An efficient decision rule-based system for the protein residue-residue contact prediction

Alfonso E. Márquez-Chamorro
School of Engineering
Pablo de Olavide University
Seville, Spain
amarcha@upo.es

Federico Divina
School of Engineering
Pablo de Olavide University
Seville, Spain
fdiv@upo.es

Jaume Bacardit
School of Computer Science
University of Nottingham
Nottingham, UK
jqb@cs.nott.ac.uk

Jesús S. Aguilar-Ruiz
School of Engineering
Pablo de Olavide University
Seville, Spain
aguilar@upo.es

***Abstract*—Protein structure prediction remains one of the most important challenges in molecular biology. Contact maps have been extensively used as a simplified representation of protein structures. In this work, we propose a multi-objective evolutionary approach for contact map prediction. The proposed method bases the prediction on a set of physico-chemical properties and structural features of the amino acids, as well as evolutionary information in the form of an amino acid position specific scoring matrix (PSSM). The proposed technique produces a set of decision rules that identify contacts between amino acids. Results obtained by our approach are presented and confirm the validity of our proposal.**

I. Introduction

The prediction of the three-dimensional structure of a protein from its sequence of amino acids is one of the main open problems in structural bioinformatics. This problem is known as Protein Structure Prediction (PSP). The specific biochemical function of a protein is determined by its structure complexity, which, in turn, is determined by the specific sequence of amino acids. Therefore, solving this problem would allow to have knowledge of the protein function directly from its amino acids sequence. This would have great importance in different areas, *e.g.*, drug design or protein engineering. More and more amino acids sequences of proteins are available by the day, but their three-dimensional structures remain often unknown, and so their functions cannot be determined. Consequently the gap between protein sequence information and protein structural information is rapidly increasing. It follows that computational methods are needed in order to reduce this gap, as they would provide an inexpensive and fast way to solve the PSP problem. Soft computing techniques are particularly suited for this problem, since they rely on inexact solutions in order to deal with with complex, incomplete, uncertain and inaccurate data. Different soft computing approaches have been developed to solve the PSP problem. Among the most popular soft computing paradigms, there are artificial neural networks (ANNs), evolutionary computation (EC) and support vector machines (SVMs).

Contact maps are a simplified represention of the protein tertiary structure prediction. They are focused on determining contacts between the residues of a protein sequence. When a contact map is defined, proteins can be fold and 3D structure can be determined. Several contact map prediction methods have been applied to the PSP problem, *e.g.*, ANNs [1], SVMs [2], EC [3] and template-based modelling [4]. Evolutionary algorithms (EAs) are well suited for solving the PSP problem, since PSP can be seen as a search problem through the space determined by all the possible foldings. Such a space is highly complex and has a huge size. Finding the optimal solution in such space is very hard. In these cases, EAs have proven to be effective methods that can provide sub-optimal solutions.

In this work, we propose an EA in order to predict contact maps. In particular, the proposed method will provide a prediction model based on decision rules that express conditions on a set of biochemical properties and 1D structural features of the amino acids, as well as evolutionary information. Such a model can then be used in order to determine whether or not there is a contact between two amino acids. An advantage of such an approach is that the generated rules can be easily interpreted by experts in the field in order to extract further insight of the folding process of proteins. The main feature of our proposal is that the prediction is based on a set of amino acid properties, which are very important in the folding process [5]. 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 [3]. To the best of our knowledge, no other EA considers amino acid properties for the prediction.

The rest of paper is organized as follow: in section II, we discuss our proposal to predict protein contact maps, while in section III provides the experimentation and the obtained results. Finally, we draw some conclusions and discuss future works.

II. Methodology

This section describes our proposal for contact maps prediction. A contact map is a binary version of the distance matrix defined as a square symmetric matrix of order L, where L is the number of amino acids in the sequence. A contact map is divided into two parts: the observed part (upper triangular) and the predicted part (lower triangular). An element (i, j) of the contact map is 1 if amino acids i and j are in contact, and 0 otherwise. In this context, we consider two amino acids to be in contact if the distance between them is less than or equal to a given threshold. To this aim, a commonly used threshold is 8 angstroms (Å) [6].

