# Fitness Landscape Analysis around the Optimum in Computational Protein Design

David Simoncini
LISBP, INRA, CNRS, INSA
Toulouse, France
simoncin@insa-toulouse.fr

Sophie Barbe
LISBP, INRA, CNRS, INSA
Toulouse, France
sbarbe@insa-toulouse.fr

Thomas Schiex
MIAT, Université de Toulouse, INRA UR 875
Castanet-Tolosan, France
thomas.schiex@inra.fr

Sébastien Verel
Université du Littoral Côte d'Opale
Calais, France
verel@uni-littoral.fr

## ABSTRACT

The geometry and properties of the fitness landscapes of Computational Protein Design (CPD) are not well understood, due to the difficulty for sampling methods to access the NP-hard optima and explore their neighborhoods. In this paper, we use a state-of-the-art AI complete algorithm to enumerate all solutions within a 2 kcal/mol energy interval of the optimum of two CPD problems. We compute the number of local minima, the size of the attraction basins, and the local optima network. We provide various features in order to characterize the fitness landscapes, in particular the multimodality, and the ruggedness of the fitness landscape. Results show some key differences in the fitness landscapes and help to understand the successes and failures of metaheuristics on CPD problems. Our analysis gives some previously inaccessible and valuable information on the problem structure related to the optima of the CPD instances (multi-funnel structure), and could lead to the development of more efficient metaheuristic methods.

**ACM Reference Format:**
David Simoncini, Sophie Barbe, Thomas Schiex, and Sébastien Verel. 2018. Fitness Landscape Analysis around the Optimum in Computational Protein Design. In *GECCO '18: Genetic and Evolutionary Computation Conference, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3205455.3205626

## 1 INTRODUCTION

Present in all living organisms, proteins are polymeric chains of amino acids that play a central role in cellular processes such as gene expression, catalysis, communication, regulation, transport and signaling. The succession of amino acids in a protein sequence defines how the protein folds into a given three-dimensional (3D) structure, and thus its biological function. By changing the amino-acid sequence, protein design seeks to produce proteins with new structures and functions. Besides the ever-growing interest to produce tailor-made proteins for applications for the industry and medicine, protein design is also motivated by the desire to understand the relationships between the sequence, evolution, structure and function of biomolecules.

Experimental protein design methods consists in synthesizing various amino-acid sequences using genetic engineering and testing them. However, with a choice among 20 natural amino acids at each position, the size of the combinatorial sequence space ($20^n$ sequences for a protein of length $n$) is out of reach of current experimental methods. Therefore, computational methods have been proposed for rationalizing protein design process by focusing experimental tests on a computationally selected small library of sequences of particular relevance for the targeted structure/function.

To achieve this, Computational Protein Design (CPD) tries to identify, in this exponential amount of amino acid sequences, those that are compatible with a targeted 3D structure, chosen for its functional and/or structural properties. To achieve this, CPD relies on an energy function to assess the stability of the amino-acid sequence in the desired structure and a search algorithm to identify a sequence that can organize itself in space in such a way that it reaches optimal stability. An optimally organized sequence defines a Global Minimum Energy Conformation or GMEC.

Any amino acid can be decomposed in a central invariable part that participates in the formation of the relatively rigid polymeric chain (or backbone) and a variable and highly flexible lateral chemical component that defines the amino acid and that can be chosen among the 20 possible natural variants. For these reasons, and following an approach introduced by Ponder and Richards [17], most CPD approaches assume a rigid backbone structure while amino acid side-chains are allowed to move among a finite set of compositions and preferred organization conformations, called rotamers. While CPD is still a young and rapidly evolving field, success stories of computationally designed proteins highlight its ability to adequately capture fundamental rudiments of molecular recognition and interactions enabling the design of several kinds of proteins for different purposes [8, 28, 29].

The CPD problem can be formulated as an optimization problem which consists in searching for combinations of rotamers at designable positions that will lead to a stable 3D-fold. Despite its apparent simplicity, the rigid backbone/discrete rotamer CPD problem has been proven NP-hard [16] and hard to approximate [3]. Consequently, most CPD approaches rely on metaheuristic search methods based on simulated annealing [9, 30] (implemented into

*GECCO '18, July 15–19, 2018, Kyoto, Japan*

ACM ISBN 978-1-4503-5618-3/18/07…$15.00
https://doi.org/10.1145/3205455.3205626

the Rosetta modeling software [13]), genetic algorithms [18] or other local search algorithms [5, 11]. Although metaheuristics have the advantage of providing a solution at any time, they neither guarantee finding the global optimal solution (or GMEC) in finite time nor a bounded energetic distance to the optimal solution. To try to circumvent this limitation, multiple independent runs are performed (each with a predefined number of steps) in order to cover, as well as possible, a rugged energy landscape. However, the accuracy of metaheuristic methods drastically degrades as problem size increases [22, 30] and the probability of finding the GMEC drops very quickly close to 0 as problems get more difficult. Additionally, the average energy gap to optimality tends to increase with the number of designable positions, putting a limit on the size of systems for which a reasonably good solution can be found with confidence. The lack of knowledge on the features of the energy landscape of CPD problems makes it difficult to understand why metaheuristics methods fail on given instances.

Alternative CPD approaches rely on exact deterministic algorithms guaranteeing that the solution returned at the completion of the algorithm is the GMEC. Unfortunately, these methods, historically based on the Dead-End Elimination theorem combined with the A* algorithm [12] are often rapidly outstripped by the complexity of the search space and do not provide any solution in reasonable time. Recent artificial intelligence methods have allowed to push back these limitations [1, 22, 24, 25]. Based on graphical models and more specifically Cost Function Networks (CFN) [4] methods, they can handle complex search spaces that were previously unsolvable by state-of-the-art provable CPD methods. The CFN-based approaches speed-up search by several orders of magnitude and can usually provide a guaranteed GMEC in reasonable time. In addition to the optimal solution, these methods can provide an exhaustive list of sub-optimal solutions in a given energy threshold around the optimum. This new ability opens new opportunities, notably to exhaustively explore the energy landscape in the neighborhood of global optima. In this paper, we use CFN-based methods to access the guaranteed NP-hard optima of CPD problems and exhaustively explore their neighborhoods. This study provides previously inaccessible and highly valuable data on the CPD energy landscape. Analysis highlights some key characteristics of the two CPD problems explaining the distinct behaviors of a metaheuristic, the simulated annealing algorithm implemented in Rosetta.

