Improving the Efficiency of MECoMaP: A Protein Residue-Residue Contact Predictor

Alfonso E. Márquez-Chamorro[1], Federico Divina[1], Jesús S. Aguilar-Ruiz[1], and Cosme E. Santiesteban-Toca[2]

[1] School of Engineering, Pablo de Olavide University of Sevilla, Spain
{amarcha,fdivina,aguilar}@upo.es
[2] Centro de Bioplantas, University of Ciego de Avila, Cuba
cosme@bioplantas.cu

Abstract. This work proposes an improvement of the multi-objective evolutionary method for the protein residue-residue contact prediction called MECoMaP. This method bases its prediction on physico-chemical properties of amino acids, structural features and evolutionary information of the proteins. The evolutionary algorithm produces a set of decision rules that identifies contacts between amino acids. These decision rules generated by the algorithm represent a set of conditions to predict residue-residue contacts. A new encoding used, a fast evaluation of the examples from the training data set and a treatment of unbalanced classes of data were considered to improve the the efficiency of the algorithm.

Keywords: protein structure prediction, residue-residue contact, multi-objective optimization, evolutionary computation.

1 Introduction

One of the central goals of bioinformatics is the prediction of protein function and tertiary structure from the linear sequence of amino acids (primary structure). Determining the three dimensional structure of proteins is necessary to understand the functions of molecular protein level. On the other hand, misfolding proteins can be the principal cause of some diseases. Since protein function is determined by its structure, a misfold implies that a protein can not fulfill its function correctly. Alzheimer's disease, cystic fibrosis, bovine spongiform encephalopathy (mad cow disease) and its human variant are now all attributed to protein misfolding. The knowledge of the misfolding factors and understanding the protein folding process, would help in developing cures for these diseases.

The primary structure, or amino acid sequence, of a protein is much easier to determine than its tertiary structure. Moreover, the gap between the number of proteins with known sequence and the number of proteins with known tertiary structure is rapidly increasing. In order to reduce this gap, there have been many researches focused on determining the tertiary structure of a protein from its sequence [1,2]. The high number of protein sequences whose three-dimensional structures must be determined, make computational methods for protein structure prediction (PSP) an essential tool. We believe that EAs well suited for solving the PSP problem, since PSP can be seen as a search problem through the space determined by all the possible protein foldings. Moreover, PSP problem can be considered as a optimization problem with several objectives [3]. The task of finding one or more suboptimal solutions is called Multi-objective optimization. Our algorithm is based on these approaches.

An useful, and commonly used, representation for protein 3D structure is the protein contact map, which represents binary proximities (contact or non-contact) between each pair of amino acids of a protein. Our approach is included in this category.

The aim of this work consists of improving our proposal MECoMaP (Multi-objective Evolutionary Contact Map Predictor) [4] in order to increase the efficiency of the protein contact map prediction. The prediction is based on three physico-chemical properties: hydrophobicity (H), polarity (P) and charge (C), structural features: solvent accessibility (SA) and secondary structure (SS) and evolutionary information in form of Position Specific Scoring Matrix (PSSM). It is known that amino acid properties play an important role in the PSP problem [5]. Several PSP methods rely on amino acids properties, *e.g.*, HP models. On the other hand, a vast majority of PSP algorithms used SS, SA and PSSM as predictive features.

The remainder of this paper is organized as follows. Our multi-objective evolutionary approach is described in section 2. Section 3 presents the experimentation and obtained results. Finally, section 4, includes some conclusions and possible future works.

2 Methodology

MECoMaP is based on the Strength Pareto Evolutionary Algorithm (SPEA). Each individual of the population represents a decision rule. In particular, rules are based on the previously mentioned amino acid properties. Basically rules specify a set of conditions on each property, that, if satisfied, predict a contact between two amino acids.

In the following the preparation of data, attribute selection, the encoding, the fitness function and the genetic operators used by the EA will be presented.

2.1 Preparation of Data

We selected from PDB a protein data set (DS1) that consists of 173 non-redundant proteins with sequence identity less than 25%, and was obtained from [6]. The minimum and maximum lengths of proteins are 31 and 753 amino acids, respectively. DS1 contains 240501 positive examples (contacts) and 5034050 negative examples (non-contacts).

The second data set (DS2), with 53 non-redundant and non-homologous globulin proteins, is detailed in [7]. The sequence identity of DS2 dataset is also lower than 25%. DS2 is formed by a total of 30546 contacts and 356528 non-contacts.

