Molecular flexibility in computational protein design: an algorithmic perspective

Younes Bouchiba, Juan Cortés, Thomas Schiex, Sophie Barbe

Molecular Flexibility in Computational Protein Design: an Algorithmic Perspective

Younes Bouchiba[a,b], Juan Cortés[a], Thomas Schiex[c], Sophie Barbe[b]

[a] *LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France*
[b] *TBI, Université de Toulouse, CNRS, INRAE, INSA, ANITI, Toulouse, France*
[c] *Université Fédérale de Toulouse, ANITI, INRAE, UR 875, Toulouse, France*

Abstract

Computational protein design (CPD) is a powerful technique for engineering new proteins, with both great fundamental implications and diverse practical interests. However, the approximations usually made for computational efficiency, using a single fixed backbone and a discrete set of side-chain rotamers, tend to produce rigid and hyper-stable folds that may lack functionality. These approximations contrast with the demonstrated importance of molecular flexibility and motions in a wide range of protein functions. The integration of backbone flexibility and multiple conformational states in CPD, in order to relieve the inaccuracies resulting from these simplifications and to improve design reliability, are attracting increased attention. However, the greatly increased search space that needs to be explored in these extensions defines extremely challenging computational problems. In this review, we outline the principles of CPD and discuss recent effort in algorithmic developments for incorporating molecular flexibility in the design process.

Keywords: Computational protein design, Multistate design, Backbone perturbations, Continuous flexibility, Provable and heuristic algorithms.

Introduction

Computational structure-based protein design (CPD) seeks to identify sequences that adopt desired structures and perform targeted functions. CPD has become an increasingly important tool to engineer proteins for biotechnological and medical applications and also to test our understanding of the biophysical and functional mechanisms of naturally evolved proteins. It has produced striking successes, including the engineering of proteins with new topologies, improved thermostability, increased binding affinity, altered ligand specificity and new activities [36]. Despite these important successes, to improve the reliability and generalize the design of proteins with new and optimized functionalities, CPD faces several challenges. The main one lies in the exploration of the high-dimensional sequence and conformation space accessible to proteins and the

Preprint submitted to Protein Engineering, Design and Selection, DOI: 10.1093/protein/gzab011

discrimination of solutions. The high complexity of this task leads to approximations. The most usual one is to consider a single backbone template state of the protein with fixed 3D coordinates while sampling amino acid side-chain conformations among a finite set of discrete states (rotamers) [26]. This simplified instance of the design problem is useful for computational efficiency. However, such simplifications may compromise design success.

In this paper, we review recent advances that try to better account for the inherent flexibility of proteins. Neglecting flexibility in CPD approaches leads to a suboptimal formulation of the problem: the continuous nature of side-chain angles means that tiny adjustments could have avoided steric clashes; the plasticity of the backbone may allow to accommodate changes in amino acid properties [61, 19] (Figure 1). The lack of protein flexibility in CPD may thus neglect a significant portion of the sequence space which is otherwise accessible to properly folded and functional protein and thus introduce some biases in sequence selection. Moreover, molecular flexibility is crucial for several of the many functions of proteins [60]. Conformational changes of the protein backbone are essential for the functioning of molecular rotors, switches and pumps [71]. They also play key roles in molecular recognition [8] and enzyme catalytic process [5]. Structural flexibility, ranging from small fluctuations to large-scale rearrangements, is thus important to consider for protein design.

We review each of these limitations and how they have been partially addressed by generalized variants of the basic CPD problem, (1) by using continuous rotamers, (2) by allowing adjustments in the protein backbone or (3) by considering several possible backbone states. These extensions can, and ideally should, be combined. They still need to be improved, to ultimately target the design of the dynamic properties of proteins. But the computationally very challenging problems that they define often impose strict computational restrictions on the size of the systems that can be considered. In the conclusion, we finally discuss recent sequence-based Machine Learning approaches to CPD [20], which implicitly consider backbone flexibility by keeping the structure latent.

Fundamental formulation and algorithms

Problem formulation

The canonical problem of protein design is to produce a sequence of amino acids that will fold into a protein that performs a desired function. That is, we want to find a sequence s that stably adopts a conformation that will perform a given function. The stability can be captured by a real-valued energy function $E(s, \theta, \chi)$ that describes the conformational energy of sequence s when it adopts the three-dimensional conformation defined by the backbone torsion angles θ and amino acid side chain torsion angles χ. Equipped with these, computational protein design (CPD) reduces to the problem of identifying one or more designs (s^*, θ^*, χ^*) that adopt the target conformation θ^*, *i.e.*, such that $E(s^*, \theta^*, \chi^*) = \min_{\theta,\chi} E(s^*, \theta, \chi)$. Beyond stability, the final protein function is often assumed to be a consequence of its adopted conformation. It can be more