## 2 RESULTS

For each of two CPD instances, 2ckx and 2gkt, using the Toulbar2 CFN solver, the optimal solution (GMEC), was identified and an exhaustive enumeration of sub-optimal solutions within an energy interval of 2 $kcal.mol^{-1}$ above the GMEC energy was performed. The number of solutions within this energy interval is 497, 282 for 2gkt and 6, 209, 729 for 2ckx. These two CPD instances were selected because of the difference of performance achieved with simulated annealing. In a previous study, a simulated annealing algorithm was able to reach the GMEC 250 times out of 1000 trajectories on the instance 2gkt and failed to reach the GMEC on 2ckx, even after 1 million trajectories [22]. These results were obtained using the Rosetta software with talaris14 energy function. In order to confirm this difference in performance, we ran 1, 000 trajectories of simulated annealing on both instances using Rosetta software with the recent beta_nov16 energy function (see details in Section 3). Our result confirms that 2gkt has several features indicating that it is easier than 2ckx. For 2gkt, the simulated annealing was able to find the GMEC 24 times, and found 98 distinct sequences in the enumerated landscape (269 trajectories found a solution within 2kcal/mol of the GMEC, several trajectories finding the same sequence). In the case of 2ckx, the simulated annealing never found the GMEC and only 13 sequences (out of the 6, 209, 729 possibilities) appeared in the enumerated landscape. In this section, we present a fitness landscape analysis of the two CPD problems. For a given solution, its fitness is the difference of energy of the optimal solution and this solution (maximum at zero).

### 2.1 Fitness distance correlation

The density of states according to the fitness value decreases exponentially for both instances, with a value of $R^2$ of the regression over 0.98 (Figure 1). The density of states decreases faster for 2ckx, which means fewer solutions near the optimum and suggests that this instance is more difficult.

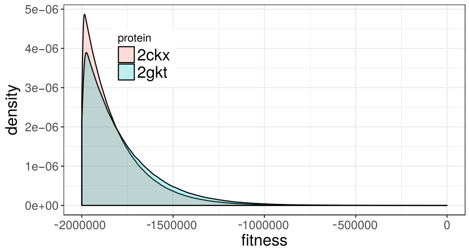

**Figure 1: Density of states over fitness.**

The distributions of the Hamming distances to the optimal sequence have different profiles (Figure 2). The distribution is unimodal for 2gkt, with a mean distance to the optimal sequence of 6. The distribution is bimodal for 2ckx, with one mode at distance 6 and the other one at distance 13 of the optimal sequence. This profile is more problematic for sampling methods because of the risk of being trapped far away from the attraction basin of the global optimum.

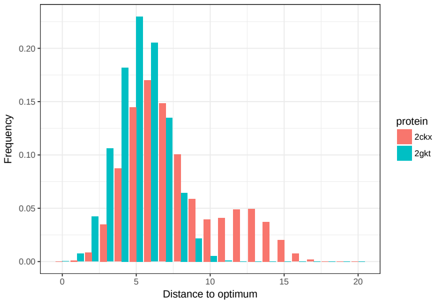

**Figure 2: Distribution of distance to the optimum sequence.**

The fitness of the solutions is better correlated to their distance to the global minimum for 2gkt (Figure 3). The linear correlation of

0.2 for 2gkt puts this instance in a class of easy problems according to the scale of Jones [7]. With a linear correlation of 0.14, 2ckx is a more difficult instance according to this measure. The distance to the optimum decreases with the fitness for both instances, which explains the performance of simulated annealing to find good approximations of the optimal solution on both instances. However, the figure shows a recess at distance 8 of the optimum for 2ckx: this shape suggests that solutions have to break through an energy barrier in order to get closer to the optimal sequence. For 2gkt, the slope is smoother and it seems easier to reach the optimal sequence without having to cross high energy valleys.

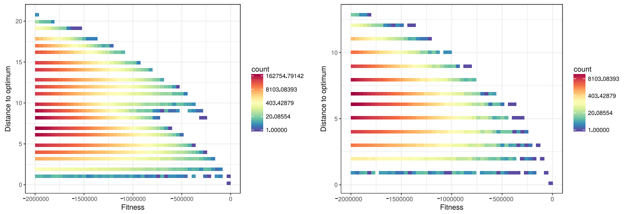

**Figure 3: Correlation between fitness and distance to the optimum sequence (FDC) with color in log scale. 2ckx (left), and 2gkt (right).**

## 2.2 Evolvability

The distribution of the number of improving neighbors (*i.e.*, the number of sequences in the neighborhood with a lower energy) is unimodal for both instances (Figure 4). The distribution is narrower for 2ckx, and has a longer tail for 2gkt, which indicates that a solution of the instance 2gkt is easier to improve.

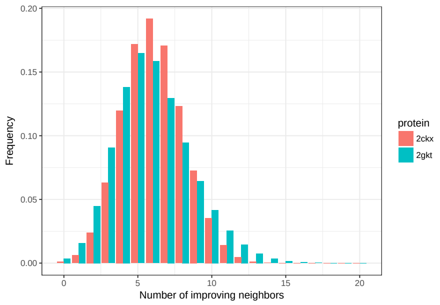

**Figure 4: Distribution of the # of improving neighbors.**

Near-optimal solutions are easier to improve for 2gkt. Figure 5 shows the 2D density between fitness values and the number of improving neighbors. The number of improving neighbors decreases with the fitness for both instances. Although there is no linear relation, the correlation is higher for 2gkt: 0.28, whereas it is 0.23 for 2ckx. Figure 5 shows differences in the bottom left quarter: here the number of improving neighbors decreases faster for 2ckx.