Our proposal for the prediction of contact maps consists of a multi-objective EA (MOEA), based on the Strength Pareto Evolutionary Algorithm (SPEA) [7]. This algorithm uses an external population of non-dominated solutions, which is obtained at the end of every generation. The algorithm is based on the strength concept, where the strength of an individual x is given by the number of individuals that x dominates. The fitness of an individual is proportional to its strength, as will be detailed in the following. Each individual of the population represents a decision rule. In particular, rules are based on three physico-chemical properties of amino acids, i.e., hydrophobicity (H), polarity (P) and charge (C), other structural features of amino acids, i.e., solvent accessibility (SA) and secondary structure (SS) and evolutionary information (PSSM values). Rules specify a set of conditions on each property, that, if satisfied, predict a contact between two amino acids.

In the following sections, we present different details of the cited properties and features of amino acids, the enconding of the algorithm, as well as the fitness function and genetic operators used. We will also discuss an efficient structure for the evaluation of the individuals.

A. Physico-chemical properties of the amino acids

The most direct information we can extract from the primary sequence of a protein are physico-chemical characteristics of its residues (in this case hydrophobicity, polarity and net charge). With this information, we can generate representations of, for instance, how the hydrophobicity varies along the sequence of the protein and obtain information about hydrophobic areas, which may help in the prediction of structural characteristics. 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 [8].

By definition, a substance is hydrophobic if it is not miscible with water. Hydrophobicity is then defined as the incapacity of interacting with the molecules of water by ion-dipole interactions or by hydrogen bonds. In chemistry, polarity refers to a separation of electric charge leading to a molecule or its chemical groups having an electric dipole or multipole moment. The net charge is the algebraic sum of all the charged groups present in any amino acid, peptide or protein.

In our proposal, we use the Kyte-Doolittle hydropathy profile [9] for hydrophobicity, the Grantham's profile [10] for polarity and Klein's scale for net charge [11]. Table I reports the values of these properties for each amino acid, according to the cited scales and normalized between -1 and 1 for hydrophobicity and polarity. For the hydrophobicity and polarity values, the higher the value, the more hydrophobic or polar is the amino acid. For the net charge, a positive, negative or neutral charge, are represented with a 1, -1 or 0, respectively. We can see, for example that amino acid I presents a hydrophobicity value equal to 1.0, which means that I is highly hydrophobic, a polarity of -0.93, which means that I is poorly polar. Moreover we can see that this amino acid possesses a neutral charge.

In addition to these properties, we also use two structural features of proteins (SS and SA) and evolutionary information, which are discussed in the next section.

TABLE I. VALUES OF DIFFERENT PROPERTIES ACCORDING TO THE CITED SCALES FOR EACH AMINO ACID. H REPRESENTS THE HYDROPHOBICITY, P THE POLARITY AND C THE CHARGE.

Prop.	A	C	D	E	F	G	H	I	K	L
H	0.40	0.56	-0.78	-0.78	0.62	-0.09	-0.71	1.00	-0.87	0.84
P	-0.21	-0.85	1.00	0.83	-0.93	0.01	0.36	-0.93	0.58	-1.00
C	0	0	-1	-1	0	0	0	0	1	0

Prop.	M	N	P	Q	R	S	T	V	W	Y
H	0.42	-0.78	-0.36	-0.78	-1.00	-0.18	-0.16	0.93	-0.20	-0.30
P	-0.80	0.65	-0.23	0.38	0.38	0.06	-0.09	-0.75	-0.88	-0.68
C	0	0	0	0	1	0	0	0	0	0

B. Structural features of protein residues

As mentioned before, our proposal relies also on structural features of protein residues. In particular, it uses secondary structures and solvent accessibility. 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. A 3-state representation of SS (helix, sheet or coil) is employed in our approach. The prediction is performed using PSIPRED [12].

