The Role of a Priori Information in the Minimization of Contact Potentials by Means of Estimation of Distribution Algorithms

Roberto Santana, Pedro Larrañaga, and Jose A. Lozano

Department of Computer Science and Artificial Intelligence
University of the Basque Country, Donostia-San Sebastian, Spain
rsantana@si.ehu.es, pedro.larranaga@ehu.es, ja.lozano@ehu.es

Abstract. Directed search methods and probabilistic approaches have been used as two alternative ways for computational protein design. This paper presents a hybrid methodology that combines features from both approaches. Three estimation of distribution algorithms are applied to the solution of a protein design problem by minimization of contact potentials. The combination of probabilistic models able to represent probabilistic dependencies with the use of information about residues interactions in the protein contact graph is shown to improve the efficiency of search for the problems evaluated.

Keywords: estimation of distribution algorithm, protein design, energy minimization algorithms.

1 Introduction

The goal of protein design is to find sequences of aminoacids with desired structural and functional properties. The problem has been approached by the application of directed search methods which cast the search as an optimization method. The approach requires the definition of a simplified model of the proteins, a fitness function that associates a value to each solution according to its 'quality', and a search procedure to efficiently sample the search space. In the field of protein design, these methods have been called “directed approaches to protein design”.

Another class of methods has been covered under the umbrella of “probabilistic approaches to protein design” [14]. They use site-specific aminoacid probabilities rather than specific sequences and are usually employed in domains where the information available about the problem is incomplete. Probabilistic approaches include the use of consensus sequences [8] to determine low energy sequences and other methods where the probabilities learned can be used to guide search algorithms.

In this paper, we present a different approach which is based on the use of estimation of distribution algorithms (EDAs) [7,9,13]. EDAs are evolutionary algorithms that construct an explicit probability model of a set of selected solutions.

E. Marchiori, J.H. Moore, and J.C. Rajapakse (Eds.): EvoBIO 2007, LNCS 4447, pp. 247–257, 2007.

EDAs have been used for protein structure prediction in simplified models [17], protein side chain placement [18] and *de novo* peptide design [2]. Their suitability to deal with protein problems is given by the incorporation of machine learning techniques in the construction of the models. These learning algorithms automatically extract relevant regularities and complex structural patterns shared by promising solutions. The information learned can be compactly stored in the probabilistic model, which is later used to guide the exploration of the search space. EDAs are also different from probabilistic approaches that use probabilities to bias the search (e.g. Monte Carlo based techniques [21]) and where probabilities are unchanged during the search.

The paper is organized as follows. In the following section we present the energy function and introduce the problem of finding the aminoacid sequence with the lowest energy. Section 3 describes the main characteristics of EDAs and introduces the EDAs based on tree models used in our application. Section 4 gives a description of the experimental framework. The numerical results are shown in Section 5. Section 6 presents the main conclusions of our work and discuss future work.

2 Approach to Protein Design: Finding the Sequence with the Lowest Energy

In this section, we introduce the problem of finding the aminoacid sequence with the lowest energy for a given energy function. We use X_i to represent a discrete random variable. A possible value of X_i is denoted x_i. Similarly, we use $\mathbf{X} = (X_1, \ldots, X_n)$ to represent an n-dimensional random variable and $\mathbf{x} = (x_1, \ldots, x_n)$ to represent one of its possible values.

We will approach the protein design problem following a strategy that is based on the optimization of contact functions. Contact potentials or scoring functions [19,20] measure how likely it is for a sequence to fold to a given structure. Although the potential functions have been mainly used to distinguish native from decoy structures [19,20], they can also be employed to study the distribution of native-like features in sequence space [11].

In [10,11], the sequence evolutionary selection mechanisms are analyzed focusing on the stability energy of sequences. Although the 'survival probability' of a protein sequence depends on a number of other factors such as protein function and protein flexibility, the sequence-structure relationship can be analyzed in terms of energy. The analysis assumes that native sequences were selected because they were highly probable as a function of energy.