## 2.3 Local optima and basins of attraction

The number of local optima is higher for 2ckx (459) than for 2gkt (151). The number of local optima is generally directly linked to the difficulty of the problems. However, in this study, the number of solutions is different for both instances and a direct comparison is difficult. If we only look at the best 497, 282 solutions for 2ckx, a number that fits the sample size of 2gkt, the number of local minima is 220, which is still higher than for 2gkt.

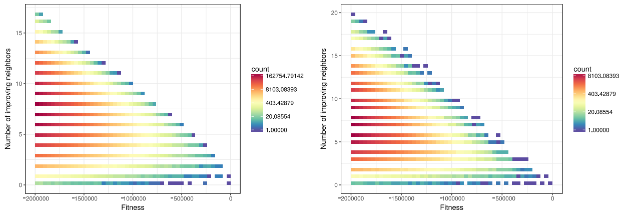

**Figure 5: Correlation between fitness and the number of improving neighbors with color in log scale. 2ckx (left), and 2gkt (right).**

The fitness distribution of local optima has a larger tail for 2gkt (Figure 6), and local optima are more evenly distributed. In contrast, the number of local minima drops down quickly with the quality of the fitness for 2ckx.

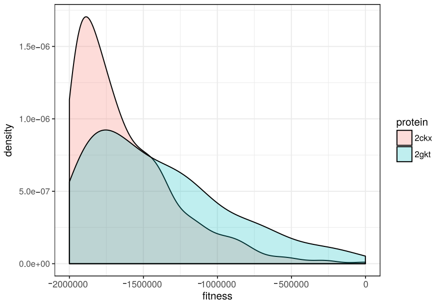

**Figure 6: Fitness distribution of local optima.**

Figure 7 shows the scatter plot of the basin size normalized by the number of solutions (in log scale) as a function of the fitness value of the corresponding local optima. The logarithm of the basins size is linearly correlated to the fitness of the local optima. The correlation is lost for high energy levels, but this may be an effect of the enumeration threshold: some parts of the attraction basins may not belong to the enumerated ensemble of solutions. The attraction basin sizes tend to be smaller for 2ckx away from the global optimum. This trend is inverted for near-optimal basins. For 2ckx, the cumulative basin size of local optima with high fitness values is larger than those of 2gkt: even taking into account the basin sizes, the near-optimal solutions are more difficult to reach for 2ckx.

The length of an adaptive walk (number of steps to reach a local optimum with a steepest-descent algorithm), is positively highly correlated with the logarithm of the corresponding basin size and its fitness value (Figure 8). This property could potentially be used to design a restart strategy. According to the length of steepest-descent, and the fitness value of the local optima found, a restart distance could be estimated in order to maximize the probability to escape from the local optimum, and to find a better local optimum in

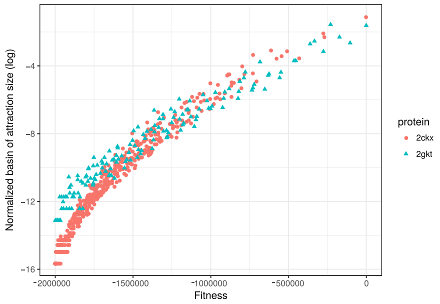

**Figure 7: Scatter plot of the fitness of local optima and their corresponding normalized basin of attraction size. Notice the log scale for the basin size.**

a next steepest-descent. Moreover, notice that the length of adaptive walks of 2ckx are shorter than the length of 2gkt, and indicates a more difficult multi-modal landscape.

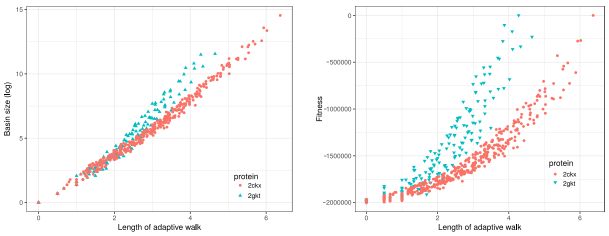

**Figure 8: Scatter plot between the basin size, and fitness of local optima and the average length of steepest descents to reach the local optima. Notice the log scale for the basin size.**

## 2.4 Local optima network

Fig. 9 shows different views of the local optima networks. The graph representing the network of 2gkt is more densely connected than the one of 2ckx. Even though the set of local optima is split in two clusters, the edges show that there is a probability to escape from the basins of attraction of sub-optimal solutions and to reach the cluster leading to the global optimum. For 2ckx, there are several clusters of basins of attraction as well, but no escape edge is visible. Furthermore, the basins of attraction are larger for 2ckx than for 2gkt which confirms that the probability to be trapped into a sub-optimal local optimum basin is higher for 2ckx.

The nodes strength of 2gkt is higher than the nodes strength of 2ckx (Figure 10-left): on average 0.142 for 2gkt, and 0.093 for 2ckx. The strength is the sum of out-going weights, and then the opposite of self-loop weights which is the probability to remain in the same basin of attraction. So, the probability to escape from local optima is 1.5 times higher for 2gkt than for 2ckx: it is easier to escape the basin of attraction of 2gkt, in particular for near-optimal fitness values. This confirms the 2D/3D representation of LON. The local density of the network measured by the weighted clustering coefficients consolidates the picture. 2ckx is locally more clustered ($wCC = 0.103$) than 2gkt ($wCC = 0.034$), and it should be even more difficult to escape from a cluster of local optima for 2ckx. As a consequence of the local structure of LON, the average path length from one random node to the global minimum node is longer for 2ckx (349.0) than for 2gkt (110.0). The structure of the LON indicates that 2ckx is a more difficult problem than 2gkt: the network of 2ckx is more clustered, with few edges toward better nodes, and a higher probability to stay in the basin of attraction.