SA refers to the degree to which a residue interacts with the solvent molecules. The SA of amino acid residues provides us with useful information for the prediction of the structure and function of a protein. Relative solvent accessibility (RSA) is required for the prediction. To calculate the RSA of a residue, we use the DSSP program [13], and then obtain the actual SA of each residue as described in [14]. SA is divided by the maximum accessible surface in the extended conformation of its AA type. We finally obtain a 5-state representation (ranging from 0 to 4) for SA, where lower values mean a buried state and higher values represent exposed states. 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*.

C. Evolutionary information

Evolutionary information is also used by the algorithm in order to predict contacts. We have included in our representation the evolutionary information obtained from PSI-BLAST [15] using non-redundant protein sequences database. Sequence alignment is a standard technique in bioinformatics for visualizing the relationships between residues in a collection of evolutionary or structurally related protein. Position-specific scoring matrices (PSSM) are obtained from sequence alignments. A PSSM determines the substitution scores between amino acids according to their positions in the alignment. Each cell of the matrix is calculated as the log_2 of the observed substitution frequency at a given position divided by the expected substitution frequency at that position. Thus, a positive score (ratio $>$ 1) indicates that the observed frequency exceeds the expected frequency, suggesting that this substitution is surprisingly favored. A negative score (ratio $<$ 1) indicates the opposite, i.e., the observed substitution frequency is lower than the expected frequency, suggesting that the substitution is not favored. We normalize the PSSM values between -1 and 1.

D. Encoding

Each individual in our algorithm represents a decision rule which determines whether two amino acids i and j are in contact, with $1 \leq i < j \leq L$. For this purpose, 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 $j-3, j-2, j-1, j, j+1, j+2, j+3$. For each amino acid k belonging to the two windows, we define the descriptor Q_k (where $k \in \{i-3, i-2, i-1, i, i+1, i+2, i+3, j-3, j-2, j-1, j, j+1, j+2, j+3\}$) which represents a set of conditions for the amino acid k, as shown in equation 1.

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

We define the decision rule $R_{i,j}$ for amino acids i and j, encoded in each individual of our algorithm, as in equation 2, for each $k \in \{i-3, i-2, i-1, i, i+1, i+2, i+3, j-3, j-2, j-1, j, j+1, j+2, j+3\}$.

$$R_{i,j} = \{Q_k\} \tag{2}$$

A rule covers the pair i, j if that pair satisfies Q_k for all k (where $k \in \{i-3, i-2, i-1, i, i+1, i+2, i+3, j-3, j-2, j-1, j, j+1, j+2, j+3\}$). The pair i, j satisfies Q_k if it fulfills the following equations for all k:

$$\begin{gathered} H_{min} \leq H(k) \leq H_{max} \\ P_{min} \leq P(k) \leq P_{max} \\ PSSM_{min}^{1..20} \leq PSSM(k) \leq PSSM_{max}^{1..20} \\ C(k) = C \\ SS(k) = SS \\ SA(k) = SA \end{gathered} \tag{3}$$

where $H(k)$ is the hydrophobicity of the amino acid k, $P(k)$ its polarity, $C(k)$ its charge, $SS(k)$ its secondary structure, $SA(k)$ its solvent accessibility and $PSSM(k)$ represent the PSSM value for each residue k and for the 20 different amino acids.

An example of encoding for the element Q_i of an individual $R_{i,j}$ is shown in figure 1.

$$\underbrace{\begin{array}{|c|c|c|c|c|c|c|c|c|}\hline H_1 & H_2 & P_1 & P_2 & C & SS & SA & PSSM_1^k & PSSM_2^k \\ \hline\end{array}}_{i}$$

Fig. 1. Example of encoding for the element Q_i of an individual $R_{i,j}$. H_1, H_2, P_1 and P_2 are lower and upper bounds for the hydrophobicity and polarity and volume values. C represents the charge value of the residue. SS represents secondary structure and SA indicates the solvent accessibility value. Finally, $PSSM_1$ and $PSSM_2$ represent the lower and upper bounds for the position-specific scoring matrix values for each k amino acid.