We will denote the native sequence corresponding to the structure σ as $\mathbf{x}^{\sigma}$. $E(\mathbf{x}, \sigma)$ is the energy of sequence $\mathbf{x}$ in structure σ and $E_{\sigma} = E(\mathbf{x}^{\sigma})$ is the native energy of sequence $\mathbf{x}^{\sigma}$ in structure σ. The quantity $N(E_{\sigma})$ is the number of sequences whose energy in σ would be no greater than that of the actual native sequence. $N(E_{\sigma})$ is called the *evolutionary capacity* of structure σ because it reflects how far the current state of molecular evolution σ is from the possible optimum in terms of energy [11].

Given a protein structure σ and an energy function defined on the space of aminoacid sequences with cardinality 20^n, where n is the number of aminoacids, we address the problem of finding the lowest energy sequence among the 20^n possible solutions.

2.1 Energy Function

The $TE13$ potential function was introduced in [19] to correctly rate the native structure in relation to a set of decoy structures. This potential function was calculated using a linear programming approach.

Before presenting the function, let us to introduce some notation. $u_{\alpha\beta}(r)$ will denote a step potential between a pair of aminoacids α and β, and $p_{\alpha\beta}$ its asociated parameter calculated from solving linear inequalities. The distance between the geometric centers of two aminoacid side chains, r, is divided into 13 steps between 2 and 9 $\mathring{A}$. The first step along r is between 2 and 3 $\mathring{A}$, and the rest of the 12 steps are 0.5 $\mathring{A}$ each. Each of the $u_{\alpha\beta}(r)$ (as a function of the index β) is 1 only at one of the windows (steps) and zero elsewhere.

The total potential energy is:

$$E(\mathbf{x}, \sigma) = \sum_{\alpha,\beta} p_{\alpha\beta} n_\alpha u_{\alpha\beta}(r) \equiv \sum_{\lambda} p_\lambda n_\lambda \qquad (1)$$

where index α parameterizes the type of the two interacting aminoacids. n_α is the number of contacts of a specific type found when threading the complete sequence $\mathbf{x}$ into the known shape σ. n_α and $u_{\alpha\beta}(r)$ are combined together to form n_λ, the number of contacts of a specific type and at a specific distance, λ, of structure σ. For each of the λ-s, there is a corresponding independent parameter p_λ calculated from solving linear inequalities. The total number of parameters is $((21 \times 20)/2) \times 13 = 2730$.

$TE13$ is defined for the distances between the side chain centers that have to be given. Therefore, the side chain center has to be given and from this information we calculate the distance r for each pair of residues. We construct the contact graph of each protein considering the existence of edges between two vertices if the corresponding residues have contact distances below $9\mathring{A}$.

3 Estimation of Distribution Algorithms

We will work with positive probability distributions denoted by $p(\mathbf{x})$. Similarly, $p(\mathbf{x}_S)$ will denote the marginal probability distribution for $\mathbf{X}_S$, where $S \subset \{1, \ldots, n\}$.

In EDAs, each individual represents one possible solution and it is encoded using the vector representation introduced above. A key characteristic and crucial step of EDAs is the construction of the probabilistic model.

The simplest EDA is the univariate marginal distribution algorithm (UMDA) which uses a probabilistic model where all variables are considered independent. The probabilistic model used by UMDA is described by Equation (2).

$$p_{UMDA}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i) \tag{2}$$

The probability of each solution is equal to the product of the variables' univariate probabilities.

In this paper, we will apply UMDA to the minimization of contact potentials. We also propose the application to the protein design problem of a model that captures bivariate dependencies between the variables. This probabilistic model is based on a tree where each variable may depend on at most another variable that is called the parent.

A probability distribution $p_{Tree}(\mathbf{x})$ that is conformal with a tree is defined as:

$$p_{Tree}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|pa(x_i)) \tag{3}$$

where $pa(x_i)$ denotes a configuration of $Pa(X_i)$, the parent of X_i in the tree, and $p(x_i|pa(x_i)) = p(x_i)$ when $Pa(X_i) = \emptyset$, i.e. X_i is the root of the tree. The distribution $p_{Tree}(\mathbf{x})$ itself will be called a tree model when no confusion is possible. Probabilistic trees are represented by acyclic connected graphs.