## 2.5 Ruggedness

2ckx is slightly more rugged than 2gkt. Fig. 11 (top) shows the number of improving neighbors during random walks computed on the sample with the solution with lowest energy. The autocorrelation length of 2gkt (20) is a little bit longer than 2ckx (14). The autocorrelation function of fitness computed from random walks starting from random solution and with an energy bounded to the energy of the initial solution (Figure 12) also confirms this trend. 2gkt seems to be slightly smoother than 2ckx. However, during the random walks, the fitness value suddenly increases or decreases. As a consequence, the sequence of fitness values does not fit an autoregressive model, and another fitness landscape tool should be created to analyze such landscapes. However, this first difference of autocorrelation opens a way to infer the structural difference between the two fitness landscapes from a random solutions, and computationally not expensive sampling.

# 3 METHODS

## 3.1 Cost function networks.

Cost function networks (CFNs) are deterministic Graphical Models derived from Constraint Satisfaction Problems [20]

*Definition 3.1.* A CFN $(X, W, k)$ is defined by:

- a set $X$ of discrete variables $x_i \in X$ indexed by $I = \{1, \ldots, n\}$, each variable $x_i$ takes its values in a finite domain $D_i$ of maximum cardinality $d$.
- a set of cost functions $w_S \in W$ each involving a subset $\{x_i \in X \mid i \in S\}$ of all variables and taking *non negative* values in $[0, k]$.
- The value $k$ is a finite or infinite cost representing an upper bound on costs: a cost of $k$ or above is considered as forbidden.

The set $S \subset I$ of a cost function $w_S$ is called the scope of the cost function. We denote by $D^S$ the Cartesian product of the domains of all variables indexed in $S$: $D^S = \prod_{i \in S} D_i$. Given a tuple $t \in D^S$, and $S' \subset S$ we denote by $t[S']$ the projection of $t$ on $D^{S'}$.

The cost of an assignment $t$ of all variables is defined as the sum $\sum_{w_S \in W} w_S(t[S])$ of all cost functions. If it is strictly less than $k$, the assignment is said to be a solution. The weighted constraint satisfaction problem (WCSP) is to identify a solution of guaranteed minimum cost over all $t \in D^X$. Because of the non negativity of all cost functions, the cost function $w_\varnothing \in W$, a constant cost function with no parameters, defines a lower bound on this minimum cost.

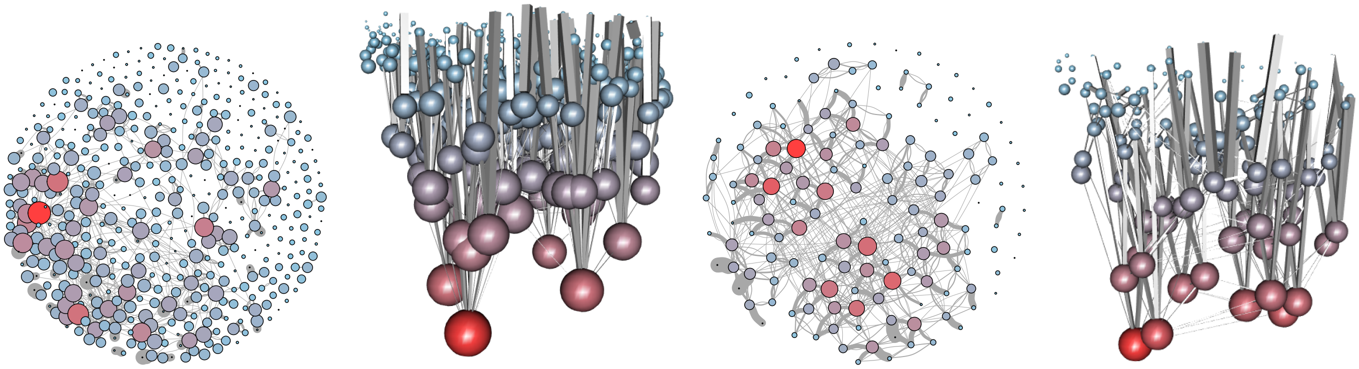

**Figure 9: Local optima network with basin edges for 2ckx (left side) and 2gkt (right side) in 2D and 3D representations. The size is the log of the basin of attraction size. The color is the fitness: the better, the warmer. The thickness of the edges is proportional to the probability of passing from one basin to the other.**

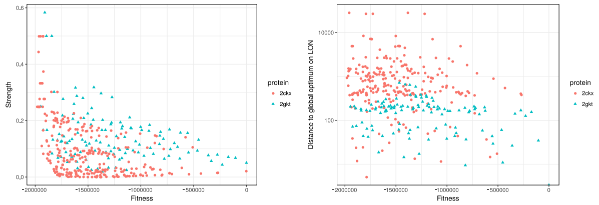

**Figure 10: Strength and distance to global optimum on the network as a function of fitness.**

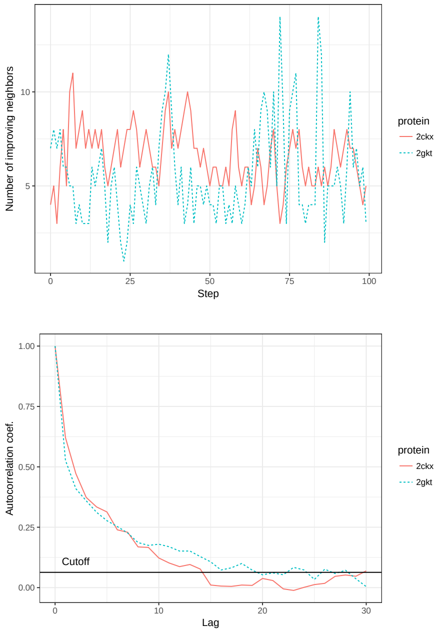

**Figure 11: Example of random walks. At each step the number of improving neighbors and autocorrelation function of the random walks are computed.**