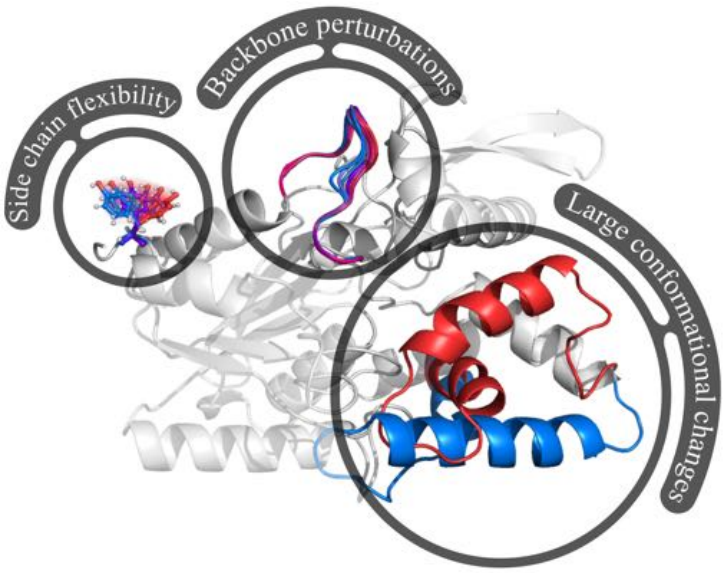

Figure 1: Three different levels of protein flexibility described in the paper: side chains can move continuously, the backbone may slightly fluctuate, and some regions of the proteins may shift between distinct conformational states.

explicitly represented by including additional terms in the minimized energy function, defining a final objective "score function". It can also be captured by constraining the search to a subspace of function-favorable conformations. This formulation can be generalized to protein complexes using additional rigid body degrees of freedom (set by docking). Given the continuous nature and high dimensionality of θ, χ, the discrete nature of s and the non-convex nature of E, the CPD problem [10] was reduced to a fully discrete problem where the backbone is rigid (θ being fixed), the side-chain angles χ are discretized into a set of statistically significant conformations (or rotamers) and the energy function is approximated by a sum of terms capturing interactions between at most two bodies. The resulting "rigid-backbone, discrete rotamers, pairwise decomposable energy" design problem requires to identify the "Global Minimum Energy Conformation" (GMEC) (s^*, χ^*) that minimizes the decomposable energy E on the backbone (θ^*), a problem that remains computationally very challenging (NP-hard [49]). This formulation, however, makes it possible to precompute the terms that contribute to $E(s, \theta^*, \chi)$ for every possible rotamer and pair of rotamers of every residue (and pair of residues). These precomputed terms define the so-called energy matrix [53], making energy computations extremely efficient.

At his core, this widely studied and adopted formulation is flawed by the flexibility of protein backbone, which implies that even if (s^*, χ^*) minimizes the energy of the rigid target backbone, there may be different backbone states (θ_i) that could decrease the energy of s^* even further. This single target back-

bone formulation is also intrinsically incapable to account for the importance of protein dynamics in many of their functions, including their ability to act as switches that can adopt several stable conformations. We ignore in this paper the additional complexities (and flexibility) that can be raised by the presence of ligands and co-factors. In practice, they are often addressed by the same approaches that try to capture protein flexibility.

Heuristic and provable algorithms

Because of its computational complexity, the rigid-backbone problem has been tackled using various heuristics, including greedy local search [45] and stochastic optimization approaches such as Monte Carlo (MC) [35] or genetic algorithms [50]. In practical settings, these stochastic methods only offer asymptotic guarantees of convergence to the GMEC and detecting convergence reliably can only be achieved using provable methods. Indeed, in finite time, these routines may remain trapped in local minima far from the global one. To try to avoid this problem, multiple independent finite runs are performed for a heuristically set number of times, with the hope that all the low energy landscape will be covered. However, even on small redesign problems, non negligible energy gaps may still exist between the actual GMEC energy and what established stochastic optimization algorithms can sometimes produce [68, 57]. On a set of 100 test protein designs, Simoncini *et al.* [57] showed that Simulated Annealing, as implemented in Rosetta [37], could fail to identify the GMEC for most of the design problems, even after one thousand repeats. This means that there is always an (unknown) limit on the size of systems for which a solution of sufficiently low energy can be found with confidence with stochastic methods. While these limitations are somewhat mitigated by the fact that the optimized energy function is only approximate and does not need to be absolutely optimized, in the end, when a design (s, χ) experimentally fails, the possibility that the optimization algorithm is the source of the failure remains.

Provable algorithms guarantee that the GMEC will be returned in finite time, ensuring that discrepancies between CPD predictions and experimental results come exclusively from modeling inadequacies. Once dominated by algorithms combining the Dead-End-Elimination theorem with Best-First A* search [38], the state of the art in provable rigid backbone protein design is now defined by automated reasoning algorithms, initially introduced in artificial intelligence [26]. They rely on the encoding of the pairwise energy matrix as a so-called Cost Function Network (CFN) [63, 62]. They are often able to solve problems with more than 100 mutable residues in reasonable time, while remaining provable. In many situations, and thanks to their increased efficiency, it becomes very comfortable to obtain a guaranteed result even with limited computational resources. Surprisingly, these algorithms can even be significantly faster than heuristic approaches, as they know when the global optimum is reached. On a large set of CPD benchmarks, the CFN prover `toulbar2` has been shown to offer speed-ups of several orders of magnitude compared to other state-of-the-art provable methods, giving access to guaranteed GMECs for design problems that were previously out of reach of provable

algorithms [63, 1, 57, 62, 64]. The energy function being only approximate, it is also possible to ask for weaker, easier to produce, proofs that just guarantee a bounded distance to the GMEC. On a benchmark set of 99 design problems and especially for full protein redesign, randomized variants of CFN algorithms [6] often provided solutions with lower energies than sophisticated Monte Carlo replica exchange methods [66], although without a complete proof of optimality. These progresses are important when more challenging design problem formulations accounting for flexibility need to be tackled and some of these algorithms have made their way into established CPD software [29].