The construction of the tree structure from data implies the detection of the most important bivariate interactions between the variables. This can be done applying statistical independence tests [15] or methods based on the analysis of the mutual information between variables as in [1]. We follow the second approach. The pseudocode of the tree-based EDA (Tree-EDA) is shown in Algorithm 1.

Algorithm 1: **Tree-EDA**

```
$D_0$ ← Generate $M$ individuals randomly
$l = 1$
do {
    $D^s_{l-1}$ ← Select $N \leq M$ individuals from $D_{l-1}$ according to a selection method
    Compute the univariate and bivariate marginal frequencies $p^s_i(x_i|D^s_{l-1})$ and
    $p^s_{i,j}(x_i, x_j|D^s_{l-1})$ of $D^s_{l-1}$
    Calculate the matrix of mutual information using univariate and bivariate
    marginals
    Calculate the maximum weight spanning tree from the matrix of mutual
    information
    Compute the parameters of the model
    $D_l$ ← Sample $M$ individuals (the new population) from the tree
} until A stop criterion is met
```

As presented in Algorithm 1, the bivariate probabilities are initially calculated for every pair of variables. From these bivariate probabilities, the mutual information between variables is found. To construct the tree structure, an algorithm introduced in [3], that calculates the maximum weight spanning tree from the matrix of mutual information between pairs of variables, is used. To sample the solutions from the tree, we have used probabilistic logic sampling (PLS) [5].

One problem faced by the algorithms that learn probabilistic models within EDAs is the arousal of spurious correlations between variables. This is due, among other factors, to the small datasets used to estimate the probabilities. Spurious correlations deteriorate the accuracy of the models and negatively influence the efficiency of the search. Our initial experiments showed that the quality of the learned trees could be improved when the interactions represented in the tree structure were constrained to those between residues that are making contacts in the contact graph of the protein as it was defined in Section 2.1. Therefore, one variant of the tree learning algorithm constrains the calculation of bivariate probabilities and mutual information to those pairs of variables corresponding to residues that are in contact in the contact graph. The variant of Tree-EDA that restricts the interactions represented by the tree structure to interacting pairs of variables is called Tree-EDA^r.

Tree-EDA^r can be considered as an example of the class of EDAs that use a priori problem information in order to improve the search. However, the information about the structure does not completely determine the final structure of the model as it is common in other EDAs that employ a priori problem information [12,16]. Instead, the mutual information between variables in the current selected population is taken into account to define the final structure of the probabilistic model.

4 Experiments

The objectives of our experiments are:

- To determine the ability of the probability model used by EDAs to capture relevant features of the protein design problem considered.
- To evaluate the capacity of UMDA, Tree-EDA and Tree-EDA^r to solve the energy minimization problem.
- To establish the influence of using information about the contact graph of the protein in the quality of the solutions obtained.

To search sequences with the lowest energy, we selected a set of 61 protein instances[1] from an initial set of 3901 protein instances[2]. This is a reduced and non-redundant set of protein shapes used for fold recognition. It is a good representative of the known folds of the protein databank [10]. For each protein, information about the side chain geometric centers of the protein is available[3].

[1] The list of the selected sequences is available from http://www.sc.ehu.es/ccwbayes/EDA/EDAProteinProblems.html

[2] These instances have been obtained from Prof. Leonid Meyerguz's page: http://www.cs.cornell.edu/∼leonidm/counting/protein_list.txt