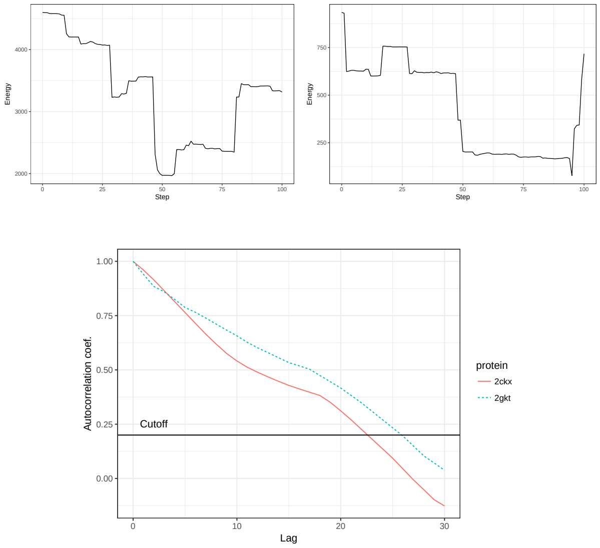

**Figure 12: Example of random walks from a random solution and energy bounded to the energy of the initial point. At each step of the walk, the fitness of the solution is computed.**

## 3.2 CPD modeling

We model CPD with a fixed protein backbone, a set of statistically preferred side chain orientations, or rotamers, and a pairwise decomposable energy function taking the form:

$$E(c) = E_t + \sum_i E(i_r) + \sum_{i<j} E(i_r, j_s)$$

$E_t$ is a constant energy terms that captures the internal energy of the protein backbone and the energy of the interactions between the protein backbone and the environment. $E(i_r)$ captures internal side-chain energies and rotamer-backbone interactions for rotamer $r$ at position $i$. $E(i_r, j_s)$ captures pairwise interactions between rotamers $r$ and $s$ at positions $i$ and $j$ respectively. Because the energy function

<

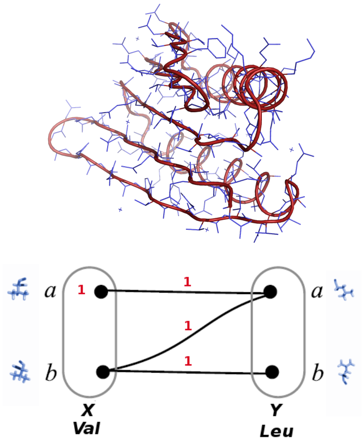

**Figure 13: Top: the backbone of the protein is shown in red. The side chains are shown in blue. In the fixed backbone representation, only side chains can move. Statistically preferred side-chain orientations are stored in rotamer libraries. Bottom: toy CFN example representing the interactions between the side chains of two amino acids: a Valine (variable $X$) and a Leucine (variable $Y$). Values $a$ and $b$ represent rotamers. Assigning $a$ to $X$ has a unary cost of $1$. The assignments $X = a, Y = a$, $X = b, Y = a$, and $X = b, Y = b$ have binary costs of $1$. All other assignments have costs of $0$.**

is pairwise decomposable, all terms can be precomputed and stored in an energy matrix. The objective is to find the combination of rotamers which minimizes the energy of the protein. The side chains differentiate the amino acid types. So, selected rotamers determine the sequence of the protein and solve the CPD problem.

With this representation, modeling CPD as a CFN is straightforward and has be done several times in the literature [1, 22, 24]. Each position $i$ in the protein sequence defines a variable $x_i$ which takes its values in the set of possible rotamers. The precomputed energy terms are converted into costs by shifting them by a constant in order to make them non negative, and by rounding after multiplication by a large constant in order to make them integers. The constant energy term $E_t$ is represented as the constant cost function $w_\varnothing$, rotamer-backbone interactions as unary cost functions and pairwise interactions between rotamers as binary cost functions. The backbone dependent CPD representation and an example of CPD modeling as a CFN are illustrated on Figure 13.

We use ToulBar2[1] to compute the optima and enumerate suboptimal solutions. ToulBar2 explores a search tree with a Branch and Bound algorithm. At the root of the search tree, all variables are unassigned. At each node, the algorithm either assigns a value to a variable (left branch) or removes this value (right branch). Each leaf of the tree is an assignment of all variables. The optimal solution can be obtained by examining the leaves and identifying the assignment that gives the best cost. In order to avoid exploring the whole exponentially sized tree, ToulBar2 maintains an upper bound and a fast incremental lower bound on the cost of solutions to prune branches that cannot lead to better solutions [4]. In a minimization problem, the cost of the best solution found so far defines the upper bound, and the lower bound provides an underestimation of the best cost below an unexplored branch. If the lower bound reaches the upper bound, the branch is pruned. When every branch has been either explored or pruned, the algorithm returns the optimal solution. In order to perform enumerations, we run our algorithm with a fixed upper bound $k = x^* + r$ where $x^*$ is the optimum and $r$ the radius of the enumeration around the optimum. This way, a branch is cut only if it contains solutions whose costs fall outside of the enumeration radius.

## 3.3 Models preparation

The structures of two proteins used as targets were downloaded from the Protein Data Bank (PDB ID : `2ckx` and `2gkt`) and relaxed in all-atom representation with PyRosetta, using the energy function `beta_nov16`. From the relaxed models, pairwise energy matrices were computed and converted into `wcsp` format using PyRosetta, the `beta_nov16` energy function, the Dunbrack backbone dependent rotamer library [21] and an in-house Python script. `2gkt` is 51 amino acid residues long, `2ckx` is 84 amino acid residues long. The global optima were computed from the `wcsp` files using ToulBar2 with the following options: `-dee: -O=-3 -B=1 -A -s --cpd --scpbranch`. The enumerations were performed by running ToulBar2 with the options: `-dee: -A -s -hbfs: --cpd --scpbranch --bestconf -a -ub=threshold`, where `threshold` is the cost of the global optimum augmented by the cost of the desired enumeration radius.

For each instance, 1, 000 simulated annealing trajectories were performed using the `fixbb` program from the Rosetta software with the same energy function and the relaxed protein models as input.

