Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Computational protein design

A Structural Homology Approach for Computational Protein Design with Flexible Backbone

David Simoncini[1,4], Kam Y.J. Zhang[2], Thomas Schiex [3] and Sophie Barbe [1]*

[1]LISBP, Université de Toulouse, CNRS, INRA, INSA, Toulouse, France.
[2] Laboratory for Structural Bioinformatics, Center for Biosystems Dynamics Research, RIKEN, Yokohama, Kanagawa, Japan.
[3] MIAT, Université de Toulouse, INRA, Auzeville-Tolosane, France.
[4] IRIT, UMR 5505-CNRS, Université de Toulouse, Toulouse, France.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Structure-based Computational Protein design (CPD) plays a critical role in advancing the field of protein engineering. Using an all-atom energy function, CPD tries to identify amino acid sequences that fold into a target structure and ultimately perform a desired function. Energy functions remain however imperfect and injecting relevant information from known structures in the design process should lead to improved designs.
Results: We introduce Shades, a data-driven CPD method that exploits local structural environments in known protein structures together with energy to guide sequence design, while sampling side-chain and backbone conformations to accommodate mutations. Shades (Structural Homology Algorithm for protein DESign), is based on customized libraries of non-contiguous in-contact amino acid residue motifs. We have tested Shades on a public benchmark of 40 proteins selected from different protein families. When excluding homologous proteins, Shades achieved a protein sequence recovery of 30% and a protein sequence similarity of 46% on average, compared to the PFAM protein family of the target protein. When homologous structures were added, the wild-type sequence recovery rate achieved 93%.
Availability: Shades source code is available at `https://bitbucket.org/satsumaimo/shades` as a patch for Rosetta 3.8 with a curated protein structure database and ITEM library creation software.
Contact: `Sophie.Barbe@insa-toulouse.fr`
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The development of Computational Protein Design (CPD) methods is motivated by the ever-increasing practical needs for improving, modifying, and /or expanding the function of natural proteins. CPD seeks to identify amino acid sequences that will fold into a given target structure with sufficient stability and, ultimately perform a desired function. It enables the *in silico* evaluation of amino acid sequences on a scale which is out of reach of experimental methods. The application of CPD is broad, ranging from medicine, biotechnology, and synthetic biology to nanotechnologies (Khoury *et al.*, 2014).

In the last two decades, CPD has proven to be a valuable tool for protein engineering: it has been successfully applied to optimize protein properties (stability, binding affinity…) (Sammond *et al.*, 2011; Whitehead *et al.*, 2012), introduce new binding specificity toward several types of (macro)molecules (Potapov *et al.*, 2008; Ollikainen *et al.*, 2015; Verges *et al.*, 2015), create new protein folds (Kuhlman *et al.*, 2003; Koga *et al.*, 2012), self-assembling proteins (Stranges *et al.*, 2011; King *et al.*, 2012; Voet *et al.*, 2014; Noguchi *et al.*, 2019) and *de novo* proteins with new functions (Rothlisberger *et al.*, 2008; Jiang *et al.*, 2008; Eiben *et al.*, 2012). Despite these impressive results, a successful design is never guaranteed and several limitations still need to be addressed (Setiawan *et al.*, 2018). A clear challenge for CPD methods is the shear size of the combined protein

sequence and conformation space that seems essentially out of reach of existing computational methods. For this reason, several simplifying assumptions are usually made. Beyond the limitation of the sequence space to amino acid regions of the target protein, the conformational space is usually restricted by assuming that the protein backbone is a rigid body and that the amino acid side-chain adopt conformations extracted from a finite set of statistically preferred conformations (Dunbrack and Cohen, 1997). In this still daunting space, the stability of the protein needs to be efficiently computed. It is usual to rely on a simplified and approximate pairwise decomposable energy function. In this setting, and despite the problem NP-hardness, exact methods with proven optimality have been used to fully redesign proteins of length up to 100 residues (Traoré *et al.*, 2013; Simoncini *et al.*, 2015; Traoré *et al.*, 2016) and to approximate partition functions for protein-protein binding affinity predictions (Viricel *et al.*, 2016, 2018).

However, the fixed backbone approximation ignores the natural flexibility of proteins. This is considered as one of the major causes of design failures. Experiments have demonstrated that protein backbones adjust to sequence mutations, and researchers have been trying to take this into account since the late 90's (Su and Mayo, 1997; Harbury *et al.*, 1998; Desjarlais and Handel, 1999). In practice, the lack of backbone flexibility may lead to the filtering of a significant fraction of the sequence space which is otherwise accessible to properly folded and functional proteins (Humphris and Kortemme, 2008; Murphy *et al.*, 2012; Jackson *et al.*, 2013). This introduces undesirable biases in sequence selection. The introduction of the backrub motion, inspired by conformational changes observed in high-resolution crystal structures contributed to better simulation of protein conformational plasticity and concerted optimization of side-chains orientations and backbone movements (Davis *et al.*, 2006; Smith and Kortemme, 2008; Ollikainen *et al.*, 2015).

Furthermore, and even when stability is the only design target, existing fast empirical energy functions remain approximate. Protein structure prediction (PSP) has been facing the same complex space, defined by the many continuous degrees of freedom of protein backbones. One of most successful approach in this area consists in extracting knowledge from known 3D protein structures by converting local sequence fragments into libraries of structural fragments that can then be suitably assembled into a global structure (Bowie and Eisenberg, 1994; Mackenzie and Grigoryan, 2017). The successes of these approaches in the Critical Assessment of Structure Prediction experiments (Vincent *et al.*, 2005) support the idea that short local structural fragments may encode sufficient physical properties to describe the native 3D structure.

