Bivariate Estimation of Distribution Algorithms for Protein Structure Prediction

Daniel Bonetti[1], Alexandre Delbem[1], Jochen Einbeck[2]

[1] Universidade de São Paulo, São Carlos, SP, Brazil
[2] Department of Mathematical Sciences, Durham University, UK

E-mail for correspondence: daniel.bonetti@gmail.com

Abstract: A real-valued bivariate 'Estimation of Distribution Algorithm' specific for the ab initio and full-atom Protein Structure Prediction problem is proposed. It is known that this is a multidimensional and multimodal problem. In order to deal with the multimodality and the correlation of dihedral angles ϕ and ψ, we developed approaches based on Kernel Density Estimation and Finite Gaussian Mixtures. Simulation results have shown that both techniques are promising when applied to that problem.

Keywords: Estimation of Distribution Algorithm; Protein Structure Prediction; Finite Gaussian Mixture; Kernel Density Estimation.

1 Background

Protein Structure Prediction (PSP) is a key problem in Biology. It tries to find protein configurations in order to help in the development of new medicines. Computer methods have received attention in order to bypass the high costs and time needed by experimental methods [Bujnicki, 2009]. Despite that computer simulations do not have the same capability to predict proteins as the experimental methods have, new methods have been introduced over the last two decades.

In this paper we present a novel computer method to predict protein configurations. We use an Estimation of Distribution Algorithm (EDA) [Larranaga and Lozano, 2002] to guide our search process in order to find good protein configurations. EDAs are from the family of Evolutionary Algorithms and they are general optimization techniques.

Most Evolutionary Algorithms use two or three solutions to compose new ones. In contrast, EDAs have the capability to extract significant statistical information from a set of promising solutions in order to create the offspring. This is an important step in the optimization process, since it guides the search process properly toward good solutions.

There are several types of EDAs. The simplest one uses the mean and variance of variables to compose the offspring. However, as we know, the PSP

problem is a multivariate and multimodal problem. Thus, simply using mean and variance will not describe our set of solutions properly. We designed a new EDA specific for PSP with ab initio and full-atom modelling. Basically, we fed our EDA with three statistical methods. The first method, which serves as a reference algorithm, treats the variable as independent ones, that is, the univariate. Further, we modeled the fixed correlation between the dihedral angles ϕ and ψ with two-dimensional Kernel Density Estimation (KDE) [Venables and Ripley, 2002] and Finite Gaussian Mixtures (FGM) [McLachlan and Peel, 2004].
We evaluated our approach with a 25 residues protein. The results showed that, indeed, EDAs with appropriate statistical methods are able to find good solutions for small protein configurations.

2 Estimation of Distribution Algorithms for Protein Structure Prediction

EDAs are a relatively new class of optimization algorithms. They explore the search space by building and sampling probabilistic models from promising solutions. The whole set of solutions is called population. From a randomly initialized population $\boldsymbol{p}$ all solutions (also called individuals) are evaluated using a fitness function. It is a quality measure function that describes how good a solution is. Next, solutions are chosen to be part of the set of selected individuals s. This new set of individuals usually has promising solutions, in which the probabilistic model will be created. It is also important to have a diversified set of selected solutions in order to avoid premature convergence. Next, a probabilistic model of the selected individuals is built. As we discussed in the previous section, we developed three methods. From this model, we generated a new set of individuals, called offspring o. Finally, we can mix the population and the offspring ordered by the fitness value and truncate it with the size of population and overwrite it. All these steps are called a generation (or iteration) and they may continue until a convergence criterion is reached as, for example, a predefined variation of the fitness of the population.
The fitness function of our algorithm has eight different energy types. However, in this paper, only van der Waals energy is used since it describes well the interaction among atoms and makes the experimental analysis easier to understand.
The population is denoted as $\boldsymbol{p} = (p_1^1, \ldots, p_n^m)$, in which n is the population size and m is the length of the vector (an individual). A real-valued vector holds all the dihedral angles of a protein configuration, ranging in $(-180.0, 180.0)$. Each residue in a protein has its own number of dihedral angles. For example, the smallest residue Glycine has only two dihedral angles, ϕ and ψ, while Arginine has six, $\phi, \psi, \chi_1, \chi_2, \chi_3$ and χ_4. Thus, the number of dihedral angles of a protein configuration depends on the primary sequence of the amino acids. In this paper, all experiments were

performed using a 25 residues protein called 1A11. This yielded in a vector with $m = 95$ positions.

3 Univariate EDA for PSP

The Univariate (UNI) version is a simple and fast algorithm. First, an individual randomly chosen from s is used to generate an o_1^1 value. Next, we cause a perturbation to that value by adding a Gaussian random number with mean zero and standard deviation σ. Then, another individual from s is selected to generate the value of o_1^2 and the perturbation is added. That process is repeated until the vector o_1 is filled. At this point, we are ready to compute the fitness of individual o_1. That entire process is repeated until all individuals of o have been filled, that is, until o_n has been created.