E. Fitness Function

The objective of the algorithm is to find both general and precise rules for identifying residue-residue contacts. To this aim, we consider two objectives to be optimized: rule coverage and rule accuracy. Rule coverage represents the proportion of contacts covered by each rule, while rule accuracy evaluates the correctly predicted contacts rate by each rule. Therefore, $Rule\ coverage = \frac{C}{C_t}$ and $Rule\ accuracy = \frac{C}{C_p}$, 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. The fitness of an individual is equal to the number of individuals that it dominates. The formula for the fitness function for an individual i is detailed in equation 4, where j represents an individual of the population and N represents the size of the population.

$$fitness(i) = \sum_{j}^{N} f(i,j) \tag{4}$$

Function $f(i,j)$ is defined in equation 5 and determines if individual i dominates individual j according to the two objectives used, where $i \succeq j$ *iff* $o_t(i) \geq o_t(j)$ for all $t = 1..M$, and $o_t(i) > o_t(j)$ for some $t \in 1..M$, where o_t are the objectives and M=2.

$$f(i,j) = \begin{cases} 1\ if\ i \succeq j \\ 0\ otherwise \end{cases} \tag{5}$$

F. Genetic Operators

We have applied a one-point crossover operator. This crossover operator acts at the level of the amino acid properties. This crossover operator randomly selects a point inside two parents and then builds the offspring using one part of each parent. It follows that the resulting rule has to be tested for validity, since it could contain incorrect ranges. An example of one-point crossover is shown in figure 2.

$$\begin{array}{rl} Par.\ 1 & \underbrace{\begin{array}{|c|c|c|c|c|c|c|}\hline \mathbf{0.19} & \mathbf{0.39} & -0.78 & -0.68 & 0.00 & 0.83 & 1.03 \\ \hline\end{array}} \\ Par.\ 2 & \underbrace{\begin{array}{|c|c|c|c|c|c|c|}\hline 0.52 & 0.92 & \mathbf{-1.00} & \mathbf{-0.93} & \mathbf{0.00} & \mathbf{0.77} & \mathbf{0.97} \\ \hline\end{array}} \\ & \Downarrow \\ Off.\ 1 & \begin{array}{|c|c|c|c|c|c|c|}\hline 0.19 & 0.39 & -1.00 & -0.93 & 0.00 & 0.77 & 0.97 \\ \hline\end{array} \end{array}$$

Fig. 2. An example of one-point crossover for the element Q_i of two parent individuals $Par.1$ and $Par.2$ and the offspring individual $Off.1$. The random cut is established between H_2 and P_1.

The algorithm uses two different mutation operators. The first operator, called Gaussian operator, mutates a randomly selected element of Q_k of an individual $R_{i,j}$, where $k \in \{i-3, i-2, i-1, i, i+1, i+2, i+3, j-3, j-2, j-1, j, j+1, j+2, j+3\}$. The operator can either increase or decrease (with

0.5 probability) the value of the selected element by an amount which is randomly chosen between the allowed ranges for each properties following a Gaussian distribution. If the values of a mutated individual are not within the allowed ranges for each properties, the mutation is discarded. An example of this mutation operator is shown in figure 3.

$$\underbrace{\begin{array}{|c|c|c|c|c|c|c|}\hline 0.52 & \mathbf{0.75} & 0.15 & 0.60 & 0 & 0 & 3 \\ \hline\end{array}}$$
$$\Downarrow$$
$$\begin{array}{|c|c|c|c|c|c|c|}\hline 0.52 & \mathbf{0.95} & 0.15 & 0.60 & 0 & 0 & 3 \\ \hline\end{array}$$

Fig. 3. Example of Gaussian mutation for the element Q_i of an individual $R_{i,j}$ with an increment value of $+0.2$ for the H_2 property.