Algorithms considering a single input structure

Side chain flexibility

A first missing source of flexibility in the conventional CPD formulation lies in the continuous nature of side chain rotations (Figure 1). The choice of a limited set of discrete conformations can lead to situations where a given rotamer will not fit because of steric clashes that could be removed by tiny continuous adjustments. This situation is often dealt with heuristically, by lowering the contribution of the repulsive van der Waals term in the score function (called soft variants in Rosetta) or continuous pre-minimization of the contribution of each rotamer to the score function in one and two bodies terms [18]. This last approach may however lead to a representation where a rotamer is assumed to adopt different continuous positions in different two-bodies terms. This inconsistency is avoided in Osprey [29], where the continuous aspect of the side-chain angles is explicitly dealt with: each rotamer represents a subspace of continuous side chain angles, and terms in the score function matrix become lower bounds on their final contributions, assuming ideal minimized conformations as above. As the design progresses, these lower bounds are tightened by post-hoc minimization until an optimal continuous side chain design is found and proven. In this sense, Osprey offers provable continuous rotamer design, although this relies on the assumption that the score function in the continuous subspace covered by each rotamer can be effectively minimized (e.g., is convex in the rotamer subspace). LUTE is an alternative approach that machine learns a decomposed energy function of discrete rotamers on continuously minimized samples (or other non pairwise decomposable energy functions), that can be later provably optimized as in the discrete rotamer case [28]. The quality of the fit of the learned function can be empirically estimated through residuals.

Side chain flexibility also induces entropic effects: a backbone conformation compatible with a large number of side chain conformations can have higher probability to be observed (i.e., higher stability at room temperature) than one corresponding to a single conformation at the GMEC. This effect can be accounted for by explicit free energy computations in the context of one sequence (a computationally expensive #P-hard problem). Again, Monte Carlo methods [56] compete here with provable methods [67, 32]. For design, optimizing the explicit side-chain free energy defines a computationally even more challenging problem (closely related to the $\mathsf{NP}^{\mathsf{PP}}$-complete Marginal MAP problem

in graphical models [48]). It can nevertheless be tackled by Osprey, at least on small design problems [46], with recent computational speedups obtained in MARK* [32], using LUTE and structural similarities to accelerate energy integration.

Backbone flexibility

A simple approach to introduce backbone flexibility in CPD protocols consists in interleaving sequence optimization and backbone relaxation within an iterative algorithm, usually called design-and-relax (D&R). This type of approach was first proposed using a MC-based minimization procedure to relax the protein structure after the resolution of a fixed-backbone CPD problem [37, 43]. A variant of this method performs a stronger relaxation by applying Rosetta FastRelax method instead of MC-based energy minimization [34, 65]. In our knowledge, this type of iterative D&R approaches have only been proposed in the framework of stochastic algorithms, although, in principle, they are also applicable to provable algorithms. Provable approaches however require to recompute the energy matrix at each iteration, which can be expensive.

Heuristic CPD algorithms can also take into account backbone flexibility by interleaving side-chain moves/mutations and local backbone (or ligand's pose) moves at each step of the stochastic optimization process. This type of method is called Coupled-Moves in Rosetta [47, 39]. The first version of the Coupled-Moves method [47] used the Backrub technique [16], illustrated in Figure 2, to perform local backbone moves. A variant of the Coupled-Moves method proposed by Loshbaugh and Kortemme [39] replaced Backrub moves by robotics-inspired KIC moves [40] (Figure 2), showing enhanced overall performance. KIC moves are usually applied to larger flexible fragments such as loops, and give access to larger-amplitude motions of the backbone. After the application of a local backbone move, Coupled-Moves applies a strategy based on Boltzmann probabilities to select the most promising side-chain move or mutation. This requires to recompute interaction energies for all the residues implied in the backbone move, which may have a significant cost when the overall process is repeated a larger number of times. Note that local Backrub-like moves are also used in SHADES, which also constrains the sequence search space using natural spatially-close non-contiguous amino acid patterns for sequence determination [58].