In this paper, following the idea that CPD is the inverse of protein structure prediction, we thread local sequence motifs extracted from a library of structural fragments built from the target structure and selected through their ability to improve the energy of the resulting all-atom structure. Ultimately, our goal is to use structurally relevant information on the sequence space, extracted from the large amount of existing protein structures, to constrain the search for low-energy designs to regions that match the sequence-structure relationship observed in natural proteins. The idea of using fragments (Potapov *et al.*, 2008; Jacobs *et al.*, 2016) or gathering information from known protein structures (Mitra *et al.*, 2013) for the design of proteins or protein-protein interfaces is not new, and in-depth analyses of the feasibility of fragment-based approaches for CPD exist (Verschueren *et al.*, 2011; Mackenzie and Grigoryan, 2017).

In CPD, fragments can exploit the target structure to sample the sequence space. This represents a promising strategy for CPD, since sequences extracted from fragments can report key determinants of the target structure. Most existing approaches however, consider structural fragments defined by contiguous residues (such as what were gathered in the BriX database (Vanhee *et al.*, 2011)) and thus essentially ignore contacts between residues which may be close in the 3D protein structure but not in the primary sequence. Here, we consider possibly non-contiguous sets of amino acid residues that are in direct contact in the 3D protein structure. We call these sets "In-contact residue TErtiary Motifs" or *ITEMs*. We hypothesize that these ITEMs should better encode physical features of local 3D structural environments than just contiguous amino acid residue fragments. Similarly to what has been done in Protein Structure Prediction, we match the in-contact residue tertiary motif that appears at each position of the target backbone (or target ITEM) with similar structural motifs found in protein structures available in the Protein Data Bank (PDB) (Berman *et al.*, 2000). The set of all matches defines a position specific library of non-contiguous sequence fragments. These libraries of candidate ITEMs are then used to guide the CPD sampling process by only exploring naturally-occurring residue 3D neighborhoods: instead of introducing single mutations, we substitute all the amino acid residues of the target ITEM by the amino acids of a chosen candidate ITEM extracted from the library, thus preserving contacts inside the target ITEM. Recently, Kuhlman and co-workers also proposed a method using non-contiguous fragments from existing proteins but with the aim of generating designable structure templates (Jacobs *et al.*, 2016) rather than directly generating sequences, as in this paper.

In our approach, the design of a full sequence for a given protein scaffold relies on iterative cycles involving substitutions of amino acid residues of a target ITEM in the target protein structure by those of a candidate ITEM followed by backbone motions, amino acid side-chain repacking and energy-based evaluation. The search for a low-energy fully-designed protein model is based on an Estimation of Distribution Algorithm (Mühlenbein and Paaß, 1996), a stochastic population based optimization technique that continuously estimates the propensity of each candidate sequence motif in low-energy models during optimization. Our method, Shades (for Structural Homology Algorithm for protein DESign), is fully automatized, from the preparation of a curated protein structure database, the generation of target ITEMs and associated candidate ITEMs libraries, to the design of a novel sequence for an entire protein.

We have tested Shades on a public benchmark of 40 proteins selected from different protein families. Our results show that protein sequences can be effectively reconstructed by assembling non-contiguous residue sequences coming from similar in-contact residue tertiary motifs in unrelated proteins. Shades is able to recover a 30% sequence identity and 46% sequence similarity compared to the PFAM family of a benchmark of 40 proteins of various structural classes. We show that target protein sequences can be nearly completely reconstructed when the known protein structure database contains homologous structures. Finally, a comparison with a flexible backbone design protocol from the Rosetta modeling software shows that Shades achieves higher sequence recovery rates even in the absence of structural homologs in the database, and runs faster by one order of magnitude.

2 Materials and Methods

2.1 General framework

Shades is a new flexible backbone Computational Protein Design method that guides sequence space sampling by the corresponding sequence motifs retrieved from known 3D protein structures based on the three dimensional organization of in-contact amino acid residues. This full protein design method involves four steps (see Figure 1):

1. Preparation of a curated database of known protein structures (extracted from the Protein Data Bank) and associated ITEMs resulting from the analysis of the neighborhood of every amino acid residue of every protein in the database.

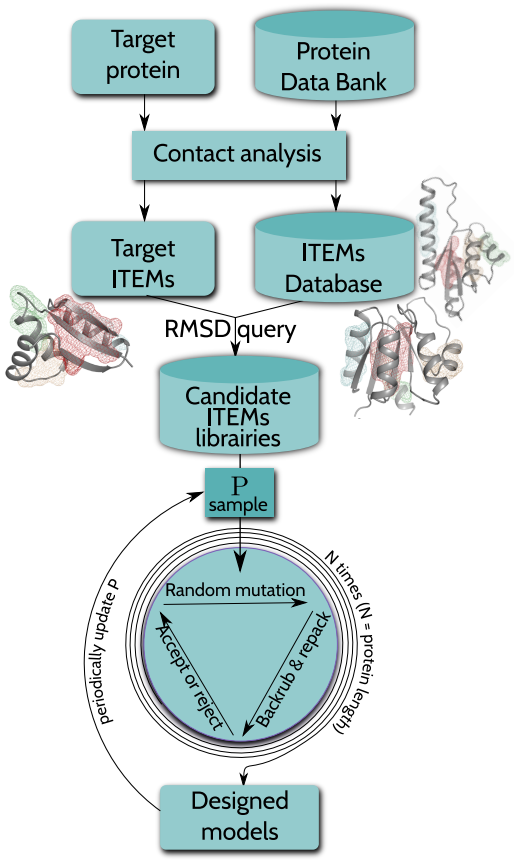


Fig. 1. An overview of Shades: a curated fraction of the PDB is used to prepare a database of ITEMs (In-contact residue TErtiary Motifs), the same process is applied to the target structure to compute target ITEMs. For each target ITEM, a customized candidate ITEMs library is computed. An Estimation of Distribution Algorithm (EDA) algorithm is then used to design sequences and periodically update the candidate ITEMs distributions P_r for every residue r.

