Scoring function for protein design problem*

Author A.[1]***, Co-author C.***[2]***, Co-author C.***[2]
author@site.ru
[1]Moscow Institute of Physics and Technology; [2]Organization

We address the problem of computational protein design (CPD) - optimization of the protein primary structure given its backbone. The design of proteins with desired shapes and properties is required in such areas as medicine, biotechnology, synthetic biology, nanotechnologies, etc. Unfortunately, CPD has a lot of complexities. As there are many possible amino-acids for each position in the protein chain, the variety of sequences for a specific 3D shape grows exponentially. Moreover, the CPD needs to be formulated as an optimization problem that is valid in a physical sense. Finding its solution is often problematic. In our project we face such difficulties as formulating and solving that optimization problem. We define the objective as a scoring function trained on the inverse problem of protein quality assessment. The final objective is quadratic in the amino-acid types of primary sequence and leads to a formulation of CPD as a problem of constrained binary quadratic programming. As a result, having energy matrix corresponding to the given backbone we need to identify a sequence with a Global Minimum Energy Conformation or GMEC. The experiments demonstrate the efficiency of the proposed method and verify the direct problem solution (given with primary structure determine the skeleton quality using special metrics).

Key words: *computational protein design, cost function networks, global minimum energy conformation, near optimal solutions.*

1 Introduction

The main aim of Computational Protein Design (CPD) is to find amino-acid sequences that have a desired tertiary structure (fold into a given 3D-scaffold). It has become an important tool that allows to alter inherent properties (e.g. stability, binding affinity) of existing proteins or to bestow new functionalities on them, resulting in generation of new therapeutic proteins, enzymatic catalysts, self-assembling protein structures, and protein–protein interfaces [1]. This technology can be applied broadly: from biotechnology, synthetic biology and medicine to nanotechnologies [7].

The greatest issue of CPD is a large size of protein sequence and conformational space to explore since they grow exponentially with number of residues (20 possible amino-acids for each one).

The problem of finding GMEC is often formulated as an optimization problem characterized by:

- pairwise decomposable energy function that corresponds to a protein stability
- discretized description of the amino-acid conformational space based on a library of frequent side-chain conformations (rotamers)
- rigid backbone

Under such assumptions the problem of seeking for a minimum energy conformation is NP-hard. That is why many solutions count on stochastic methods. For example simulated annealing in Rosetta Molecular Modeling suite [6] and Genetic Algorithm used by EGAD [2]. Nevertheless,

*Acknowledgements

absence of finite time convergence guarantees encouraged usage of deterministic algorithms. Nowadays, the most popular one is the Dead End Elimination theorem (DDE) [3] complemented with A^* search implemented in Osprey design suite [4]. This algorithm has been noticeably enhanced after the problem was expressed as a Cost Function Network (CFN) and this novel approach [10] was injected into the well established CPD package. Unfortunately, provable algorithms still need exponential time and space in worst case and thus are not efficient for medium or huge datasets.

Our research consists of three main parts. Firstly, we generate energy matrices for a given tertiary protein structure using scoring function [developed by Mikhail Karasikov, Guillaume Pagès and Sergei Grudinin]. In the second part for a given matrix we predict sequence of amino acids in the protein chain not considering spatial conformations (rotamers) by formulating and solving an optimization problem [9]. This approach decrease the computational complexity of the problem significantly: basic information about the primary structure of a protein can be retrieved using much less resources, consequently allowing to find the sequence of amino acids for more complicated instances. The corresponding optimization problem (inverse protein folding problem) is reduced to the quadratical programming problem. Energy functions, which describe only interactions between all possible pairs of amino acids, but not rotamers [8], are used as objective ones. Final step is the assessment of the predictions quality using the BLOSUM62 [5] substitutional matrix.