## 3.4 Fitness landscape analysis

In evolutionary computation, fitness landscapes (FL) is one of the fundamental approaches to understand the geometry of the search space from the point of view of local search techniques such as evolutionary algorithm, simulated annealing, etc. On one side, FL depicts the search space with metaphorical pictures such as mountains, peaks, valleys, plateaus, etc. that could help one researcher to design better algorithms; On the other side, it brings a set of numerical metrics that can be used to compare problem difficulty, or be used as features for machine learning techniques that predict algorithm performance. One goal of a fitness landscape analysis is then to contrast optimization difficulty of different problem instances, representations, or neighborhood relations, etc.

Formally, a fitness landscape [23] is a triplet $(\mathcal{X}, \mathcal{N}, f)$ where $\mathcal{X}$ is the set of potential solutions of the optimization problem, $f : \mathcal{X} \to \mathbb{R}$ is the fitness function, and $\mathcal{N} : \mathcal{X} \to 2^{\mathcal{X}}$ is the neighborhood relation between solutions: for each solution $x \in \mathcal{X}$, $\mathcal{N}(x)$ is the set of solutions so-called *neighbors* of $x$. Several features of fitness has been proposed and are detailed in the following paragraphs.

The *Density Of States* (DOS) [19] is the distribution of fitness values of solutions from the search space. It has been shown that the speed of decrease of the distribution tail towards good solutions is an indicator of the problem difficulty. The faster the decrease, the higher the difficulty. Indeed, this distribution corresponds to

[1] http://www.inra.fr/mia/T/toulbar2

the probability density of a random search to reach a fitness value. In our experiments, in order to compare distributions, the fitness value is shifted in such a way that the global minimum of energy defines an optimal zero fitness.

One of the oldest fitness landscape feature is the Fitness Distance Correlation (FDC) [7]. The idea of FDC is to measure how the fitness function can guide search towards the global optimum. The FDC is the scatter plot of the fitness *vs.* the distance of solutions to optimum. The problem becomes easier when the correlation is higher (higher than $0.15$ according to the scale proposed by Jones [7]). The global optimum is required to compute the FDC, which is usually a drawback of the feature. But, for the CPD problem, the global minimum is known. The distance used in this work is the Hamming distance *i.e.*, the number of amino-acid substitutions between two sequences. The distance is clearly related to the local search operator used by existing Simulated Annealing implementations.

The set of FL metrics is related to evolvability [2] which is defined in this context by the ability to evolve towards better solutions. One tool to measure the evolvability is the Fitness Cloud (FC) [26] which is the bi-variate distribution of fitness of solution before and after applying a local search operator *op*: $(f(x), f(op(x)))$. Several statistics can be computed from the FC. For example, the number of improving solutions in the neighborhood: $n^+(x) = \sharp\{x' \in \mathcal{N}(x) \ : \ f(x') < f(x)\}$. A high number means that the probability to improve the current solution is higher. So, the correlation between the fitness of the solution, and the number of improving neighbors is an indicator of the difficulty of the problem: the probability decreases more slowly for an easier problem.

The main geometries of fitness landscape are the neutral ones dominated by large plateaus of solutions with the same fitness values, and the multi-modal ones with many local optima. The ruggedness of fitness landscapes often impacts the multi-modality of the fitness landscapes. Indeed, the ruggedness is the local continuity of the fitness function thanks to the neighborhood relation. More rugged FL tends to be more multi-modal. The autocorrelation of fitness during a random walks [31] is one of the metric of ruggedness. A random walk is a sequence of solutions $(x_1, x_2, x_3, \ldots, x_n)$ such that $x_{t+1}$ is a neighbor of $x_t$. The first solution $x_1$ is a random solution from the search space, and the next solution $x_{t+1}$ of the walk is selected at random in the neighborhood of $x_t$. The autocorrelation function $\rho(s)$ of the fitness is the correlation coefficient of $\{(x_t, x_{t+s}) \ : \ 1 \leq t \leq n - s\}$. The autocorrelation length $\ell$ is defined by the smallest integer such that $|\rho(\ell)| < 2/\sqrt{n}$. It measures the degree of ruggedness of the landscape. A larger autocorrelation length indicates a smoother and easier landscape.

One major feature of multi-modal fitness landscapes is the number of local optima: a high number of local optima usually leads to a difficult problem. However, a number of research works have shown that the link between the number of local optima and difficulty must be contrasted by the size of the basins of attraction [6]. The basin of attraction of a local optimum $x^\star$ is the set of solutions which converge to $x^\star$ by steepest-descent search. A steepest-descent is a walk such that $x_{t+1}$ is the best neighbor of $x_t$ according to the fitness function. Intuitively, if the size of the basin of attraction of the global minimum is "small", then the computation effort to reach it is "important". The size of a basin of attraction can be estimated using the average length of steepest-descent towards the local optimum [27]. It has also been observed on several classes of combinatorial problems (nk-landscapes, Quadratic Assignment Problem, Flow Shop Scheduling Problem) that the fitness of the local optima is correlated with the logarithm of the size of basins of attraction: a deeper local optimum has more chance to have a larger basin of attraction. Notice that the size of global minimum usually remains exponentially small compared to the search size.