2. Analysis of the target 3D protein structure to build the in-contact amino acid residue tertiary motifs (target ITEMs) appearing at each position of the target structure.
3. Extraction of position-specific candidate ITEMs libraries from the curated ITEM database: for a given target ITEM, all candidate ITEMs with a matching 3D topology are extracted.
4. Full sequence assembly by iteratively substituting all the amino acid residues of a target ITEM by the amino acid types of a candidate ITEM, adapting the backbone and repacking all side chains. These cycles are encapsulated in an estimation of distribution algorithm for energy optimization. This protein design method is implemented as a Rosetta protocol and is available as a patch for the release 3.8 of the Rosetta modeling suite.

We now describe this more precisely.

2.2 Residue-residue contacts

Contacts between protein main-chain atoms are defined using Voronoi diagrams, as in CAD-score (Olechnovič *et al.*, 2013). Each atom of the protein backbone is represented as a sphere with its van der Waals radius. In a Voronoi diagram, the Voronoi cell of a sphere contains all the points that are closer to this sphere than to any other sphere. Two spheres are neighbors if their cells share an edge in the Voronoi diagram. In order to restrict ourselves to inter-residue contacts, we ignore contacts between atoms of the same residue. This notion of neighborhood is not sufficient to decide whether two atoms are in-contact or not: two atoms may be neighbors in the Voronoi diagram but distant in Euclidean space. We therefore consider that two neighbor residues in the Voronoi diagram to be "in-contact" only if a water molecule of radius 1.4 Å cannot fit between them. The volume of the overlap between the water molecule and the two residues is used to discriminate contacts and is referred to in the following as the *water overlap volume*. The Voronoi diagram of the protein main-chain atoms and the water overlap volumes between two main-chain positions were computed using the Voroprot software (Olechnovič *et al.*, 2010).

2.3 Definition of ITEMs

For a given residue r appearing in a given protein structure, we define its associated "In-contact residue TErtiary Motif" as the combination of four pieces of information:

1. the protein structure identifier (usually a PDB ID)
2. the type and position of the amino acid residue r in the sequence of the protein.
3. for each residue r' in contact with r, in their order of appearance in the protein sequence, the amino acid type and position of r' in the primary structure and the water overlap volume between r' and r.
4. a triplet giving the number of contacts r' of r which respectively (i) precede r by less than 5 positions (*preceding short range*), (ii) follow r by less than 5 positions (*following short range*) or (iii) are more than 5 positions away (*long range*) in the primary sequence.

The residue r is called the Central Contact Residue (CCR) of the ITEM and the number of contacts of the CCR in each category of relative positions (as defined in 4) is called the *contact signature* of the ITEM.

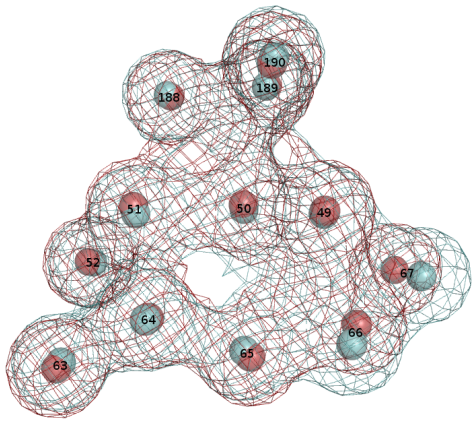
2.4 Protein structure and ITEM database preparation

The database was built by selecting known protein structures from the Protein Data Bank (PDB) (Berman *et al.*, 2000). Several filters were applied (see Table S1), and all structures with missing intra-chain amino acid residues were then removed, yielding a database with $8,965$ protein structures. For each structure of the curated database, Voronoi diagrams and residue-residue contact information were computed. Each amino acid residue of each structure was used as a reference in order to extract the set of all the residues making a contact with it and thus computed the associated ITEM (one ITEM per residue and protein structure in the database).

2.5 Target ITEMs and candidate ITEM library generation

For each amino acid residue of the protein structure to be designed, Voronoi diagrams, residue-residue contact information and associated ITEM were computed, defining a list of target ITEMs. For each target ITEM, the protein structure and ITEM database was then scanned to find candidate ITEMs having similar topology to the target ITEM. The contact signatures of the target ITEM are used as a seed to query the protein structure database in order to build candidate ITEM libraries. A maximum of $1,000$ ITEMs with the same signature and closest water volumes overlaps with the target ITEM was then selected.

Since the contact signatures of the selected ITEMs and the query target ITEM are identical, all the CCRs share the same number of preceding short range, following short range and distant contacts. The selected ITEMs are then superimposed with the query target ITEM and a Root Mean Square Deviation (RMSD) of all main chain atoms is computed using the Ranker tool from the protein clustering software Durandal (Berenger *et al.*, 2012) (see Figure 2). Selected ITEMs are ranked according to their RMSD with the query target ITEM and the top 25 are stored in the candidate ITEM library. Each entry in the candidate ITEM library contains the necessary



11 G49 L51 T52 I63 D64 L65 T66 E67 I188 S189 R190

Fig. 2. A target ITEM superimposed with a candidate ITEM. Only C_α atoms are represented as spheres. Blue spheres are contacts in the target ITEM while red spheres are contacts in the candidate ITEM. The numbers are residue positions in the primary sequence of the target protein. In this example, only two in-contact amino acid residues (at positions 66 and 67) are not perfectly aligned. The C_α RMSD is 0.64 Å. The bottom line shows the corresponding entry in the candidate ITEM library. The central contact residue (50) has 11 contacts: G49, L51, ..., R190.

information for Shades to run: the sequence position and amino acid type of its CCR and all of its contacts.

2.6 Computational protein design method

Our CPD method is implemented as a Rosetta protocol and is available as a patch for the release 3.8 of the Rosetta modeling suite (Leaver-Fay *et al.*, 2011). It relies on an iterative population-based method based on the Estimation of Distribution Algorithm (EDA (Mühlenbein and Paaß, 1996)). EDA algorithms simultaneously optimize a population of solutions and estimate the probability P_r that each candidate ITEM appears at residue r in a good solution. In our context, these distributions are used to stochastically select ITEMs. Initially, these distributions are uniform. EDA algorithms have already been successfully used in the field of protein structure prediction (Simoncini *et al.*, 2012; Simoncini and Zhang, 2013; Simoncini *et al.*, 2017).