A second mutation operator, called Enlarge operator, randomly selects an element of Q_k of an individual, that is related to a given property, and enlarges its range to its maximum. For instance, if the property is the hydrophobicity, this operator varies the range to $[-1, 1]$. This means that the rule does not take into account this property in this case. An example of this mutation operator is represented in figure 4.

$$\underbrace{\begin{array}{|c|c|c|c|c|c|c|}\hline 0.1 & 0.5 & \mathbf{0.38} & \mathbf{0.50} & 1 & 1 & 4 \\ \hline\end{array}}$$
$$\Downarrow$$
$$\begin{array}{|c|c|c|c|c|c|c|}\hline 0.1 & 0.5 & \mathbf{-1.00} & \mathbf{1.00} & 1 & 1 & 4 \\ \hline\end{array}$$

Fig. 4. Example of enlarge mutation operator for the element Q_i of an individual $R_{i,j}$ for P_1 and P_2 properties.

The parameter setting of the algorithm used in this paper is shown in table II. This setting was determined after several preliminary trial runs.

TABLE II. PARAMETER SETTING USED IN THE EXPERIMENTS.

Population size	100
Crossover probability	0.5
Gaussian mutation probability	0.5
Enlarge mutation probability	0.1
Max number of generations	100
Tournament size	2

G. The Algorithm

The pseudo code of the algorithm is shown in Algorithm 1. It can be seen that, the EA is run $numIt$ times where $numIt$ is the number of iterations. Thus, the final set of rules ($Results$ in the code) is built in an incremental way. At the end of each run of the EA, the algorithm adds to the final set the best rules found by the EA. This is done in the following way: first, the best individual, according to its $F-measure$, is selected and added to the final solution. Then the next best individual is added, and the global $F-measure$ of the final solution is calculated. This process is repeated until the addition of a rule causes the $F-measure$ of the final solution to decrease. The $F-measure$ is defined as in equation 6:

$$Fmeasure = 2 \cdot \frac{Rule\ coverage \cdot Rule\ accuracy}{Rule\ coverage + Rule\ accuracy} \tag{6}$$

Repeated or redundant rules are not included in the final solution. Each pair of rules $(R_{a,b}, R_{c,d})$ is checked, where a, b and c, d are two pairs of amino acids in contact. If we find that $R_{a,b}$ is contained in $R_{c,d}$, then $R_{a,b}$ is removed from our final rule set. In this context, a rule $R_{a,b}$ is contained in another rule $R_{c,d}$ if the values of the elements of Q_k (for each $k \in [a-3, a+3] \cup [b-3, b+3]$) and the values of the elements of $Q_{k'}$ (for each $k' \in [c-3, c+3] \cup [d-3, d+3]$) satisfy the conditions shown in equation 7.

$$\begin{gathered} H_{min} \geq H'_{min} \wedge H_{max} \leq H'_{max} \\ P_{min} \geq P'_{min} \wedge P_{max} \leq P'_{max} \\ PSSM^{1..20}_{min} \geq PSSM'^{1..20}_{min} \wedge \\ PSSM^{1..20}_{max} \geq PSSM'^{1..20}_{max} \\ C = C' \\ SS = SS' \\ SA = SA' \end{gathered} \tag{7}$$

Algorithm 1 ALGORITHM FOR CONTACT MAP PREDICTION

INPUT set of protein subsequences M, maximum number of iterations $numIt$, maximum number of generations $maxGen$.
OUTPUT set of generated rules $Results$.

```
begin
  $num \leftarrow 0$, $Results \leftarrow \emptyset$
  while ($num < numIt$) do
    Initialize $P$
    $i \leftarrow 0$, $A \leftarrow \emptyset$
    while ($i < maxGen$) do
      Evaluate population $P$
      Find Non-dominated solutions $P$
      Update Non-dominated solutions set $A$
      $P' \leftarrow A$
      $P' \leftarrow$ Selection Method with binary tournament($P$)
      $P' \leftarrow$ 2-point Crossover Method with binary tournament($P$)
      $P' \leftarrow$ Mutation Method($P$)
      $P \leftarrow P'$
      $i \leftarrow i + 1$
    end while
    $Results \leftarrow$ the best combination of rules from $P$
    $num \leftarrow num + 1$
  end while
  return $Results$
end
```