2 Problem Statement

Let the protein chain consist of N amino acids. The set $\mathcal{C} = \{1, 2, \ldots, 20\}$ contains indexes that encode all 20 possible amino acids. Let $\boldsymbol{a} = (a_1, \ldots, a_N)$ be the sequence of residues, where $a_i \in \mathcal{C}$. Let's denote tertiary structure (backbone) of a protein with coordinates of each molecule as $\boldsymbol{b}$, sequence of residues as $\boldsymbol{a}$ and energy function of molecular interactions corresponding to them as $E(\boldsymbol{a}, \boldsymbol{b})$. The aim of the inverse computational protein design is to find for the given $\boldsymbol{b_0}$ an optimal sequence of amino-acids $\boldsymbol{a^*}$ - the vector of residues corresponding to the Global Minimum Energy Conformation for the energy function:

For each optimal sequence of amino acids a^* there is a backbone $\boldsymbol{b_0}$:

$$\boldsymbol{b}_0 = \arg\min_{\boldsymbol{b}} E(\boldsymbol{a^*}, \boldsymbol{b}). \tag{1}$$

Generally, we can formulate inverse CPD problem as an optimization one:

$$\|\boldsymbol{b}_0 - \arg\min_{\boldsymbol{b}} E(\boldsymbol{a}, \boldsymbol{b})\| \to \min_{a}. \tag{2}$$

We would like to to reduce such inverse CPD problem to a quadratic optimization problem using the method described in [9]. For this reason we need to slightly change our optimization problem. Let functions $E_{kl} : \mathcal{C}^2 \to \mathbb{R}$ define symmetrical pairwise interaction energies between residues k and l. If on position Then the problem of energy minimization is represented as

$$\varphi_{\boldsymbol{b}}(a) := E(\boldsymbol{a}, \boldsymbol{b}) = \sum_{j=1}^{N} \sum_{i=1}^{N} E_{ij}(a_i, a_j) \to \min_{a_1, a_2, \ldots, a_N \in \mathcal{C}}. \tag{3}$$

The problem can also be written in the following way:

$$\begin{aligned} \underset{\boldsymbol{x}=[\boldsymbol{x}_1^\mathsf{T}, \ldots, \boldsymbol{x}_N^\mathsf{T}]^\mathsf{T}}{\text{minimize}} \quad & \boldsymbol{x}^\mathsf{T} \mathbf{Q} \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{x}_k \in \{0, 1\}^{20}, \qquad k = 1, \ldots, N, \\ & \|\boldsymbol{x}_k\|_0 = 1, \qquad k = 1, \ldots, N, \end{aligned} \tag{4}$$

where

$$\mathbf{Q} = \begin{bmatrix} [E_{11}] & [E_{12}] & \cdots & [E_{1N}] \\ [E_{21}] & [E_{22}] & \cdots & [E_{2N}] \\ \vdots & \vdots & \ddots & \vdots \\ [E_{N1}] & [E_{N2}] & \cdots & [E_{NN}] \end{bmatrix},$$

$$E_{ij} = \begin{bmatrix} E_{ij}(c_1, c_1) & E_{ij}(c_1, c_2) & \cdots & E_{ij}(c_1, c_{20}) \\ E_{ij}(c_2, c_1) & E_{ij}(c_2, c_2) & \cdots & E_{ij}(c_2, c_{20}) \\ \vdots & \vdots & \ddots & \vdots \\ E_{ij}(c_{20}, c_1) & E_{ij}(c_{20}, c_2) & \cdots & E_{ij}(c_{20}, c_{20}) \end{bmatrix},$$

$$\mathbf{Q} \in \mathbb{R}^{20N \times 20N}, \quad E_{ij} \in \mathbb{R}^{20 \times 20}.$$

Finally, it is practical to re-write the problem as

$$\begin{aligned} \underset{\boldsymbol{x} \in \{0,1\}^{20N}}{\text{minimize}} \quad & \boldsymbol{x}^{\mathsf{T}} \mathbf{Q} \boldsymbol{x} \\ \text{subject to} \quad & \mathbf{A}\boldsymbol{x} = \mathbf{1}_N, \end{aligned} \tag{5}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 \cdots 1 & 0 \cdots 0 & \cdots\cdots & 0 \cdots 0 \\ 0 \cdots 0 & 1 \cdots 1 & \cdots\cdots & 0 \cdots 0 \\ \vdots & \vdots & \ddots & \vdots \\ \underbrace{0 \cdots 0}_{20} & \underbrace{0 \cdots 0}_{20} & \cdots\cdots & \underbrace{1 \cdots 1}_{20} \end{bmatrix}, \quad \mathbf{A} \in \{0,1\}^{N \times 20N}.$$

Therefore, we need to to construct a matrix Q of pairwise energy matrices $[E_{ij}]$ by finding their elements.