Our CPD design method is described in Algorithm 1. At each iteration (line 2), a protein model is randomly initialized (line 3) by substituting each target ITEM of the target protein scaffold by a randomly selected candidate ITEM using P_r, in random order.

An ITEM substitution consists in mutating all amino acid residues of the target ITEM (*i.e.*, the CCR and its contact residues) in the protein model by the amino acid types in the candidate ITEM. Such a substitution defines a basic move in Shades algorithm. There is no independent single residue mutation: when a residue is mutated, it is through the joint adoption of a compatible ITEM identified in another protein. Some amino acid residues positions may appear in several ITEMs, which creates overlaps: part of a previously ITEM substitution can be erased by a new one.

After initialization, we successively mutate a random position r with a randomly selected candidate ITEM using distribution P_r (line 6-7), perform 10 small backrub-based backbone perturbations (line 8) and repack the side-chains (line 9). If the combination of these three operations leads to an energy improvement, the ITEM substitution is accepted (line 11). It is otherwise rejected (line 12). For backbone perturbations, we used the Backrub move method implemented in Rosetta (Smith and Kortemme, 2008), with default parameters (all residues can be used as pivot points, C_α atoms are used as pivot points, the minimum backrub segment size is set at 3 amino acid residues and the maximum at 34 residues (see Rosetta documentation for details). The SideChainPacking module implemented in Rosetta is used for the repacking of amino acid residue side chains (Leaver-Fay *et al.*, 2011). Once the substitution of a target ITEM by a candidate ITEM has been tried as many times as there are residues in the target (line 5), a relaxation of the redesigned protein model is performed with either the FastRelax protocol (Nivon *et al.*, 2013) (with or without backbone restraints) or the energy minimization protocol implemented in Rosetta (line 13) and the model is added to the set of redesigned protein models (line 14). Periodically, the distribution over candidate ITEMs at each position is re-estimated (line 16). This changes the probabilities of selecting each candidate ITEM in the following iterations, taking into account their frequency in previously produced low-energy models. The probability distributions P_r over candidate ITEM are updated by $P_r \leftarrow (1-\alpha)\cdot P_r + \alpha \cdot D_r$ where P_r is the probability over candidate ITEMs for residue r, D_r is the empirical distribution of candidates ITEM in the set of the 10% lowest energy models produced since last estimation and $\alpha \in [0,1]$ is a forgetting rate ($\alpha = 0.2$ in our experiments).

Algorithm 1: Shades: Computational Protein Design algorithm.

input : L {candidate ITEM library}
input : B {Target protein backbone}
input : $nbiterations$ {Number of iterations}
in/out : P {candidate ITEM sampling distributions}
input : $period$ {Reestimation period}
output : D {set of designed protein models}

1 $D \leftarrow \emptyset$;
2 **for** t *in* $[1..nbiterations]$ **do**
3 $d \leftarrow randomized\_init(B, P)$;
4 $d_m \leftarrow d$; (d,d_m: protein models)
5 **for** i *in* $[1..nbresidues]$ **do**
6 $r \leftarrow RandomResidue(B)$;
7 $MutateWithSampledITEM(d_m, r, P_r)$;
8 $BackRub(d_m, 10)$;
9 $SideChainPacking(d_m)$;
10 **if** $score(d_m) < score(d)$ **then**
11 $d \leftarrow d_m$ (Accept)
else
12 $d_m \leftarrow d$ (Reject)
13 $Relaxation(d_m)$; (improve structure)
14 $D \leftarrow D \cup d_m$;
15 **if** $((t \mod period) = 0)$ **then**
16 $ReEstimateDistributions(D, P)$;
17 **return** D;

2.7 Benchmark and set up

We use a published dataset initially proposed for amino acid covariation analysis and subsequently made available to the community for benchmarking new CPD methods (O Conchuir *et al.*, 2015). This dataset contains 40 proteins of various sizes (between 50 and 150 residues) and distinct folds from 40 different protein PFAM families (O Conchuir *et al.*, 2015) (see Table S2). For each protein in the dataset, candidate ITEM libraries were built from our structure database. Structures from the same PFAM protein family and blast hits with an e-value lower than 10 were excluded from the database in order to build homolog-free candidate ITEM libraries. This stringent filter ensures that no candidate ITEM comes from structurally similar PDB scaffold. With this setup, $50,000$ models were

produced for each protein (updating all distributions P_r every $2,000$ produced models). For the comparison with Rosetta Design, a subset of 5 proteins from different PFAM families of the benchmark dataset, was selected (see Table 1). For this comparison, Shades was also run with homologs (*i.e.* ITEMs libraries were built from the entire database, including proteins from the same PFAM family). Using Shades with homologs, $25,000$ models were produced for each of the 5 proteins. It is sufficient because the EDA converges faster since Shades easily identifies ITEMs from homologous structures.

2.8 Comparison to Rosetta Flexible Design method

We compared the performance of Shades with Rosetta FastRelax running in "redesign" mode. In this setting, FastRelax is allowed to mutate the sequence of a protein while relaxing its structure. We compared the performance of both methods on a subset of 5 protein targets of various sizes and folds taken from our benchmark. We measured the CPU time needed for Shades to generate $50,000$ models and allowed the same CPU time for FastRelax to make as many predictions as possible.

FastRelax, being a local search method which explores the mutation space following trajectories, needs a random starting point. Therefore, we generated 100 random starting points by mutating the wild type sequences to random sequences and performing an initial relaxation, for each of the 5 protein targets. Predictions were made using these batches of random starting points.