[3] http://www.cs.cornell.edu/∼leonidm/counting/pdb_sample.tar

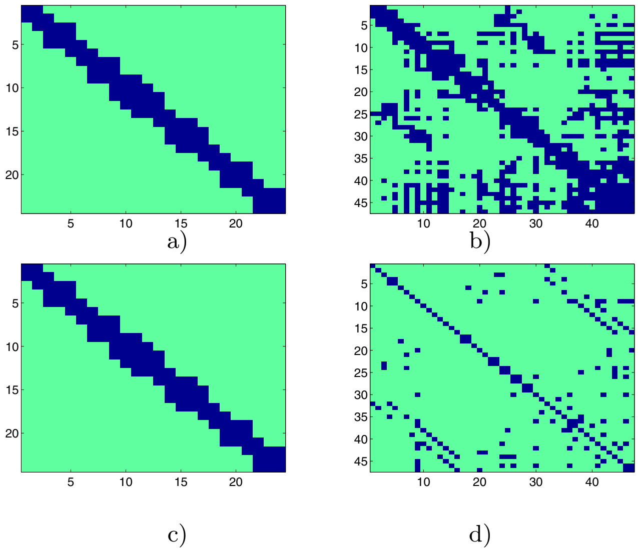


Fig. 1. Contact matrices of proteins pdb4clg-A (a) and pdb1aoo (b). Most frequent interactions found by Tree-EDA in the first generation for proteins pdb4clg-A (c) and pdb1aoo (d).

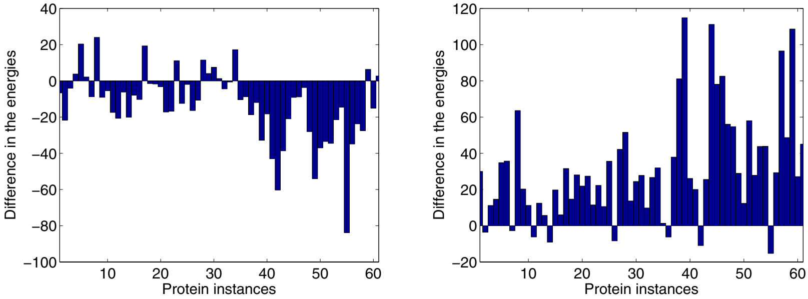
4.1 Parameters of the Algorithms

The parameters of the EDAs have been set as follows. The population size was set at 5000. The maximum number of generations is 500. Truncation selection with parameter $T = 0.15$ has been used. In this selection scheme, the best $T \cdot N$ individuals of the population are selected to construct the probabilistic model. We apply a replacement strategy called best elitism in which the selected population at generation t is incorporated into the population of generation $t+1$, keeping the best individuals found so far and avoiding to revaluate their fitness function. The algorithm stops when the maximum number of generations is reached or the selected population has become too homogeneous (no more than 10 different individuals).

4.2 Design of the Experiments

To compare the results of the algorithms we conducted 50 experiments for each instance and algorithm. The performance of the algorithms was evaluated considering the fitness of the best solution found in each experiment, the best fitness among all the best solutions found, and the number of experiments in which the best fitness overall the 50 experiments was found.

To determine whether differences between the fitness of the solutions found by the algorithms are statistically significant the Kruskal-Wallis test [6] was employed. The test significance level was 0.05.



a) b)

Fig. 2. Improvement in the energies of the solutions found by Tree-EDA and Tree-EDA^r with respect to those found by UMDA for the 61 protein instances. a) Difference between the average energies of Tree-EDA and UMDA. b) Difference between the average energies of Tree-EDA^r and UMDA.

5 Numerical Results

As an initial experiment, we investigated if there was any mapping between the statistical dependencies learned by the tree learning algorithm and the structure of the problem. We stored the tree structures learned in the first generation of Tree-EDA for protein instances pdb4clg-A and pdb1aoo (see footnote 1). The algorithm was run 1000 times for each instance. We counted the number of times each edge appeared in the trees. Figure 1a) and 1b) show the contact matrices constructed for edges that were in at least 50 of the 1000 trees learned. The similarity between the structures learned and the original contact matrices of the graphs (Figures 1c) and 1d)) is remarkable.