4 Bivariate EDA for PSP

Considering that all variables can interact with each other we could consider the entire individual as a m–dimensional problem. However, as we showed in previous section, even a small protein with 25 residues would produce a 95–dimensional problem. That is not computationally efficient and may not render good results. For this reason, we decided to split the problem per residue. Thus, for a 25 residues protein configuration we will have 25 two–dimensional algorithms to perform instead of one of 95–dimensional.
We considered that dihedral angles ϕ and ψ within the same residue are strongly correlated, since rotations produced in ϕ will, in general, cause stereochemical constraints in angle ψ. Furthermore, every time one sees a Ramachandram plot, it is correlating ϕ and ψ of the same amino acid.
Two methods to handle the bivariate data were implemented. One uses two-dimensional Kernel Density Estimation (KDE). For each $[\phi; \psi]$ pair, it creates a kernel density map from the real values of s. Then, the ϕ values are generated independently, and ψ values are generated conditional on the values of ϕ. To do that, we need also to create a one-dimensional KDE of ϕ and sample a value from that distribution. Next, we choose the closest point from value ϕ in our just created two-density map. At this point, there is a one-dimensional KDE that is conditional to the previous value of ϕ. Finally, we generate a new ψ value based on that distribution.
The second method uses a two-dimensional Finite Gaussian Mixture (FGM) in order to estimate values of the pair $[\phi; \psi]$. For each $[\phi; \psi]$ pair, it performs a whole bivariate Expectation-Maximization (EM) algorithm until convergence, for a given number of components. The value of $[\phi; \psi]$ of the offspring is generated at once. First, a component is randomly chosen with probability π. Next, a bivariate Gaussian random number is generated according to the mean and covariance matrix of the chosen component.

In order to keep performance, in both two-dimensional KDE and FGM methods, we generated all these steps for all individuals of the offspring before going to the next pair $[\phi; \psi]$.

5 Results

All techniques were performed, which changed the population size, selection pressure and tournament size. Specifically, UNI also had the σ changed and FGM also had the number of components and λ (used to avoid singular matrices) modified. That rendered in a total of 1680 combinations and took over than 3000 hours of CPU time of the LNCC cluster. Figure 1 shows the results achieved. At the left, a performance comparison is made. As we expected, UNI was the fastest. FGM was better than KDE, despite it has some outliers caused by the runnings with the high number of mixture components. Figure 1 (middle) shows a scatterplot between energy and RMS (quality measurement in PSP: low RMS means high quality). Points of Pareto Front are highlighted by a dashed line. Finally, Figure 1 (right) shows a protein configuration predicted (blue) fitted with a native (green).

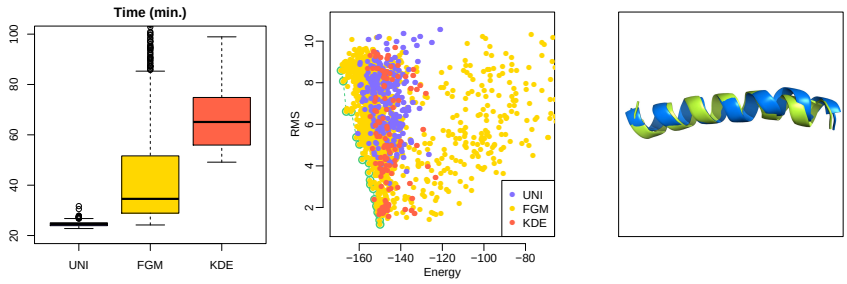

FIGURE 1. Left: Performance comparison; Middle: Scatterplot between energy and RMS; Right: A predicted configuration (blue) aligned to native (green).

References

Bujnicki, J. M (2009). *Prediction of Protein Structures, Functions, and Interactions* West Sussex: Wiley

Larranaga, P. and Lozano, J. A. (2002). *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation* New York: Spring

McLachlan, G. and Peel, D. (2004). *Finite Mixture Models*. Hoboken, NJ, USA: Wiley

Venables, W. N. and Ripley, B. D. (2002). *Statistics with S Fourth edition*. New York: Springer.

Daniel Bonetti is PhD student in Instituto de Cincias Matemticas e de Computao (ICMC) at Universidade de So Paulo (USP), Brazil. He is bachelor in Computer Science by Escola de Engenharia de Piracicaba, Brazil and got his Master's degree in 2010 at ICMC - USP with the work Enhance the Van der Waals energy efficiency calculi in genetic algorithms for protein structure prediction. His research interests include Evolutionary Algorithms, High Performance Computing and Protein Structure Prediction.