3 Results and Discussion

3.1 Performance of Shades

Besides using backrub, backbone flexibility can be handled in three different ways in Shades: by unrestrained Rosetta FastRelax protocol, by Rosetta FastRelax protocol with harmonic restraints on the backbone, and by Rosetta minimization method. Figure 3 shows the impact of the flexibility mode on the average Cα RMSD on the whole benchmark between each of the 100 lowest energy designs and the corresponding native structure. Unrestrained FastRelax gives a bit more deviation then minimization. The density curve is also flatter, which means that variance is higher: there is less control on the deviation between the native structure and the designed protein models. The density curve of the FastRelax mode with restraints is centered on 0.2 Å and shows little variance: using this mode, the output models will typically deviate around 0.2 Å from the native structure with good confidence.

Sequence recovery between designed proteins and natural sequences remains the best *in silico* means to evaluate the accuracy of CPD methods (Kuhlman and Baker, 2000; Dai *et al.*, 2010; Gainza *et al.*, 2012), given that two natural proteins generally share the same fold when they have over 30% sequence identity (Rost, 1999). Therefore, for each protein, we measured the sequence recovery between the best solution and the PFAM family of the target protein (see Figure 4). The best solution is defined as the sequence-conformation model with the lowest energy. The values reported in Figure 4 are the native sequence identity and positive values between the best solution and the closest member in the corresponding target PFAM family in unrestrained FastRelax mode. The identity value is the usual sequence recovery measure: the percentage of identical residues when aligning two sequences. The positive value measures similarity and is defined as the percentage of residues that get a positive score from the Blosum62 similarity matrix in their alignment. The average sequence recovery is 30% and the average sequence similarity is 46%. The sequence recovery (resp. similarity) values range from 19% (resp. 32%) to 39% (resp. 62%). Note that Shades achieves these sequence recovery and sequence similarity rates with ITEMs libraries excluding protein structures

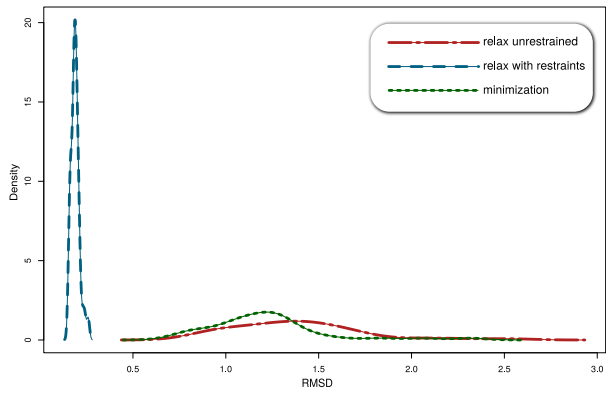


Fig. 3. Density of the C_α RMSD to native structure averaged over the top 100 models of all targets for all flexibility modes. The average RMSD is 1.4Å using unrestrained FastRelax, 1.2Å using minimization and 0.2Å using FastRelax with restraints.

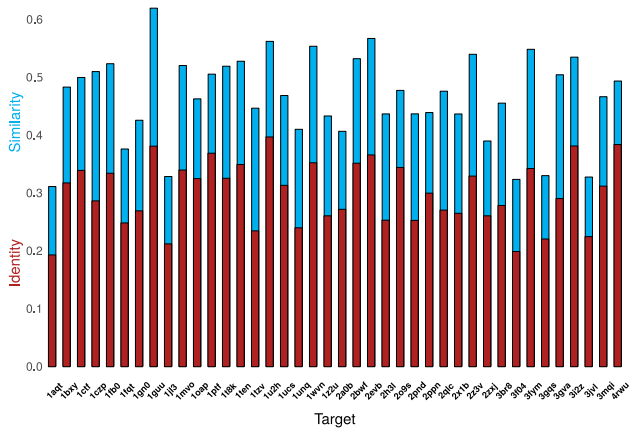


Fig. 4. Sequence identity and similarity values between the best model and the target PFAM family sequences. The sequence identity and similarity toward the closest member of the target PFAM family is reported. The similarity values are computed according to the BLOSUM62 matrix: two amino acid types are considered similar if their score is greater than zero.

similar to the targets: the sequences were designed by assembling ITEMs taken from protein structures completely unrelated to the targets.

Surprisingly, the flexibility mode does not really impact the performance in terms of sequence recovery: the average sequence identity and positive value are comparable for the three different modes. However, the distribution of sequence identity varies depending on the flexibility mode (Figure 5). Using FastRelax with restraints mode, the distribution is unimodal and narrow which means that the probability to obtain a sequence recovery around 30% is higher than for the two other modes. At the other extreme, the distribution of sequence recovery in unrestrained FastRelax mode is bimodal and flatter. Using this mode, it is possible to achieve better sequence recovery rates, but the risk of achieving a poor performance is also higher. In between, using the minimization mode, the distribution is unimodal and slightly flatter than for the restrained FastRelax mode. Using this mode, it is possible to get models with a RMSD deviation to the native structure over 1 Å with reliable performance relatively to the unrestrained FastRelax mode.

We checked if our method was able to identify the best candidate ITEMs and efficiently select them to construct a full sequence. For this purpose, we built a candidate ITEM library from a protein structure database including ITEMs taken from the target protein. We selected

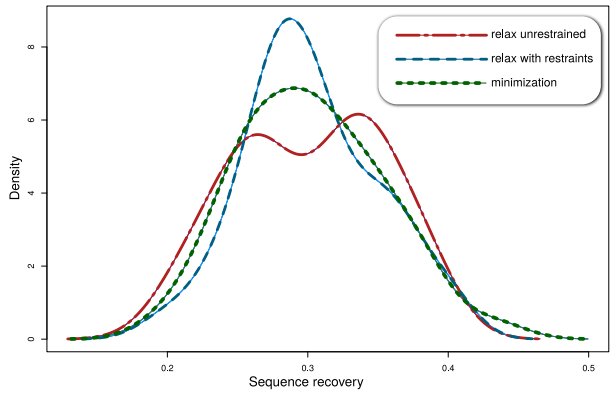


Fig. 5. Distribution of sequence recovery depending on the flexibility mode used.

the protein target with the pdb ID `1jl3` for this test because it was one of the most difficult and longest targets in our benchmark. Our ITEM extraction procedure ranked ITEMs from `1jl3` first for each residue position, proving that these tertiary motifs were correctly identified as the most compatible with the target. Then, we ran Shades for 25 iterations and produced $50,000$ models. The model of lowest energy was produced at iteration 21 and reached a wild-type sequence recovery of 93%. This result shows that after correctly identifying the in-contact residue tertiary motifs, our algorithm has been able to select them to almost entirely reconstruct the native sequence.