Provable algorithms have also been extended for including backbone perturbations. Actually, the first algorithm to incorporate Backrub moves into protein design was a provable algorithm [21]. More recently, building on the DEE/A* algorithm, Donald and coworkers developed provable approaches to incorporate continuous backbone flexibility in the design process [27, 25]. As for continuous side chain flexibility [19], backbone flexibility can be handled by computing bounds on the energies of the backbone continuous internal coordinates in a vicinity around the starting structure backbone. The main difficulty relies on the choice of coordinates to appropriately represent the flexibility of the backbone in the surrounding of the mutations, without leading to an intractable search space. Indeed, unless they are very specific, local backbone changes may

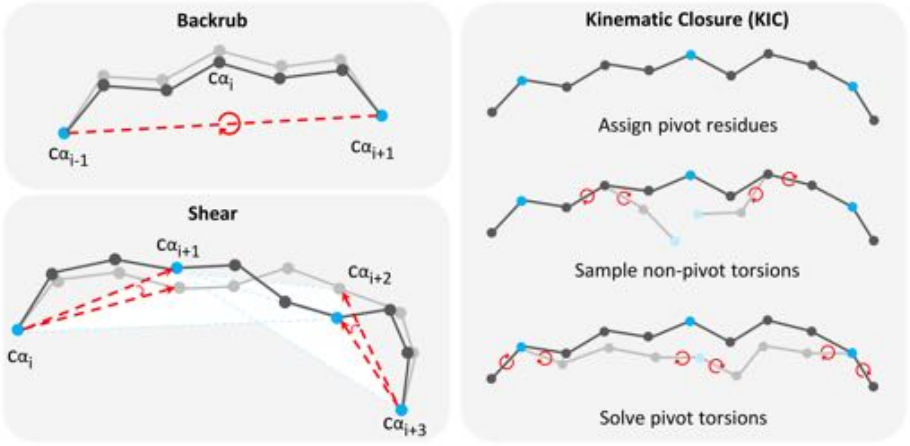

Figure 2: Several methods to perform local backbone perturbations. **Backrub** moves [16] perform a subtle rotational motion of a backbone fragment around the axis defined by the Cα atoms at both ends of the fragment. They usually involve three consecutive Cα atoms, as illustrated here, but can be applied to larger fragments. **Shear** moves also perform slight local rearrangements inspired from the observation of crystallographic alternate structures [61]. A Shear move is defines from 4 Cα atoms. Cα_{i+1} is slightly rotated about Cα_{i+1} in the plane Cα_i-Cα_{i+1}-Cα_{i+2}, and Cα_{i+4} is rotated about Cα_{i+3} in the plane Cα_{i+2}-Cα_{i+3}-Cα_{i+4}, while preserving the distance between Cα_{i+2} and Cα_{i+3}. **KIC** moves, inspired from robotics, give access to a broader conformational space. They are illustrated for a small fragment of 5 residues here. Two pivot Cα atoms define the ends of the sampled fragment, and a third Cα atoms is selected within this fragment. The backbone ϕ, ψ angles of the rest of the residues in the fragment are perturbed, either using values sampled from a Gaussian distribution around their current value or using statistical information extracted from experimentally determined polypeptide structures [59]. Then, the ϕ, ψ angles of the three pivot residues are determined using an inverse kinematics solver [9].

propagate along the protein to a distant region. To avoid such propagated changes, the dead-end-elimination algorithm with perturbations (DEEPer) uses small backbone perturbations such as Backrub or Shear [27] (Figure 2). To enable a larger degree of continuous motion of a backbone fragment, a new type of backbone coordinate system was introduced by Hallen and Donald [25]. The approach allows to compute all atomic coordinates as a function of the novel degrees of freedom, by calculating Coordinates of Atoms by Taylor Series (CATS). DEEPer and CATS can be used in combination with continuous side-chain flexibility [27, 25].

Algorithms considering several input structures

The previously described methods allow for local conformational search, considering only one backbone structure as input (algorithms referred as Single State Design (SSD)). When the final backbone conformation is uncertain or when it is explicitly desirable to simultaneously stabilize (or destabilize) several states, Multistate Design (MSD) makes it possible to simultaneously consider

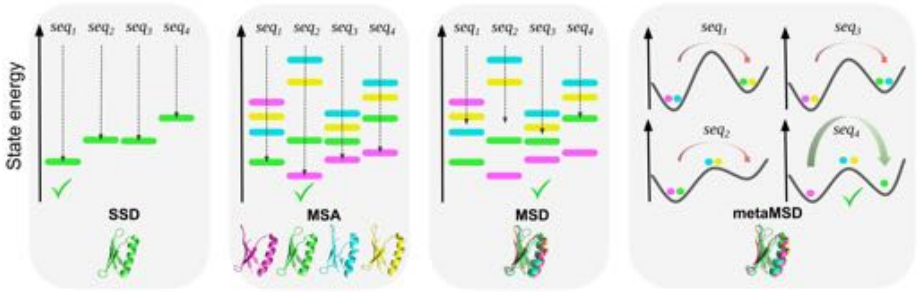

Figure 3: Sequences scoring and ranking according to CPD methods using one or several backbone template(s) as input. In **SSD**, a single state (green) is used to score and rank sequences ($seq_1, seq_2, seq_3, seq_4$) according to their energy, which defines the fitness function (dashed arrow): the best sequence is seq_1. In **MSA**, **MSD** and ***meta*-MSD**, an ensemble of (here) four backbone states (magenta, green, cyan and yellow) is used to score and rank sequences. In **MSA**, the fitness of each sequence (dashed arrow) is computed using the minimum of the sequence energies in each state. The sequence seq_2 is ranked first as it has the best energy on the magenta backbone state. In **MSD**, the fitness of each sequence (dashed arrow) is defined by the (weighted) sum of the sequence energies over all backbone states: the best sequence is seq_3. ***meta*-MSD** relies on : (i) the classification of each designed state into three microstates (major (left), transition and minor (right)) according to their structural features; (ii) the analysis of an energy profile with relative energy differences between states determining whether sequences are expected to transit between the states. seq_4, being the only sequence that stably populates the three states and with a low transition barrier, is ranked as the best sequence.