As we can see, the positive and negative classes (contact and non-contacts) are notably unbalanced. We have performed a resampling of data using 1:1 and 2:1 contact/non-contacts ratios. Using 1:1 ratio we obtain a higher rate of predicted contacts, however the rate of false positives of the predictor is increased. Specifically, the accuracy results for both ratios on DS1 and DS2 are shown in Table 1. As seen in the table, the 2:1 ratio presented better performance. This is also the case for DS2 data set. The optimization of this parameter also implies a lower computational cost for the algorithm. Based on the results of the table, we decided to perform a re-sampling using the 2:1 ratio.

Table 1. Average accuracy results obtained for different contact/non-contacts ratios for the DS1 and DS2 protein data set

Ratio	Data Set	$Accuracy_\mu$
1:1	DS1	$0.21_{\pm 0.10}$
2:1	DS1	$0.23_{\pm 0.08}$
1:1	DS2	$0.16_{\pm 0.13}$
2:1	DS2	$0.20_{\pm 0.11}$

2.2 Feature Selection

As stated before, the prediction is based on a set of amino acid properties which are very important in the folding process. The reason for basing the prediction on such properties, is that it has been shown that amino acids that are in contact, are characterized by similar properties [8]. We selected Kyte-Doolittle hydropathy profile [9], the Grantham profile [10] for polarity and the Klein scale for net charge [11]. Hydrophobic amino acids are generally found in the inner of proteins protected from direct contact with water. Inversely, the hydrophilic amino acids are generally found on the outside of proteins as well as in the active centers of enzymatically active proteins. The net charge takes into account the charged groups present in any amino acid, peptide or protein nd the pH of its environment. In addition to these properties, we also use two structural features of proteins (SS and SA) and evolutionary information, in form of PSSM.

Secondary structure prediction consists of predicting the location of α-helices, β-sheets and turns from a sequence of amino acids. The location of these motifs could be used by approximation algorithms to obtain the tertiary structure of the protein. We obtain SS predictions using PSIPRED. SA refers to the degree to which a residue interacts with the solvent molecules. The prediction of SA value is performed using ICOS Server for the prediction of structural aspects of protein residues *http://cruncher.cs.nott.ac.uk/psp/prediction*.

A PSSM determines the substitution scores between amino acids according to their positions in the alignment. Each cell of the matrix represents the observed substitution frequency at a given position divided by the expected substitution frequency at that position. PSSM is obtained using PSI-BLAST.

H, P and PSSM values were normalized between -1 and 1. C values are represented with -1, 0 and 1 for negative, neutral and positive charges. SS values are identified with 1, 2 and 0 for alpha-helices, beta-sheets and random coils, respectively. SA values are ranging from 0 to 4 according to the exposure level.

The procedure scheme of preproccessing of the data is represented in Figure 1. We have obtained five different files with the information of the properties. They constitute the training data of the algorithm.

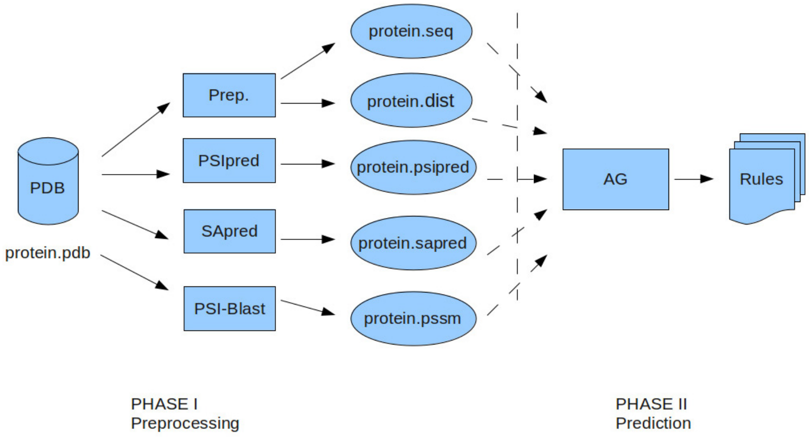

Fig. 1. Preprocessing procedure scheme

2.3 Encoding

An individual is constituted by six blocks which represent the different properties of amino acids. Each block indicates the values of a respective property in all the positions of the residues in the window. We use two windows of ±3 residues centered around the two target amino acids i and j. Therefore, one window is relative to amino acids $i-3, i-2, i-1, i, i+1, i+2, i+3$ and the other one is relative to amino acids $j-3, j-2, j-1, j, j+1, j+2, j+3$.