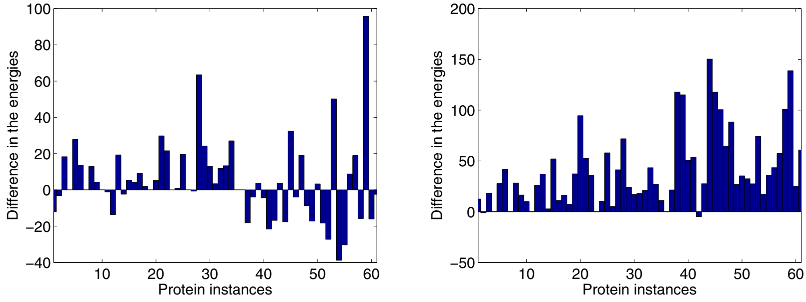
The decision of analyzing only the structures learned in the first generation was motivated by the fact that, as the EDA evolves, diversity in the population is lost and spurious correlations due to sample arise. The possibility of capturing many of the interactions that are in the original contact graph shows that EDAs are able to recover problem information from the protein structure from a statistical analysis of the data generated along the evolution.

We evaluated the performance of EDAs to sample the space of low energy sequences for function $TE13$. We do not have information about the actual lowest energy sequences. Therefore, the comparison between algorithms can be done only in relative terms. UMDA, Tree-EDA and Tree-EDA^r were run on the 61 instances of the protein benchmark. In Table 1, the results of Tree-EDA and Tree-EDA^r for each of the instances are presented. The table shows the best value of the fitness[4] found in all the experiments ($-f$), the number of times the best solution has been found (S) and the average fitness ($\bar{f}$) of the solutions. It can seen from the table that, in terms of best solution found and the average

[4] Original fitness values are negative, however notice that the table actually shows their opposite values.

Table 1. Results of the Tree-EDA and Tree-EDA^r

pdb	Tree-EDA			Tree-EDA^r			pdb	Tree-EDA			Tree-EDA^r		
	$-f$	S	$-f$	$-f$	S	$-f$		$-f$	S	$-f$	$-f$	S	$-f$
1afp	1493.46	1	1458.46	1494.83	1	1468.78	1zwd	658.84	1	621.14	663.07	2	642.17
1apq	1426.39	1	1367.53	1442.43	2	1405.37	2mrb	806.87	1	784.90	808.44	4	797.63
1bba	800.31	7	775.36	800.31	20	788.67	2pta	882.57	1	837.70	903.44	1	867.30
1bh4	758.44	3	734.41	759.42	4	743.75	3ins-A	446.54	2	418.86	446.54	12	428.22
1bkv-A	425.23	25	411.48	425.23	56	419.71	3znf	679.72	1	668.07	682.03	7	670.64
1bqf-A	479.03	5	458.60	479.03	18	466.56	8tfv-A	368.23	80	367.04	368.23	85	367.42
1btd-A	703.33	1	678.32	726.62	1	694.10	1aoj-A	1252.12	1	1163.01	1258.01	1	1226.78
1ciq-B	452.90	45	449.99	452.90	62	451.01	1b7d-A	1750.34	1	1666.33	1822.28	1	1743.25
1clv-I	926.78	1	892.70	933.72	4	913.21	1bbr-H	3615.33	1	3458.34	3947.21	1	3798.36
1dec	947.68	1	907.02	955.11	3	924.31	1bf0	1665.05	1	1596.53	1711.79	1	1643.32
1dfn-A	695.52	1	668.45	703.26	2	676.09	1cdq	2122.12	1	2043.20	2192.00	1	2100.08
1eiu	907.32	6	898.07	908.19	1	903.53	1doy	3324.07	1	3266.49	3373.96	1	3325.90
1gnf	1167.75	1	1116.09	1172.37	1	1147.62	1ehd-A	3243.84	1	3159.73	3282.18	1	3236.33
1gps	1323.88	1	1284.45	1329.07	1	1301.12	1eo0-A	2242.45	1	2070.38	2389.50	1	2227.81
1iva	1300.78	1	1282.35	1301.56	3	1294.18	1fxr-A	1924.04	1	1831.44	1988.22	1	1910.87
1ktx	1002.84	1	941.52	1016.46	1	960.75	1gam-A	2356.16	1	2266.02	2465.97	1	2386.30
1mct-I	766.62	1	745.84	771.84	2	757.00	1gat-A	1495.89	1	1401.05	1513.75	1	1457.24
1mea	691.47	2	664.44	691.47	7	679.30	1hd0-A	2282.86	1	2179.34	2331.39	1	2252.34
1mhu	907.43	1	880.05	907.43	4	889.83	1if1-A	2984.11	1	2840.86	3051.07	1	2993.47
1myn	1289.52	1	1230.85	1299.16	1	1263.31	1imp	2478.64	1	2374.79	2513.36	1	2458.20
1pnh	687.83	1	669.68	784.83	1	691.89	1ivl-A	3338.91	1	3195.89	3470.71	1	3407.85
1pyc	1101.69	1	1050.11	1123.14	1	1083.41	1kst	2166.32	1	2080.83	2203.51	1	2138.72
1qdp	1104.75	1	1060.80	1114.81	2	1083.88	1nra	1888.56	1	1806.72	1914.91	1	1863.79
1qfn-B	338.55	28	330.84	338.55	65	335.45	1pba	2154.58	1	2033.58	2199.66	1	2112.74
1qk6-A	816.63	1	780.12	817.42	2	797.95	1qd9-A	3929.16	1	3762.24	4026.28	1	3952.83
1res	1038.58	1	981.14	1069.92	1	1026.72	1vfy-A	1857.82	1	1797.28	1912.55	1	1844.27
1roo	975.74	32	969.57	977.36	1	971.67	1whf	2443.19	1	2293.68	2487.02	1	2424.10
1sh1	1405.55	1	1348.21	1436.10	1	1386.09	2hgf	2923.62	1	2848.07	3046.69	1	2960.59
1sp2	673.46	1	608.36	673.46	18	652.37	2r63	1829.48	1	1678.30	1912.22	1	1802.58
1ter	528.31	4	503.82	528.31	26	513.96	4mt2	1864.31	1	1809.06	1884.13	2	1849.47
5cro	1527.11	1	1449.72	1605.88	1	1516.00							