independent conformations of the protein (or complex). These conformations can be geometrically close to each other, but can also represent large, functionally important conformational changes. They can typically be extracted from natural structures or models of the same protein, from iterated local backbone perturbations using the aforementioned move classes (Figure 2), or from MD simulations. In MSD, for a fixed sequence, side-chains will still tend to organize themselves following a minimum energy (or optimal) conformation. The aim of MSD is then to find a sequence that optimizes a combination of these optimal energies over the various input backbones, often denoted as the fitness of the sequence. Several fitness functions can be considered depending on the design target (Figure 3). When the aim is to stabilize any of the considered conformational states, the Boltzmann-weighted average of the energies (defined as the sum of optimal energies, weighted by their Boltzmann probabilities) in each state may be an attractive criteria [14]. Because this gives an exponential advantage to the backbone with lowest energy, the computation of this fitness has been approximated by the minimum optimal energy [12, 33, 69] defining what is called "multistate analysis" (MSA) [13]. MSA showed interesting results when combined with local backbone fluctuation search algorithms for each state. Such multi-state design has been applied successfully in several protein design cases [30, 2, 4, 14, 5]. When, instead, the aim is to design a sequence that fits several conformational states that must be adopted for the targeted function (e.g., states defining conformational switches), it is important that the energies

of all states contribute to the definition of the fitness. In this case, it is usual to optimize the average of the optimal energies over all states, a problem we denote as Σ-MSD [69, 54]. Computationally speaking, SSD, MSA and Σ-MSD define "positive design" problems that try to stabilize desirable states, and are just NP-complete [69].

When one seeks a sequence that also needs to destabilize some undesirable states (e.g. to optimize specificity or a bound vs. unbound state), a sequence that maximizes the difference in optimal energies between desirable and undesirable states is often sought. These "negative design" problems define computationally far more challenging $\mathsf{NP}^{\mathsf{NP}}$-complete problems [69]. This is consistent with the fact that solving a negative multi-state design problem requires to explore the sequence space, and for each sequence and state, to explore its conformation space. However, provable algorithms can exploit properties allowing to prune both of these spaces, leading to algorithms that explore only a minute fraction of these spaces.

Similarly to what LUTE later used with continuous rotamers [28], CLASSY [44] exploits a decomposable energy on sequences, which has been machine learned on side-chain-optimized samples. This solves the problem of computing "optimal" energies for a sequence in a given state, bringing the problem back to NP-completeness (at the cost of the approximations made in the learned energy model). Provable Integer Linear Programming optimization techniques can then directly be used on the learned function. Without these approximations, COMETS relies on an extension of DEE to a multistate situation [24]. Due to the extreme problem complexity, COMETS remains usable only on relatively small design spaces. Given their efficiency for SSD, CFN-based algorithms solving the "Weighted Constraint Satisfaction Problem" have been used by iCFN [33], an MSD tool which allows for larger design spaces using discrete rotamers. iCFN can perform both positive and negative design, but cannot use the average energy criteria of Σ-MSD, making it less suitable when large conformational changes must also be considered.

Lately, by using a simple problem reduction, POMP$^{\mathrm{d}}$ emerged as one of the most efficient provable positive MSD algorithm, outperforming iCFN for discrete rotamers positive design problems, whether with an MSA or Σ-MSD fitness [69].

Finally, when the objective is to design proteins that dynamically exchange between conformations, sequences must be designed to yield an energy profile that allows conformational transition to occur on a functional timescale (Figure 3). Multiple states must have sufficiently low energies and the energy differences between states be small enough for the conformational transition to occur. Chica and co-workers [15] have recently proposed "*meta*-multistate design" (*meta*-MSD) which has been successfully applied for the design of Streptococcal protein G domain $\beta 1$ (G$\beta 1$) variants capable of spontaneous conformation switching in a millisecond timescale. The procedure relies on the generation of an ensemble of backbone templates that sample the conformational landscape, split into micro-states in order to reduce complexity. These micro-states are generated by optimizing rotamers for predefined sequences on all backbone

templates and are divided into minor, major and transition states according to their structural features. The selected sequences must stably populate the predefined major and minor states with a transition state barrier small enough to allow switching between these two states.

Conclusion and future directions

Outstanding progresses in CPD algorithms and protocols, leading to successful designed proteins have been achieved in the few last decades [53]. Especially, provable CPD algorithms have advanced significantly, becoming competitive with respect to stochastic algorithms to handle complex CPD problems while yielding provable guarantees on the solutions [26]. It has become clear that the incorporation of backbone flexibility in CPD methods is essential to improve the accuracy of their predictions and enable more challenging designs. Design methods including different degrees of protein flexibility tend to predict sequences with lower energy and to recapitulate known sequence profiles at designed positions more accurately [39]. However, algorithmic and methodological advances are still needed for introducing greater conformational variability at both local and global scales while exploring a large sequence space. Such advances would enable the design of proteins to perform more complex tasks involving dynamics, thus paving the way for innovative applications.