3.2 Comparison with Rosetta Flexible Design

We compared the performance of Shades with Rosetta FastRelax running in "redesign" mode, which we refer to in the following as "Rosetta flexible design". When using Shades for real-life protein design applications, there is no reason to exclude structural homologs from the database. Actually, Shades was specifically designed to identify similar structural patterns in known structures and extract ITEMs to inform the search. Furthermore, Shades' search space becomes more constrained when all structural homologs are removed from the database. In contrast, FastRelax has access to the whole sequence space and is only limited by computational efficiency. In this context, it is expected that the method with the larger sequence space achieves lower energy levels, and thus energies are not comparable between Rosetta flexible design and Shades. For this reason, we opted for sequence recovery comparisons. Energy values are given for reference in SI, along with RMSD values of the protein models and the number of models generated by Rosetta flexible design. We compare the CPD results of FastRelax and Shades with and without homologs.

Shades with homologs is the most accurate in terms of sequence recovery out of the 3 tested methods (see Table 1). Shades without homologs is able to slightly better recapitulate the sequences of the target PFAM on average compared to Rosetta flexible design . We measured the average sequence recovery between the model of lowest energy and the whole PFAM members of the target proteins. Shades with homologs achieves a sequence recovery rate (sequence identity) of 0.71 on average. Shades without homologs achieves 0.28 and Rosetta flexible design 0.25. The same trend exists for similarity recovery.

Rosetta flexible design is 25 times slower than Shades: it was able to generate on average about 25 times less models than Shades using the same amount of CPU time (see SI, Table S3). Rosetta flexible design is able to reach much lower energy levels, without being able to get closer to the sequences found in the PFAM of the protein targets. These lower energy levels can be explained by the finer granularity of the basic moves of the sampling method, which are single residue mutations attempts, and by the bigger size of the sequence space which contains the whole amino acid alphabet. On the other side, for Shades, the sequence space is restrained by the amino acid types represented in the ITEMs libraries. Furthermore, the basic moves of the sampling method, which are ITEM insertion attempts, have a coarser granularity. While Shades with homologs achieves the worse energy levels out of the 3 methods, it is able to generate sequences which could be classified in the same PFAM family as the protein targets according to the high sequence recovery rates. The average RMSD between the initial target backbone and the model of lowest energy backbone is also lower on average for Shades with homologs.

3.3 Shades search dynamics

We examined the evolution of the energy of the best models following each iteration. The average over the 40 targets of the energy of the best model is shown on Figure 6. In unrestrained FastRelax mode, it improves from -190 Rosetta Energy Units (REU) initially to -245 REU at iteration 18. The energy dramatically improves from iterations 1 to 13, then stabilizes for 5 iterations and starts to slowly increase from iteration 19. This behavior may indicate that the algorithm converges on average in 20 iterations and in that sense confirms that 25 iterations are sufficient. By convergence, we mean that the search is focused on a particular region of the search space with highly unlikely exploration of other regions. The convergence is thus related to the performance: once the algorithm is focused on a particular region, either this region is the global optimum attraction basin and the performance is optimal or it is a local minimum attraction basin and the performance is limited by the energy of that local minimum. With restrained FastRelax and minimization modes, the energy quickly drops and then achieves a plateau at around iteration 20. The levels of energy vary a lot according to the flexibility mode used. Allowing more flexibility gives access to lower energy levels.

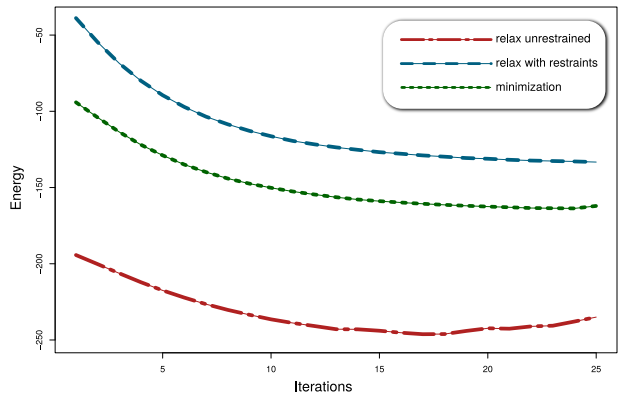


Fig. 6. Energies in Rosetta energy units averaged by iterations over all targets for each flexibility mode.

3.4 Analysis of candidate ITEM libraries

We compared the secondary structure type of each CCR (Central Contact Residue) in the candidate ITEM libraries with the secondary structure type of CCRs of the target protein structure using the Kabsch/Sander notation from DSSP (Kabsch and Sander, 1983). Candidate ITEM library's secondary structure types are generally in agreement with target secondary structure types. Figure 7 shows one typical example of this analysis. The secondary structure type frequency of each CCR in the candidate ITEM library is plotted per residue position as a WebLogo (Crooks *et al.*, 2004). Shades is thus able to capture the target's secondary structures and to

	Rosetta flexible design		Shades with homologs		Shades without homologs	
PDB code	Blast identities	Blast positives	Blast identities	Blast positives	Blast identities	Blast positives
1czp	0.31	0.40	0.85	0.87	0.28	0.51
1fqt	0.23	0.35	0.69	0.70	0.20	0.38
1oap	0.31	0.41	0.73	0.78	0.32	0.44
1wvn	0.19	0.46	0.58	0.65	0.34	0.50
3fym	0.22	0.27	0.70	0.78	0.26	0.45
Average	0.25	0.38	0.71	0.76	0.28	0.46

Table 1. Sequence recovery comparison between Shades and Rosetta flexible design.