To achieve that we will use the earlier mentioned SBROD (Smooth-Backbone-Reliant Orientation-Dependent) scoring function [developed by Mikhail Karasikov, Guillaume Pagès and Sergei Grudinin]. It is deduced from a training set of protein models. Its output is "energy"(or just a score) for a given sequence of aminoacids $\boldsymbol{a}$ and a backbone $\boldsymbol{b}$, which does not depend on rotamers.

The SBROD scoring function is composed of four terms related to different structural features (residue-residue orientations, contacts between backbone atoms, hydrogen bonding and solvent-solvate interactions). It can be considered as a linear transformation of a feature vector $\boldsymbol{f}(\boldsymbol{a}, \boldsymbol{b}_0)$ as following:

$$\varphi_{\boldsymbol{b}_0}(\boldsymbol{a}) = K \cdot (\boldsymbol{w}^T \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{b}_0) + C) \tag{6}$$

Where K, $\boldsymbol{w}$ and C are parameters learned by the scoring function and the feature vector is composed of four subvectors:

$$\boldsymbol{f}(\boldsymbol{a}, \boldsymbol{b}_0) = [\boldsymbol{f}^{(1)}(\boldsymbol{a}, \boldsymbol{b}_0), \boldsymbol{f}^{(2)}(\boldsymbol{a}, \boldsymbol{b}_0), \boldsymbol{f}^{(3)}(\boldsymbol{a}, \boldsymbol{b}_0), \boldsymbol{f}^{(4)}(\boldsymbol{a}, \boldsymbol{b}_0)]$$

Each feature subvector $\boldsymbol{f}^{(k)}(\boldsymbol{a}, \boldsymbol{b}_0)$ is also a vector that is composed of pairwise components $\boldsymbol{f}_{ij}^{(k)}(a_i, a_j, \boldsymbol{b}_0)$, namely it is the sum of them for every possible pair of residues i and j:

$$\boldsymbol{f}^{(k)}(a, \boldsymbol{b}_0) = \sum_i \sum_j \boldsymbol{f}_{ij}^{(k)}(a_i, a_j, \boldsymbol{b}_0) \in \mathbb{R}^{n_k} \tag{7}$$

Due to such linear representation, by using (6) we can easily extract $E_{ij}(a_i, a_j)$ from (3) as following:

$$E_{ij}(a_i, a_j) = K \cdot (\boldsymbol{w}^T \boldsymbol{f}_{ij}(a_i, a_j, \boldsymbol{b}_0) + C) \tag{8}$$

Where

$$\boldsymbol{f_{ij}}(a_i, a_j, \boldsymbol{b}_0) = [\boldsymbol{f_{ij}}^{(1)}(a_i, a_j, \boldsymbol{b}_0), \boldsymbol{f_{ij}}^{(2)}(a_i, a_j, \boldsymbol{b}_0), \boldsymbol{f_{ij}}^{(3)}(a_i, a_j, \boldsymbol{b}_0), \boldsymbol{f_{ij}}^{(4)}(a_i, a_j, \boldsymbol{b}_0)]$$

where $\boldsymbol{f_{ij}}^{(k)}(a_i, a_j, \boldsymbol{b}_0) \in \mathbb{R}^{n_k}$ and are found from (7).

3 Basic Algorithm

The are two main stages in SBROD. Firstly, features from each protein model in the dataset are extracted. Secondly, based on that features the scoring function assigns a score to each processed protein. After extracting and preprocessing that features a Ridge Regression model is trained to predict the GDT-TS of protein models. Having score function trained we can asses a given backbone. However, we will need not the score itself but rather its components and parameters, given in (6) and (7). After that we construct matrix Q. We do that by calculating $E_{lk}(a_i, a_j)$ from from (8) for each pair of aminoacids a_i and a_j. Finding 20×20 of such numbers for each pair of the residue positions l and k for the given backbone we create matrices $[E_{lk}]$. And these matrices are elements of the Q matrix. Then we are moving to the next step - solution of the optimization problem. We will apply to it relaxation methods from [9] to proceed to a quadratic optimization problem. Then we solve it by the more appropriate method.

4 Basic experiment

Real CASP datasets for training the SBROD were used by [Mikhail Karasikov, Guillaume Pagès and Sergei Grudinin]. We can use trained scoring function to assess samples taken from (CASP11, CASP12, and MOULDER) just as the authors did. Then we will construct Energy matrices as described in the previous section.