A major direction for broadening the range of protein functionalities lies in the ability to design proteins capable of conformational changes. Recent advances have been made, including the design of systems that dynamically exchange between two conformational states [31, 15, 17, 70]. Although multi-state design approaches have opened new avenues to address the design of switchable protein systems, significant advances are needed to generalize the design of a dynamic mode in a protein fold. Aiming to go further in this direction, interesting algorithmic approaches have been proposed to simultaneously explore conformational transitions and sequences for the design of optimized protein motions [41]. However, the practical applicability of these methods remains very limited, and more research efforts are required to tackle real-world protein motion design problems.

An alternative exciting avenue lies in sequence-based generative Machine Learning methods. These can be trained on sets of sequences sharing a common fold or function and used to generate new sequences that should behave similarly. The Direct Coupling Analysis (DCA) used for contact map prediction [42, 55] produces a generative probabilistic model over sequences from a multiple sequence alignment. This probabilistic model can be sampled and the resulting sequences have been experimentally shown to often produce functional proteins, including enzymes [7, 52]. Deep Learning (DL) models with various architectures have also been trained on sets of homologous sequences and used to generate active peptides [22], or enzymes [51] (with success rates similar to those of DCA [52]). The main limitation of these approaches, beyond the need for training data and risks of over-fitting, is that they reproduce existing functions/folds, which is rarely the sole aim of protein design. However, the latent

space learned by an auto-encoder can be used to interpolate or extrapolate sequences [11] and more complex architecture have been designed to slowly force generative models towards sequences of interest [23]. A promising direction for future research is to extend the more sophisticated structure-aware DL-based design approaches [3] to design for an ensemble of backbones.

Funding

This work has been supported by the French ANR through grant ANR-19-PI3A-0004.

References

[1] Allouche, D. *et al.* (2014). Computational protein design as an optimization problem. *Artificial Intelligence*, **212**, 59–79.

[2] Ambroggio, X. I. and Kuhlman, B. (2006). Computational Design of a Single Amino Acid Sequence that Can Switch between Two Distinct Protein Folds. *Journal of the American Chemical Society*, **128**(4), 1154–1161. Publisher: American Chemical Society.

[3] Anand, N. *et al.* (2020). Protein Sequence Design with a Learned Potential. *bioRxiv*, page 2020.01.06.895466. Publisher: Cold Spring Harbor Laboratory Section: New Results.

[4] Ashworth, J. *et al.* (2006). Computational redesign of endonuclease DNA binding and cleavage specificity. *Nature*, **441**(7093), 656–659.

[5] Broom, A. *et al.* (2020). Ensemble-based enzyme design can recapitulate the effects of laboratory directed evolution in silico. *Nature communications*, **11**(1), 1–10.

[6] Charpentier, A. *et al.* (2019). Variable Neighborhood Search with Cost Function Networks To Solve Large Computational Protein Design Problems. *Journal of Chemical Information and Modeling*, **59**(1), 127–136. Publisher: American Chemical Society.

[7] Cheung, N. J. and Yu, W. (2019). Sibe: a computation tool to apply protein sequence statistics to predict folding and design in silico. *BMC bioinformatics*, **20**(1), 1–11.

[8] Clark, J. J. *et al.* (2019). Inherent versus induced protein flexibility: Comparisons within and between apo and holo structures. *PLOS Computational Biology*, **15**(1), e1006705. Publisher: Public Library of Science.

[9] Coutsias, E. A. *et al.* (2004). A kinematic view of loop closure. *Journal of Computational Chemistry*, **25**(4), 510–528. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.10416.

[10] Dahiyat, B. I. and Mayo, S. L. (1997). De Novo Protein Design: Fully Automated Sequence Selection. *Science*, **278**(5335), 82–87. Publisher: American Association for the Advancement of Science Section: Research Article.

[11] Das, P. *et al.* (2018). PepCVAE: Semi-Supervised Targeted Design of Antimicrobial Peptide Sequences. *arXiv:1810.07743 [cs, q-bio, stat]*. arXiv: 1810.07743.

[12] Davey, J. A. and Chica, R. A. (2012). Multistate approaches in computational protein design. *Protein Science : A Publication of the Protein Society*, **21**(9), 1241–1252.

[13] Davey, J. A. and Chica, R. A. (2017). Multistate Computational Protein Design with Backbone Ensembles. *Methods in Molecular Biology (Clifton, N.J.)*, **1529**, 161–179.

[14] Davey, J. A. *et al.* (2015). Prediction of stable globular proteins using negative design with non-native backbone ensembles. *Structure*, **23**(11), 2011–2021.

[15] Davey, J. A. *et al.* (2017). Rational design of proteins that exchange on functional timescales. *Nature chemical biology*, **13**(12), 1280–1285.

[16] Davis, I. W. *et al.* (2006). The backrub motion: how protein backbone shrugs when a sidechain dances. *Structure (London, England: 1993)*, **14**(2), 265–274.

[17] DeGrave, A. J. *et al.* (2018). Large enhancement of response times of a protein conformational switch by computational design. *Nature Communications*, **9**(1), 1013.