a) b)

Fig. 3. Improvement in the energies of the solutions found by Tree-EDA and Tree-EDA^r with respect to those found by UMDA for the 61 protein instances. a) Difference between the energies of the best solutions overall run found by Tree-EDA and UMDA. b) Difference between the energies of the best solutions found by Tree-EDA^r and UMDA.

fitness, Tree-EDA^r consistently finds better results than Tree-EDA. However, for most of instances, the best solution is found only once.

Figures 2 and 3 show the improvements achieved by Tree-EDA and Tree-EDA^r with respect to UMDA. Negative values mean that UMDA outperformed the other algorithm. Figures 2 a), b) show the improvement taking into account the average energies calculated from the best solution found in each of the 50 experiments conducted. In Figure 3 a), b), the improvement was calculated considering the absolute best solution found among the 50 experiments.

The application of the Kruskal-Wallis test found significant statistical differences between UMDA and Tree-EDA for 24 of the 61 instances. For 19 of these instances, UMDA was better than Tree-EDA. Significant statistical differences between UMDA and Tree-EDA^r were found for 52 of the 61 instances. For 49 of these instances, Tree-EDA^r outperformed UMDA. The statistical tests confirmed what can be seen from the figures: Considering the average energy, Tree-EDA was not able to improve results found by UMDA. For the best solutions, Tree-EDA found better solutions than UMDA in only 32 of the 61 instances. Nevertheless, the results of the statistical tests showed that the performance of Tree-EDA^r was consistently and clearly superior to UMDA both in average and best solutions. This example shows that the use of problem information can be critical for successful application of EDAs that consider interactions. The failure of Tree-EDA to improve (on average) results achieved by UMDA may be due to a small population size, insufficient to accurately compute the statistics needed to detect the correct interactions.

6 Conclusions and Future Work

EDAs belong to a new class of stochastic optimization methods that are able to capture, by the application of machine learning techniques, relevant features about the problem domain. The results presented in this paper show how solutions achieved in protein design problems can be improved by the use of probabilistic models able to represent interactions between the variables. This general procedure can be applied to other problems in protein design. Furthermore, as it has been shown in this paper, the information stored in the probabilistic models learned during the search could be useful to reveal important characteristics about the problem domain.

As a future research trend we envision the application of EDAs to more complex models used for protein design. In particular, we consider EDAs could be applied to the solution of the protein design problem using backbone dependent rotamer libraries [4] similarly to the way they have been used for protein design in [18].