5 Error analysis

We are planning to evaluate Predictions quality using the BLOSUM62 [5] substitutional matrix. Let $\mathbf{B}$ represents the BLOSUM62 matrix. Then, the score $\mathbf{B}(y_i, \hat{y}_j)$ is the score for the substitution of residue y_i for $\hat{y}_j$ at j-position in the chain. If the score is positive, the residues are interchangeable (or equal). Otherwise, this substitution is less likely to occur, and the prediction is wrong. Let $\boldsymbol{y}_{\text{nat}}, \boldsymbol{y}_{\text{pred}} \in \mathcal{C}^N$ be native and predicted sequences of length N, respectively. Then, the quality function can be defined as

$$S(\boldsymbol{y}_{\text{nat}}, \boldsymbol{y}_{\text{pred}}) = \frac{\sum_{k=1}^{N} \mathbf{B}((\boldsymbol{y}_{\text{nat}})_k, (\boldsymbol{y}_{\text{pred}})_k)}{\sum_{k=1}^{N} \mathbf{B}((\boldsymbol{y}_{\text{nat}})_k, (\boldsymbol{y}_{\text{nat}})_k)} \tag{9}$$

A good quality prediction corresponds to $S > 0$, and the best predictions have values of S close to 1. Negative values of S mean poor quality predictions.

Here we will place quality of prediction curves depending on various parameters of the model and methods of optimization used.

6 Model structure analysis

Based on the results in the previous section we will choose optimal parameters and optimization techniques for the model. We will consider parameters as optimal if the quality of model prediction with them will be higher than a particular level.

7 Model choice

Samples from several sources will be chosen for analysis. Models will vary by the number of subvectors of feature vector used for energy matrix construction. We can also use another quality criteria. Also we need to decide how to implement cross-validation. Using results retrieved earlier and from this analysis we need to choose the most preferable model.

8 Conclusion

Effectively trained by [Mikhail Karasikov, Guillaume Pagès and Sergei Grudinin] SBROD provides us with a pretty precise tool of constructing energy matrices on the first step of our inverse CPD problem. Probably using all for terms it consists of will be the best choice. In future we are planning to finish experiments with the SBROD probably retrain the scoring function on some new datasets to expand its generalization abilities and move on to the optimization problem solving stage.

References

[1] C. Y. Chen, I. Georgiev, A. C. Anderson, and B. R. Donald. Computational structure-based redesign of enzyme activity. *Proc. Natl. Acad. Sci. U.S.A.*, 106(10):3764–3769, Mar 2009.

[2] Arnab B. Chowdry, Kimberly A. Reynolds, Melinda S. Hanes, Mark Voorhies, Navin Pokala, and Tracy M. Handel. An object-oriented library for computational protein design. *Journal of Computational Chemistry*, 28(14):2378–2388, 2007.

[3] J. Desmet, M. De Maeyer, B. Hazes, and I. Lasters. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356(6369):539–542, Apr 1992.

[4] P. Gainza, K. E. Roberts, I. Georgiev, R. H. Lilien, D. A. Keedy, C. Y. Chen, F. Reza, A. C. Anderson, D. C. Richardson, J. S. Richardson, and B. R. Donald. OSPREY: protein design with ensembles, flexibility, and provable algorithms. *Meth. Enzymol.*, 523:87–107, 2013.

[5] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. In *Proc. Natl. Acad. Sci., USA*, pages 10915–10919, November 1992. Published as Proc. Natl. Acad. Sci., USA, volume 89, number 22.

[6] 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. Rosetta3: An object-oriented software suite for the simulation and design of macromolecules. *Methods in Enzymology*, 487(C):545–574, 1 2011.

[7] J. M. Palomo and M. Filice. New emerging bio-catalysts design in biotransformations. *Biotechnol. Adv.*, 33(5):605–613, 2015.

[8] R. Rajgaria, S. R. McAllister, and C. A. Floudas. A novel high resolution Calpha–Calpha distance dependent force field based on a high quality decoy set. *Proteins*, 65(3):726–741, Nov 2006.

[9] Andrii Riazanov, Mikhail Karasikov, and Sergei Grudinin. Inverse protein folding problem via quadratic programming. pages 561–568, September 25 2016.

[10] S. Traore, K. E. Roberts, D. Allouche, B. R. Donald, I. Andre, T. Schiex, and S. Barbe. Fast search algorithms for computational protein design. *J Comput Chem*, 37(12):1048–1058, May 2016.