Recently, the Local Optima Network (LON) [14] have been proposed to compress the information of the whole search space into a directed weighted graph where the nodes are the local optima, and the edges are the possible weighted transitions between them. Then, the graph can be analyzed using features from complex networks science. Two definitions of edges have been proposed [15]. An edge from the *escape-edges* type between local optima $i$ and $j$ is defined if the distance between $i$ and a solution of the attraction basin of $j$ is below a constant. An edge from the *basin-edges* type between $i$ and $j$ exists if there are two solutions $x_i$ and $x_j$ respectively in the basin of $i$ and $j$ such $x_j$ is a neighbor of $x_i$. The weight is defined by $w_{ij} = \frac{1}{\sharp b_i} \sum_{x \in b_i} \sum_{x' \in b_j} p(x \rightarrow x')$ where $b_i$ and $b_j$ are the basins of attraction of $i$ and $j$, and $p(x \rightarrow x')$ is the probability to pass from the solution $x$ to $x'$ with the given neighborhood structure. The main metrics of the network which are correlated with the performance of local search algorithm are the average strength, the average path length to the global optimum, and the weighted clustering coefficient [15]. The strength of a node is the sum of outgoing weights: $s_i = \sum_{j \neq i} w_{ij}$. A higher strength means a higher probability to escape from the basin of attraction. The distance $d_{ij}$ between two local optima $i$ and $j$ on the network is defined by the inverse of the weight: $d_{ij} = \frac{1}{w_{ij}}$. Intuitively, the weight is the probability to pass from a local optimum to another, so, the distance can be interpreted as the expected number of mutations needed to pass from a local optimum to another. The path length between one local optimum in the network and the global optimum is the length of the shortest path in the graph weighted by the distances $d_{ij}$. The local optima network may have more than one connected component. So, to compute the average distance, the harmonic mean is used. The weighted clustering coefficient (wCC) evaluates the local density of the graph. It measures how two neighbors of a node can be also neighbors with each other. A high clustering coefficient shows a high density network which may indicate either a smooth path to global optimum, or, conversely, point out a hard region of the search from which it is difficult to escape. The wCC remains, however, a useful metric to understand the structure of the network. As complementary information, in addition to numerical metrics, it may be useful to represent the local optima using a 3D representation [10]. The layout is obtained with Fruchterman-Reingold layout algorithm based on the weighted edges. The width of the edge is proportional to the weight, the size of nodes is proportional to the logarithm of the basin size, and the color and height of nodes is given by the fitness value of the local optima.

## 4 CONCLUSION

CPD methods play an important role in protein engineering and have many practical applications in Biotechnology and Synthetic Biology. Metaheuristics are suitable methods for this task, given the size of the search space and the NP-hardness of the problem.

However, without any knowledge of the fitness landscape near the global optimum, and without any guarantee on the quality of the solutions found, it is hard to improve the sampling of the search space and to increase the success rate of CPD-based protein engineering. Using some powerful CFN-based exact optimization methods, we were able to exhaustively enumerate all solutions in the vicinity of the global optimum for a CPD problem on which the simulated annealing algorithm implemented in Rosetta, the most famous and commonly used molecular modeling software, behaves poorly. We analyzed the fitness landscape of this CPD problem, and compared it to the fitness landscape of an easier CPD problem on which the simulated annealing algorithm behaves correctly. Our analysis highlights some crucial differences in the fitness landscapes, explaining the failure of the simulated annealing on the difficult instance. The structure of the landscape, with the presence of several sub-optimal clusters of local optima disconnected from the basin of attraction of the global minimum, is likely the main reason for the bad performance of the simulated annealing. With this type of landscapes, a method able to periodically perturb the solutions in order to help them escape the basin of attraction of a sub-optimal solution would be more suitable. Statistics on random walks also showed some differences between the two instances, in agreement with the fitness landscape analysis, and could be used to assess the difficulty of CPD problems and adapt the sampling strategy before or during sampling.

## ACKNOWLEDGMENTS

This work relied on the HPC resources of the Computing Center of Region Midi-Pyrénées (CALMIP, Toulouse, France). This work has been partially funded by the french Agence Nationale de la Recherche, under Grant ANR-16-C40-0028 and the program EMERGENCE IDEX Toulouse (project E-CODE).

## REFERENCES

[1] David Allouche, Isabelle André, Sophie Barbe, Jessica Davies, Simon de Givry, George Katsirelos, Barry O'Sullivan, Steve Prestwich, Thomas Schiex, and Seydou Traoré. 2014. Computational protein design as an optimization problem. *Artif. Intell.* 212 (2014), 59–79.

[2] Lee Altenberg and others. 1994. The evolution of evolvability in genetic programming. *Advances in genetic programming* 3 (1994), 47–74.

[3] Bernard Chazelle, Carl Kingsford, and Mona Singh. 2004. A semidefinite programming approach to side chain positioning with new rounding strategies. *INFORMS Journal on Computing* 16, 4 (2004), 380–392.

[4] M. Cooper, S. de Givry, M. Sanchez, T. Schiex, M. Zytnicki, and T. Werner. 2010. Soft arc consistency revisited. *Artif. Intell.* 174 (2010), 449–478.

[5] Johan Desmet, Jan Spriet, and Ignace Lasters. 2002. Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins* 48, 1 (July 2002), 31–43. DOI:http://dx.doi.org/10.1002/prot.10131

[6] Josselin Garnier and Leila Kallel. 2002. Efficiency of Local Search with Multiple Local Optima. *SIAM Journal on Discrete Mathematics* 15, 1 (2002), 122–141.

[7] T. Jones. 1995. *Evolutionary Algorithms, Fitness Landscapes and Search.* Ph.D. Dissertation. University of New Mexico, Albuquerque.

[8] Sagar D Khare, Yakov Kipnis, Per Greisen, Ryo Takeuchi, Yacov Ashani, Moshe Goldsmith, Yifan Song, Jasmine L Gallaher, Israel Silman, Haim Leader, Joel L Sussman, Barry L Stoddard, Dan S Tawfik, and David Baker. 2012. Computational redesign of a mononuclear zinc metalloenzyme for organophosphate hydrolysis. *Nature chemical biology* 8, 3 (March 2012), 294–300. DOI:http://dx.doi.org/10.1038/nchembio.777

[9] B Kuhlman and D Baker. 2000. Native protein sequences are close to optimal for their structures. *Proc. Natl. Acad. Sci. U.S.A.* 97, 19 (Sept. 2000), 10383–8. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=27033&tool=pmcentrez&rendertype=abstract

[10] William B Langdon, Nadarajen Veerapen, and Gabriela Ochoa. 2017. Visualising the Search Landscape of the Triangle Program. In *European Conference on Genetic Programming*. Springer, 96–113.

[11] Jonathan Kyle Lassila, Heidi K Privett, Benjamin D Allen, and Stephen L Mayo. 2006. Combinatorial methods for small-molecule placement in computational enzyme design. *Proceedings of the National Academy of Sciences* 103, 45 (2006), 16710–16715.