Acknowledgements

The authors thank Leonid Meyerguz for providing the set of instances used in our experiments. This work was supported by the SAIOTEK-Autoinmune (II)

2006 and Etortek research projects from the Basque Government. It has been also supported by the Spanish Ministerio de Ciencia y Tecnología under grant TIN 2005-03824.

References

1. S. Baluja and S. Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In *Proceedings of the 14th International Conference on Machine Learning*, pages 30–38. Morgan Kaufmann, 1997.
2. I. Belda, S. Madurga, X. Llorá, M. Martinell, T. Tarragó, M. Piqueras, E. Nicolás, and E. Giralt. ENPDA: An evolutionary structure-based de novo peptide design algorithm. *Journal of Computer-Aided Molecular Design*, 19(8):585–601, 2005.
3. C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.
4. R. L. Dunbrack. Rotamer libraries in the 21st century. *Current Opinion in Structural Biology*, 12:431–440, 2002.
5. M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In J. F. Lemmer and L. N. Kanal, editors, *Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence*, pages 149–164. Elsevier, 1988.
6. J. C. Hsu. *Multiple Comparisons: Theory and Methods*. Chapman and Hall, 1996.
7. P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.
8. M. Lehmann, D. Kostrewa, M. Wyss, R. Brugger, A. D'Arcy, L. Pasamontes, and A. van Loon. From DNA sequence to improved functionality: Using protein sequence comparisons to rapidly design a thermostable consensus phytase. *Protein Engineering*, 13:49–57, 2000.
9. J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, editors. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Springer-Verlag, 2006.
10. L. Meyerguz, C. Grasso, J. Kleinberg, and R. Elber. Computational analysis of sequence selection mechanisms. *Structure*, 12(4):547–557, 2004.
11. L. Meyerguz, D. Kempe, J. Kleinberg, and R. Elber. The evolutionary capacity of protein structures. In *Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology*, pages 290–297, San Diego, California, 2004. Morgan Kaufmann Publishers, San Francisco, CA.
12. H. Mühlenbein, T. Mahnig, and A. Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):213–247, 1999.
13. H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In *Parallel Problem Solving from Nature - PPSN IV*, pages 178–187, Berlin, 1996. Springer Verlag. LNCS 1141.
14. S. Park, H. Kono, W. Wang, E. T. Boder, and J. G. Saven. Progress in the development and application of computational methods for probabilistic protein design. *Computers and Chemical Engineering*, 29:407–421, 2005.
15. M. Pelikan and H. Mühlenbein. The bivariate marginal distribution algorithm. In R. Roy, T. Furuhashi, and P. Chawdhry, editors, *Advances in Soft Computing - Engineering Design and Manufacturing*, pages 521–535, London, 1999. Springer-Verlag.

16. R. Santana, E. P. de León, and A. Ochoa. The edge incident model. In *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 352–359, Habana, Cuba, March 1999.
17. R. Santana, P. Larrañaga, and J. A. Lozano. Protein folding in 2-dimensional lattices with estimation of distribution algorithms. In *Proceedings of the First International Symposium on Biological and Medical Data Analysis*, volume 3337 of *Lecture Notes in Computer Science*, pages 388–398, Barcelona, 2004. Springer Verlag.
18. R. Santana, P. Larrañaga, and J. A. Lozano. Side chain placement using estimation of distribution algorithms. *Artificial Intelligence in Medicine*, 39(1):49–63, 2006.
19. D. Tobi and R. Elber. Distance-dependent, pair potential for protein folding: Results from linear optimization. *Proteins*, 41(1):40–46, 2000.
20. J. Zhu, Q. Zhu, Y. Shi, and H. Liu. How well can we predict native contacts in proteins based on decoy structures and their energies? *Proteins: Structure, Function, and Genetics*, 52(4):598–608, 2003.
21. J. Zou and J. G. Saven. Using self-consistent fields to bias Monte Carlo methods with applications to designing and sampling protein sequences. *The Journal of Chemical Physics*, 118(8):3843–3854, 2003.