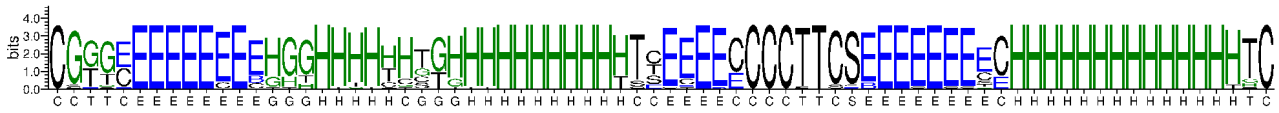


Fig. 7. Secondary structure of library candidate ITEMs and target protein. The secondary structure type of each CCR of candidate ITEMs of a library was examined. The results are shown in WebLogo representation for the 25 motifs per position in the library. The color code is green for helices, blue for beta sheets and black for coils. The letter code is taken from the Kabsch/Sander nomenclature. The design target is PDB ID `1wvn` from PFAM `PF00013` (KH domain). The target secondary structure type is shown on the x-axis.

construct appropriate tertiary motif libraries by looking at local main-chain residue contacts.

We computed the average RMSD between the ITEMs library and the query ITEM for each residue for one protein target (Figure S1). The RMSD values range from 0.11 to 4.6 Å. The regions with higher RMSD values correspond to loops connecting secondary structure elements. The average RMSD over all residues positions is 1.5 Å.

4 Conclusion

Flexible backbone CPD is a challenging problem, mainly because of the high number of degrees of freedom of proteins and the associated sampling issue. Bearing in mind the consistently increasing number of protein structures available in the PDB, we proposed to exploit this mass of data and extract information about the local environment of residues. We thus developed a fully automated flexible backbone CPD method based on tertiary motifs of in-contact amino acid residues. Non-contiguous residue sequences defined by in-contact residue tertiary motifs are assembled together in order to create chimera sequences compatible with the target backbone.

Our results show that it is possible to achieve 30% sequence recovery by assembling in-contact residue tertiary motifs coming from unrelated protein structures while simultaneously allowing for backbone flexibility. In the presence of homologous protein structures in the structure database, our CPD method was able to reconstruct target sequences at 93% recovery rate. These results support our initial hypothesis that tertiary motifs of in-contact residues at least partially capture fundamental sequence-structure relationships. We also showed that Shades outperforms a flexible backbone design application, from the Rosetta modeling software, at rebuilding target sequences.

Shades is built on top of Rosetta and is available as a patch for Rosetta version 3.8. As a Rosetta protocol, Shades accepts all relevant options from the Rosetta modeling suite. It is therefore possible to add coordinate restraints during the relaxation phases and to control the number of backrub steps. Moreover, it is possible to use customized databases as input for specific applications. Whether it is for nanotechnology or biotechnology applications, it is possible to build *ad hoc* databases and to define tailor-made in-contact tertiary motif libraries. Our ITEM-based CPD approach could be extended in future work to inform not only sequence sampling but also protein backbone sampling.

References

Berenger, F. *et al.* (2012). Durandal: fast exact clustering of protein decoys. *Journal of computational chemistry*, **33**(4), 471–474.

Berman, H. M. *et al.* (2000). The protein data bank. *Nucleic Acids Res*, **28**(1), 235–42.

Bowie, J. U. and Eisenberg, D. (1994). An evolutionary approach to folding small alpha-helical proteins that uses sequence information and an empirical guiding fitness function. *Proceedings of the National Academy of Sciences*, **91**(10), 4436–4440.

Crooks, G. E. *et al.* (2004). Weblogo: A sequence logo generator. *Genome Research*, **14**(6), 1188–1190.

Dai, L. *et al.* (2010). Improving computational protein design by using structure-derived sequence profile. *Proteins: Structure, Function, and Bioinformatics*, **78**(10), 2338–2348.

Davis, I. W. *et al.* (2006). The backrub motion: How protein backbone shrugs when a sidechain dances. *Structure*, **14**(2), 265 – 274.

Desjarlais, J. R. and Handel, T. M. (1999). Side-chain and backbone flexibility in protein core design. *Journal of Molecular Biology*, **290**(1), 305 – 318.

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

Eiben, C. B. *et al.* (2012). Increased Diels-Alderase activity through backbone remodeling guided by Foldit players. *Nature biotechnology*, **30**(2), 190–192.

Gainza, P. *et al.* (2012). Protein design using continuous rotamers. *PLOS Computational Biology*, **8**(1), 1–15.

Harbury, P. B. *et al.* (1998). High-resolution protein design with backbone freedom. *Science*, **282**(5393), 1462–1467.

Humphris, E. L. and Kortemme, T. (2008). Prediction of protein-protein interface sequence diversity using flexible backbone computational protein design. *Structure*, **16**(12), 1777 – 1788.

Jackson, E. L. *et al.* (2013). Amino-acid site variability among natural and designed proteins. *PeerJ*, **1**, e211.

Jacobs, T. M. *et al.* (2016). Design of structurally distinct proteins using strategies inspired by evolution. *Science*, **352**(6286), 687–690.

Jiang, L. *et al.* (2008). De Novo Computational Design of Retro-Aldol Enzymes. *Science*, **319**(5868), 1387–1391.

Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical

features. *Biopolymers*, **22**(12), 2577–2637.

Khoury, G. A. *et al.* (2014). Protein folding and *de novo* protein design for biotechnological applications. *Trends in biotechnology*, **32**(2), 99–109.

King, N. P. *et al.* (2012). Computational Design of Self-Assembling Protein Nanomaterials with Atomic Level Accuracy. *Science*, **336**(6085), 1171–1174.