[18] Gaillard, T. *et al.* (2016). Protein side chain conformation predictions with an mmgbsa energy function. *Proteins: Structure, Function, and Bioinformatics*, **84**(6), 803–819.

[19] Gainza, P. *et al.* (2012). Protein Design Using Continuous Rotamers. *PLOS Computational Biology*, **8**(1), e1002335. Publisher: Public Library of Science.

[20] Gao, W. *et al.* (2020). Deep learning in protein structural modeling and design. *Patterns*, pages 100–142.

[21] Georgiev, I. *et al.* (2008). Algorithm for backrub motions in protein design. *Bioinformatics*, **24**(13), i196–i204.

[22] Grisoni, F. *et al.* (2018). Designing Anticancer Peptides by Constructive Machine Learning. *ChemMedChem*, **13**(13), 1300–1302. _eprint: https://chemistry-europe.onlinelibrary.wiley.com/doi/pdf/10.1002/cmdc.201800204.

[23] Gupta, A. and Zou, J. (2019). Feedback GAN for DNA optimizes protein functions. *Nature Machine Intelligence*, **1**(2), 105–111. Number: 2 Publisher: Nature Publishing Group.

[24] Hallen, M. A. and Donald, B. R. (2016). comets (Constrained Optimization of Multistate Energies by Tree Search): A Provable and Efficient Protein Design Algorithm to Optimize Binding Affinity and Specificity with Respect to Sequence. *Journal of Computational Biology*, **23**(5), 311–321.

[25] Hallen, M. A. and Donald, B. R. (2017). CATS (Coordinates of Atoms by Taylor Series): protein design with backbone flexibility in all locally feasible directions. *Bioinformatics*, **33**(14), i5–i12.

[26] Hallen, M. A. and Donald, B. R. (2019). Protein design by provable algorithms. *Communications of the ACM*, **62**(10), 76–84.

[27] Hallen, M. A. *et al.* (2013). Dead-end elimination with perturbations (DEEPer): a provable protein design algorithm with continuous sidechain and backbone flexibility. *Proteins*, **81**(1), 18–39.

[28] Hallen, M. A. *et al.* (2017). LUTE (local unpruned tuple expansion): Accurate continuously flexible protein design with general energy functions and rigid rotamer-like efficiency. *Journal of Computational Biology*, **24**(6), 536–546.

[29] Hallen, M. A. *et al.* (2018). Osprey 3.0: Open-source protein redesign for you, with powerful new features. *Journal of computational chemistry*, **39**(30), 2494–2507.

[30] Havranek, J. J. and Harbury, P. B. (2003). Automated design of specificity in molecular recognition. *Nature structural biology*, **10**(1), 45–52.

[31] Joh, N. H. *et al.* (2014). De novo design of a transmembrane Zn2+-transporting four-helix bundle. *Science (New York, N.Y.)*, **346**(6216), 1520–1524.

[32] Jou, J. D. *et al.* (2020). Minimization-Aware Recursive K*: A Novel, Provable Algorithm that Accelerates Ensemble-Based Protein Design and Provably Approximates the Energy Landscape. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology*, **27**(4), 550–564.

[33] Karimi, M. and Shen, Y. (2018). iCFN: an efficient exact algorithm for multistate protein design. *Bioinformatics*, **34**(17), i811–i820.

[34] Khatib, F. *et al.* (2011). Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences of the United States of America*, **108**(47), 18949–18953.

[35] Kuhlman, B. and Baker, D. (2000). Native protein sequences are close to optimal for their structures. *Proceedings of the National Academy of Sciences of the United States of America*, **97**(19), 10383–10388.

[36] Kuhlman, B. and Bradley, P. (2019). Advances in protein structure prediction and design. *Nature Reviews Molecular Cell Biology*, **20**(11), 681–697. Number: 11 Publisher: Nature Publishing Group.

[37] Kuhlman, B. *et al.* (2003). Design of a Novel Globular Protein Fold with Atomic-Level Accuracy. *Science*, **302**(5649), 1364–1368. Publisher: American Association for the Advancement of Science Section: Research Article.

[38] Leach, A. R. and Lemon, A. P. (1998). Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. *Proteins: Structure, Function, and Bioinformatics*, **33**(2), 227–239.

[39] Loshbaugh, A. L. and Kortemme, T. (2020). Comparison of Rosetta flexible-backbone computational protein design methods on binding interactions. *Proteins: Structure, Function, and Bioinformatics*, **88**(1), 206–226. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.25790.

[40] Mandell, D. J. *et al.* (2009). Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. *Nature methods*, **6**(8), 551–552.

[41] Molloy, K. *et al.* (2019). Simultaneous system design and path planning: A sampling-based algorithm. *The International Journal of Robotics Research*, **38**(2-3), 375–387.

[42] Morcos, F. *et al.* (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, **108**(49), E1293–E1301.

[43] Murphy, G. S. *et al.* (2012). Increasing Sequence Diversity with Flexible Backbone Protein Design: The Complete Redesign of a Protein Hydrophobic Core. *Structure(London, England:1993)*, **20**(6), 1086–1096.

[44] Negron, C. and Keating, A. E. (2013). Multistate protein design using clever and classy. In *Methods in enzymology*, volume 523, pages 171–190. Elsevier, Amsterdam, The Netherlands.