We define each individual as a decision rule $R_{i,j}$ for amino acids i and j:

$$R_{ij} = \{\{H_{min}, H_{max}\}^{1..n}, \{P_{min}, P_{max}\}^{1..n},$$
$$C^{1..n}, SS^{1..n}, SA^{1..n}, \{PSSM_{min_{ij}}, PSSM_{max_{ij}}\}^{1..20}\} \quad (1)$$

where n indicates the total number of amino acids (in this case $n = 14$). Each element of R_{ij} must fulfill the following requirements:

$$
\begin{gathered}
-1 \leq H_{min} < H_{max} \leq 1 \\
-1 \leq P_{min} < P_{max} \leq 1 \\
C \in \{-1, 0, 1\} \\
SS \in \{-1, 0, 1, 2\} \\
SA \in \{-1, 0, 1, 2, 3, 4\} \\
-1 \leq PSSM_{min}^{1..20} < PSSM_{max}^{1..20} \leq 1
\end{gathered}
\qquad (2)
$$

This decision rule determines whether two amino acids i and j are in contact, where $1 \leq i < j \leq L$, being L the sequence length. Our representation consists in 14×2 attributes for H, 14×2 for P, 14 for C, 14 for SS, 14 for SS and $2 \times 2 \times 20$ for PSSM, 178 attributes in total.

2.4 Fitness Function

As stated in [4], we consider two objectives to be optimized: coverage and accuracy. Coverage represents the number of predicted contacts and accuracy evaluates the real predicted contacts rate. Therefore, $Coverage = C/C_t$ and $Accuracy = C/C_p$, where C is the number of correctly predicted contacts of a protein, C_t is the total number of contacts of the protein and C_p is the number of predicted contacts. We aim at finding the best compromise between these two measures. The fitness of an individual x is given by the number of individuals that x dominates.

2.5 Genetic Operators

A 2-point crossover operation was employed with a binary tournament selection and a 0.5 probability. In each tournament, we select the individual which is located in the better Pareto front.

A first mutation operator follows a Gaussian distribution for a randomly selected individual. This operator increases or decreases a gene value with a probability of 0.5 randomly interval. A second mutation operator randomly selects a gene that is related to a given property, with a 0.1 probability, and moves the bounds to the maximum or minimum of the domain, making the property irrelevant in this rule. For example, if the property is the polarity, we change the range to -1, 1 so the rule does not take into account this property in this case. After the mutation, we test if the obtained values are in the adequate ranges for the corresponding property.

The population size is set to 100, and the initial population is randomly initialized with a 0.6 probability. The maximum number of generations that can be performed is set to 100. However, if the fitness of the best individual does not increase over twenty generations, the algorithm is stopped and a solution is provided. At the end of the execution, repeated or redundant rules are discarded from the solution set.

2.6 Efficient Evaluation Structure

In order to reduce the computational time of our method, we have implemented an AVL tree [12] to order and classify the training examples according to their property values. This tree organizes the information in such a way that it is not necessary to process all the examples to evaluate individuals (candidate decision rules) from the genetic population generated by MECoMaP. The time of the operations on an AVL tree is O(log n) average, where n is the number of elements. Each node determines a condition of a property and each leaf represents a list with the training examples that fulfills all the conditions impose in the predecesor nodes. Each level of the tree represents a determined property of a determined position of an amino acid. We consider a tree example in figure 2. Level 1 represents the hydrophobicity of amino acid i and level 2 indicates the polarity of amino acid j. As example, leaf node $N1$ stores all the training examples whose amino acid in position i has a hydrophobicity value lower than 0 and a polarity value in position j is also lower than 0. We achieve a reduction of the computational cost about 50% by means of a fast evaluation of examples from the dataset.

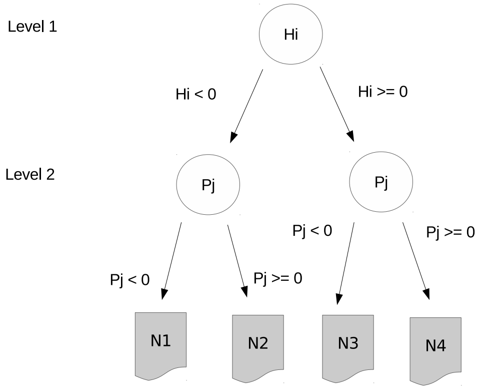

Fig. 2. Example of efficient evaluation structure (AVL tree)

3 Experiments and Results

We have built a file in arff format with all the training data information. This file is constituted by all the protein subsequences of two windows of seven amino acids encoded with the values of the cited attributes. The positive class (contact) is represented with 1 and the negative class (non-contact) is represented with 0. The ratio between the positive and negative classes was set to 2:1 for DS1 and DS2 data sets. The training data used contained all the possible subsequences with a minimum separation between contact residues of 7 amino acids for DS1 and a separation 6 amino acids for DS2. We have performed several experiments with three Weka classifiers [13]: Näive Bayes (NB), C4.5 classifier tree (J48) and Nearest Neighbor approach with $k = 1$ (IB1). The obtained results can be seen in Table 2 for a 3-fold cross-validation. We appreciate low coverage and accuracy values in all the cases. This experiment was performed with the aim of validating our representation and confirms that the new encoding provides enough information for a good performance of a learning classifier. Moreover, we can also notice that MECoMaP achieved the best results for this experiment and improve the results for DS1 and DS2 data set shown in [4].