Koga, N. *et al.* (2012). Principles for designing ideal protein structures. *Nature*, **491**(7423), 222–227.

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

Kuhlman, B. *et al.* (2003). Design of a novel globular protein fold with atomic-level accuracy. *Science*, **302**(5649), 1364–1368.

Leaver-Fay, A. *et al.* (2011). Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol*, **487**, 545–74.

Mackenzie, C. O. and Grigoryan, G. (2017). Protein structural motifs in prediction and design. *Current Opinion in Structural Biology*, **44**, 161 – 167. Carbohydrates: A feast of structural glycobiology • Sequences and topology: Computational studies of protein-protein interactions.

Mitra, P. *et al.* (2013). Evodesign: de novo protein design based on structural and evolutionary profiles. *Nucleic Acids Research*, **41**(W1), W273–W280.

Mühlenbein, H. and Paaß, G. (1996). From recombination of genes to the estimation of distributions I. binary parameters. In *Parallel Problem Solving from Nature — PPSN IV*, pages 178–187. Springer-Verlag.

Murphy, G. S. *et al.* (2012). Increasing sequence diversity with flexible backbone protein design: The complete redesign of a protein hydrophobic core. *Structure*.

Nivon, L. G. *et al.* (2013). A pareto-optimal refinement method for protein design scaffolds. *PLOS ONE*, **8**(4), 1–5.

Noguchi, H. *et al.* (2019). Computational design of symmetrical eight-bladed β-propeller proteins. *IUCrJ*, **6**(1).

O Conchuir, S. *et al.* (2015). A web resource for standardized benchmark datasets, metrics, and rosetta protocols for macromolecular modeling and design. *PLOS ONE*, **10**(9), 1–18.

Olechnovič, K. *et al.* (2010). Voroprot: an interactive tool for the analysis and visualization of complex geometric features of protein structure. *Bioinformatics*, **27**(5), 723–724.

Olechnovič, K. *et al.* (2013). Cad-score: A new contact area difference-based function for evaluation of protein structural models. *Proteins: Structure, Function, and Bioinformatics*, **81**(1), 149–162.

Ollikainen, N. *et al.* (2015). Coupling protein side-chain and backbone flexibility improves the re-design of protein-ligand specificity. *PLOS Computational Biology*, **11**, 1–22.

Potapov, V. *et al.* (2008). Computational redesign of a protein-protein interface for high affinity and binding specificity using modular architecture and naturally occurring template fragments. *Journal of Molecular Biology*, **384**(1), 109 – 119.

Rost, B. (1999). Twilight zone of protein sequence alignments. *Protein Engineering, Design and Selection*, **12**(2), 85–94.

Rothlisberger, D. *et al.* (2008). Kemp elimination catalysts by computational enzyme design. *Nature*, **453**(7192), 190–195.

Sammond, D. W. *et al.* (2011). Computational Design of the Sequence and Structure of a Protein-Binding Peptide. *Journal of the American Chemical Society*, **133**, 4190–4192.

Setiawan, D. *et al.* (2018). Recent advances in automated protein design and its future challenges. *Expert Opinion on Drug Discovery*, **13**(7), 587–604. PMID: 29695210.

Simoncini, D. and Zhang, K. Y. J. (2013). Efficient sampling in fragment-based protein structure prediction using an estimation of distribution algorithm. *PLoS ONE*, **8**(7), e68954.

Simoncini, D. *et al.* (2012). A probabilistic fragment-based protein structure prediction algorithm. *PLoS One*, **7**(7), e38799.

Simoncini, D. *et al.* (2015). Guaranteed discrete energy optimization on large protein design problems. *Journal of Chemical Theory and Computation*, **11**(12), 5980–5989.

Simoncini, D. *et al.* (2017). Balancing exploration and exploitation in population-based sampling improves fragment-based de novo protein structure prediction. *Proteins: Structure, Function, and Bioinformatics*.

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.

Stranges, P. B. *et al.* (2011). Computational design of a symmetric homodimer using beta-strand assembly. *Proc Natl Acad Sci U S A*, **108**(51), 20562–7.

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

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

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

Vanhee, P. *et al.* (2011). Brix: a database of protein building blocks for structural analysis, modeling and design. *Nucleic Acids Res*, **39**(Database issue), D435–42.

Verges, A. *et al.* (2015). Computer-aided engineering of a transglycosylase for the glucosylation of an unnatural disaccharide of relevance for bacterial antigen synthesis. *ACS Catalysis*, **5**(2), 1186–1198.

Verschueren, E. *et al.* (2011). Protein design with fragment databases. *Current Opinion in Structural Biology*, **21**(4), 452 – 459.

Vincent, J. J. *et al.* (2005). Assessment of casp6 predictions for new and nearly new fold targets. *Proteins: Structure, Function, and Bioinformatics*, **61**(7), 67–83.

Viricel, C. *et al.* (2016). *Guaranteed Weighted Counting for Affinity Computation: Beyond Determinism and Structure*, pages 733–750. Springer International Publishing, Cham.

Viricel, C. *et al.* (2018). Cost function network-based design of protein–protein interactions: predicting changes in binding affinity. *Bioinformatics*, **1**, 9.

Voet, A. R. D. *et al.* (2014). Computational design of a self-assembling symmetrical beta-propeller protein. *Proceedings of the National Academy of Sciences*, **111**(42), 15102–15107.

Whitehead, T. A. *et al.* (2012). Optimization of affinity, specificity and function of designed influenza inhibitors using deep sequencing. *Nat Biotech*, **30**(6), 543–548.

Acknowledgements

This work was granted access to the HPC resources on the TGCC-Curie supercomputer and the Computing mesocenter of Region Midi-Pyrénées (CALMIP, Toulouse, France).

Funding

This work has been supported by the EMERGENCE program of IDEX Toulouse (E-CODE project), the French National Institute for Agricultural Research (INRA) and the JSPS Kakenhi 18H02395.