [49,50]), and algorithms to optimize libraries for *in vitro* evolution of designed proteins (e.g., [51,52]).

Provable versus heuristic algorithms

Protein design, like many other problems in the field of computational structural biology, belongs to a hard class of computational problems [47•]. Consider, for example, a simple yet common biophysical model for the protein

Figure 1



Cartoon of the equicomplexity curves for computational protein design. Curves show a trade-off between biophysical model quality and protein design size given fixed computational resources (time and space). Algorithm improvements (black arrows) expand the boundaries of these trade-offs to allow higher quality biophysical models for larger protein design sizes. The *y*-axis maps the multi-dimensional input biophysical models that can be potentially used in protein design to a one-dimensional axis. The axis is ordered from the simplest models at the bottom (with a pairwise molecular-mechanics energy function) to the most advanced models at the top. Several examples of input biophysical models are shown: *Mut*: All amino acid mutations allowed at all designed residue positions; SC^R: discrete side-chain flexibility; BB^R: discrete backbone flexibility; SC^C: continuous side-chain flexibility; BB^C: continuous backbone flexibility; *MSD*: multi-state design; K^*: ensemble-based free energy calculations; *QM*: the most advanced energy function models. *PPI*: Protein–protein interaction. Example references of algorithms that improve the curves and that are cited in this review are shown next to each arrow. *Ref [9••] corresponds to the design of a peptide inhibitor of a PPI in the *x*-axis.