Table 2. Average results obtained for MECoMaP and different classification Weka algorithms for the DS1 and DS2 protein data set

Algorithm	Data Set	$Coverage_{\mu\pm\sigma}$	$Accuracy_{\mu\pm\sigma}$
J48	DS1	$0.04_{\pm0.07}$	$0.19_{\pm0.08}$
IB1	DS1	$0.08_{\pm0.05}$	$0.07_{\pm0.05}$
NB	DS1	$0.15_{\pm0.03}$	$0.08_{\pm0.02}$
MECoMaP	DS1	$0.18_{\pm0.13}$	$0.26_{\pm0.32}$
MECoMaP 2.0	DS1	$0.20_{\pm0.15}$	$0.29_{\pm0.11}$
J48	DS2	$0.10_{\pm0.02}$	$0.10_{\pm0.05}$
IB1	DS2	$0.07_{\pm0.10}$	$0.07_{\pm0.05}$
NB	DS2	$0.10_{\pm0.10}$	$0.18_{\pm0.10}$
MECoMaP	DS2	$0.12_{\pm0.01}$	$0.38_{\pm0.09}$
MECoMaP 2.0	DS2	$0.18_{\pm0.08}$	$0.39_{\pm0.07}$

4 Conclusions and Future Work

In this work, we presented some improvements to a multi-objective optimization algorithm for the residue-residue contact prediction. Two of these improvements enhance the efficiency of the algorithm: the introduction of new features based on evolutionary information (PSSM) for the encoding and a treatment for the unbalanced classes. An efficient evaluation structure for a fast evaluation of the training data is also included to reduce the time complexity of the EA. This algorithm generates rules that predict the necessary conditions for the contact between two amino acids based on their physico-chemical properties. The algorithm was tested on two sets of proteins that had been previously used in the literature and achieved better coverage and accuracy rates than the predecessor version of the algorithm. As future work, the incorporation of new evolutionary information such as correlated mutations must be taken into account. Furthermore, our algorithm must be validated with a higher number of proteins data set.

References

1. Tegge, A., Wang, Z., Eickholt, J., Cheng, J.: Nncon: Improved protein contact map prediction using 2d-recursive neural networks. Nucleic Acids Research 37(2), 515–518 (2009)

2. Jones, D.T., Buchan, D.W., Cozzetto, D., Pontil, M.: Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics 28(2), 184–190 (2012)
3. Calvo, J.C., Ortega, J.: Parallel protein structure prediction by multiobjective optimization. Parallel, Distributed and Network-based Processing 12(4), 407–413 (2009)
4. Marquez-Chamorro, A.E., Asencio, G., Divina, F., Aguilar-Ruiz, J.S.: Evolutionary decision rules for predicting protein contact maps. Pattern Analysis and Applications, PAAA (September 1-13, 2012)
5. Russell, R.B., Betts, M.J., Barnes, M.R.: Amino acid properties and consequences of subsitutions. In: Bioinformatics for Geneticists. Wiley (2003)
6. Fariselli, P., Olmea, O., Valencia, A., Casadio, R.: Prediction of contact map with neural networks and correlated mutations. Protein Engineering 14, 133–154 (2001)
7. Cheng, J., Baldi, P.: Improved residue contact prediction using support vector machines and a large feature set. Bioinformatics 8, 113 (2007)
8. Gupta, N., Mangal, N., Biswas, S.: Evolution and similarity evaluation of protein structures in contact map space. Proteins: Structure, Function, and Bioinformatics 59, 196–204 (2005)
9. Kyte, J., Doolittle, R.F.: A simple method for displaying the hydropathic character of a protein. J. J. Mol. Bio. 157, 105–132 (1982)
10. Grantham, R.: Amino acid difference formula to help explain protein evolution. J. J. Mol. Bio. 185, 862–864 (1974)
11. Klein, P., Kanehisa, M., DeLisi, C.: Prediction of protein function from sequence properties: Discriminant analysis of a data base. Bioch. Bioph. 787, 221–226 (1984)
12. Adelson-Velskii, G., Landis, E.M.: An algorithm for the organization of information. Proceedings of the USSR Academy of Sciences; Soviet Math. 3, 1259–1263
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The weka data mining software: An update. SIGKDD Explorations 11 (2009)