[12] A R Leach and A P Lemon. 1998. Exploring the conformational space of protein side chains using dead-end elimination and the A$^*$ algorithm. *Proteins* 33, 2 (Nov. 1998), 227–39. http://www.ncbi.nlm.nih.gov/pubmed/9779790

[13] Andrew Leaver-Fay, Michael Tyka, Steven M Lewis, Oliver F Lange, James Thompson, Ron Jacak, Kristian Kaufman, P Douglas Renfrew, Colin A Smith, Will Sheffler, Ian W Davis, Seth Cooper, Adrien Treuille, Daniel J Mandell, Florian Richter, Yih-En Andrew Ban, Sarel J Fleishman, Jacob E Corn, David E Kim, Sergey Lyskov, Monica Berrondo, Stuart Mentzer, Zoran Popović, James J Havranek, John Karanicolas, Rhiju Das, Jens Meiler, Tanja Kortemme, Jeffrey J Gray, Brian Kuhlman, David Baker, and Philip Bradley. 2011. ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol.* 487 (2011), 545–574.

[14] Gabriela Ochoa, Marco Tomassini, Sebástien Vérel, and Christian Darabos. 2008. A study of NK landscapes' basins and local optima networks. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation.* ACM, 555–562.

[15] Gabriela Ochoa, Sébastien Verel, Fabio Daolio, and Marco Tomassini. 2014. Local optima networks: A new model of combinatorial fitness landscapes. In *Recent Advances in the Theory and Application of Fitness Landscapes.* Springer, 233–262.

[16] Niles A Pierce and Erik Winfree. 2002. Protein design is NP-hard. *Protein Eng.* 15, 10 (Oct. 2002), 779–82. http://www.ncbi.nlm.nih.gov/pubmed/12468711

[17] Jay W Ponder and Frederic M Richards. 1987. Tertiary templates for proteins: use of packing criteria in the enumeration of allowed sequences for different structural classes. *Journal of molecular biology* 193, 4 (1987), 775–791.

[18] K Raha, A M Wollacott, M J Italia, and J R Desjarlais. 2000. Prediction of amino acid sequence from structure. *Protein science : a publication of the Protein Society* 9, 6 (June 2000), 1106–19. DOI:http://dx.doi.org/10.1110/ps.9.6.1106

[19] Helge Rosé, Werner Ebeling, and Torsten Asselmeyer. 1996. The density of states—a measure of the difficulty of optimisation problems. In *International Conference on Parallel Problem Solving from Nature.* Springer, 208–217.

[20] T. Schiex, H. Fargier, and G. Verfaillie. 1995. Valued Constraint Satisfaction Problems: hard and easy problems. In *Proc. of the $14^{th}$ IJCAI.* Montréal, Canada, 631–637.

[21] Maxim V Shapovalov and Roland L Dunbrack. 2011. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure* 19, 6 (2011), 844–858.

[22] David Simoncini, David Allouche, Simon de Givry, Céline Delmas, Sophie Barbe, and Thomas Schiex. 2015. Guaranteed Discrete Energy Optimization on Large Protein Design Problems. *Journal of Chemical Theory and Computation* 11, 12 (2015), 5980–5989. DOI:http://dx.doi.org/10.1021/acs.jctc.5b00594

[23] P. F. Stadler. 2002. Fitness Landscapes. In *Biological Evolution and Statistical Physics (Lecture Notes Physics),* M. Lässig and Valleriani (Eds.), Vol. 585. Springer-Verlag, Heidelberg, 187–207.

[24] Seydou Traoré, David Allouche, Isabelle André, Simon de Givry, George Katsirelos, Thomas Schiex, and Sophie Barbe. 2013. A New Framework for Computational Protein Design through Cost Function Network Optimization. *Bioinformatics* 29, 17 (2013), 2129–2136.

[25] Seydou Traoré, Kyle E. Roberts, David Allouche, Bruce R. Donald, Isabelle André, Thomas Schiex, and Sophie Barbe. 2016. Fast search algorithms for computational protein design. *Journal of Computational Chemistry* 37, 12 (2016), 1048–1058. DOI:http://dx.doi.org/10.1002/jcc.24290

[26] Sébastien Verel, Philippe Collard, and Manuel Clergue. 2003. Where are bottlenecks in NK fitness landscapes?. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on,* Vol. 1. IEEE, 273–280.

[27] Sébastien Verel, Arnaud Liefooghe, Laetitia Jourdan, and Clarisse Dhaenens. 2011. Pareto local optima of multiobjective NK-landscapes with correlated objectives. In *European Conference on Evolutionary Computation in Combinatorial Optimization.* Springer, 226–237.

[28] Alizée Verges, Emmanuelle Cambon, Sophie Barbe, Stéphane Salamone, Yann Le Guen, Claire Moulis, Laurence A. Mulard, Magali Remaud-Siméon, and Isabelle André. 2015. Computer-Aided Engineering of a Transglycosylase for the Glucosylation of an Unnatural Disaccharide of Relevance for Bacterial Antigen Synthesis. *ACS Catalysis* 5, 2 (2015), 1186–1198. DOI:http://dx.doi.org/10.1021/cs501288r

[29] Arnout R. D. Voet, Hiroki Noguchi, Christine Addy, David Simoncini, Daiki Terada, Satoru Unzai, Sam-Yong Park, Kam Y. J. Zhang, and Jeremy R. H. Tame. 2014. Computational design of a self-assembling symmetrical beta-propeller protein. *Proceedings of the National Academy of Sciences* 111, 42 (2014), 15102–15107. DOI:http://dx.doi.org/10.1073/pnas.1412768111

[30] C A Voigt, D B Gordon, and S L Mayo. 2000. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *J. Mol. Biol.* 299, 3 (June 2000), 789–803. DOI:http://dx.doi.org/10.1006/jmbi.2000.3758

[31] E. D. Weinberger. 1990. Correlated and uncorrelatated fitness landscapes and how to tell the difference. In *Biological Cybernetics.* 63:325–336.