[45] Nivón, L. G. *et al.* (2014). Automating human intuition for protein design. *Proteins: Structure, Function, and Bioinformatics*, **82**(5), 858–866. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.24463.

[46] Ojewole, A. A. *et al.* (2018). BBK* (Branch and Bound Over K*): A Provable and Efficient Ensemble-Based Protein Design Algorithm to Optimize Stability and Binding Affinity Over Large Sequence Spaces. *Journal of Computational Biology*, **25**(7), 726–739.

[47] Ollikainen, N. *et al.* (2015). Coupling Protein Side-Chain and Backbone Flexibility Improves the Re-design of Protein-Ligand Specificity. *PLoS computational biology*, **11**(9), e1004335.

[48] Park, J. D. (2002). Map complexity results and approximation methods. In *Uncertainty in Artificial Intelligence (UAI)*, page 388–396, Alberta,Canada. Morgan Kaufmann, Boston, USA.

[49] Pierce, N. A. and Winfree, E. (2002). Protein Design is NP-hard. *Protein Engineering, Design and Selection*, **15**(10), 779–782.

[50] Pokala, N. and Handel, T. M. (2005). Energy functions for protein design: adjustment with protein–protein complex affinities, models for the unfolded state, and negative design of solubility and specificity. *Journal of molecular biology*, **347**(1), 203–227.

[51] Repecka, D. *et al.* (2021). Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, pages 1–10. Publisher: Nature Publishing Group.

[52] Russ, W. P. *et al.* (2020). An evolution-based model for designing chorismate mutase enzymes. *Science*, **369**(6502), 440–445.

[53] Samish, I., editor (2017). *Computational Protein Design*. Methods in Molecular Biology. Humana Press, Ney-York, USA.

[54] Sauer, M. F. *et al.* (2020). Multi-state design of flexible proteins predicts sequences optimal for conformational change. *PLOS Computational Biology*, **16**(2), e1007339. Publisher: Public Library of Science.

[55] Seemayer, S. *et al.* (2014). CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*, **30**(21), 3128–3130.

[56] Silver, N. W. *et al.* (2013). Efficient Computation of Small-Molecule Configurational Binding Entropy and Free Energy Changes by Ensemble Enumeration. *Journal of Chemical Theory and Computation*, **9**(11), 5098–5115. Publisher: American Chemical Society.

[57] Simoncini, D. *et al.* (2015). Guaranteed Discrete Energy Optimization on Large Protein Design Problems. *Journal of Chemical Theory and Computation*, **11**(12), 5980–5989.

[58] Simoncini, D. *et al.* (2019). A structural homology approach for computational protein design with flexible backbone. *Bioinformatics*, **35**(14), 2418–2426.

[59] Simons, K. T. *et al.* (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions11Edited by F. E. Cohen. *Journal of Molecular Biology*, **268**(1), 209–225.

[60] Skjaerven, L. *et al.* (2011). Dynamics, flexibility and ligand-induced conformational changes in biological macromolecules: a computational approach. *Future Medicinal Chemistry*, **3**(16), 2079–2100.

[61] Smith, C. A. and Kortemme, T. (2008). Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. *Journal of Molecular Biology*, **380**(4), 742–756.

[62] Traoré, S. *et al.* (2016). Fast search algorithms for computational protein design. *Journal of computational chemistry*, **37**(12), 1048–1058.

[63] Traoré, S. *et al.* (2013). A new framework for computational protein design through cost function network optimization. *Bioinformatics*, **29**(17), 2129–2136.

[64] Traoré, S. *et al.* (2017). Deterministic Search Methods for Computational Protein Design. In I. Samish, editor, *Computational Protein Design*, Methods in Molecular Biology, pages 107–123. Springer, New York, NY.

[65] Tyka, M. D. *et al.* (2011). Alternate states of proteins revealed by detailed energy landscape mapping. *Journal of molecular biology*, **405**(2), 607–618.

[66] Villa, F. *et al.* (2018). Adaptive landscape flattening in amino acid sequence space for the computational design of protein:peptide binding. *The Journal of Chemical Physics*, **149**(7), 072302. Publisher: American Institute of Physics.

[67] Viricel, C. *et al.* (2018). Cost function network-based design of protein-protein interactions: predicting changes in binding affinity. *Bioinformatics (Oxford, England)*, **34**(15), 2581–2589.

[68] Voigt, C. A. *et al.* (2000). Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *Journal of Molecular Biology*, **299**(3), 789–803.

[69] Vucinic, J. *et al.* (2020). Positive multistate protein design. *Bioinformatics (Oxford, England)*, **36**(1), 122–130.

[70] Wei, K. Y. *et al.* (2020). Computational design of closely related proteins that adopt two well-defined but structurally divergent folds. *Proceedings of the National Academy of Sciences*, **117**(13), 7208–7215. Publisher: National Academy of Sciences Section: Biological Sciences.

[71] Zhao, R. *et al.* (2018). How Does the Flexibility of Molecules Affect the Performance of Molecular Rotors? *The Journal of Physical Chemistry C*, **122**(43), 25067–25074. Publisher: American Chemical Society.