The EA starts by randomly initialize the population. Then, it evaluates the current population P and the Pareto front is determined. Non-dominated solutions, which constitute the external population A, will be included in the population P' of the next generation. As already mentioned, four genetic operators are used: a binary tournament selection operator, a 2-point crossover operator and two mutation operators. The first 50% of the individuals in P' consist of the non-dominated individuals (external population A) and by the selected individuals with the binary tournament selection operator. The rest of the individuals in P' are generated using the 2-point crossover operator. Mutation is applied to the whole population, except to the Pareto front individuals, at the end of each generation. This process is repeated a maximum number of generations $maxGen$.

H. Efficient evaluation of the individuals

In order to reduce the computational time of our method, we have implemented an AVL tree [16] to order and classify the training examples according to their property values. A binary search tree is a binary tree with the property that all the elements stored in the left subtree of any node x are less than or equal to the item stored in x, and all the items stored in the right subtree of x are higher than the element stored in x. An AVL tree is a self-balancing binary search tree where the heights of the two child subtrees of any node differ by at most one. The time of the operations on an AVL tree is $O(logn)$ on average, where n is the number of elements. Each node determines a condition of a property (e.g. hydrophobicity of amino acid i < 0) and each leave represents a list with the training examples that fulfills all the conditions imposed in the predecessor nodes. Each level of the tree represents a determined property of a determined position of an amino acid.

The main goal of the implementation of this structure is the reduction of time complexity of the algorithm by means of a fast evaluation of examples from the dataset. 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. This structure is also similar to the Efficient Evaluation Structure (EES) described in [17].

III. Experiments and results

In this section we will present the results obtained by our algorithm. The algorithm is implemented in Java using a multithreading architecture. Furthermore, due to the enormous volume of data, all the experiments were run on a 64-bit workstation, with 32 GB DDR SDRAM and four dual-core processors. We have performed experiments on a protein data set [18], called DS1, that consists of 173 non-redundant proteins with sequence identity lower than 25%. As in [18], four subsets have been obtained according to the sequence length (L_s): $L_s < 100, 100 <= L_s < 170, 170 <= L_s < 300$, $L_s >= 300$. The minimum and maximum lengths of proteins are 31 and 753 amino acids, respectively. DS1 contains $240,501$ positive examples (contacts) and $5,034,050$ negative examples (non-contacts). In order to calculate the distances between residues of the training set, we select the respective beta carbon atom (C_β) of each amino acid, and use the Euclidean distance.

In the following experiments we employ three statistical measures which are used in CASP competitions [19], *i.e.*, coverage, accuracy and X_d, to evaluate the performance of our protein structure predictors. In particular, in CASP, coverage indicates what percentage of contacts have been correctly identified. Accuracy reflects the number of correctly predicted contacts. X_d represents the distribution accuracy of the predicted contacts, and is defined by equation 8

$$X_d = \sum_{i=1}^{15} \frac{P_i - P_a}{i} \tag{8}$$

where P_i represents the percentage of predicted pairs with a distance between $4(i-1)$ and $4i$ and P_a represents the percentage of total pairs with a distance between $4(i-1)$ and $4i$.

Before presenting the results obtained by our algorithm, we present the results of a preliminary study to justify the use of evolutionary information. The vast majority of methods in PSP use evolutionary information, such as multiple sequence alignments or PSSM matrices, *e.g.* [20], [21], [22]. The use of this information, can help obtaining a significant increment of the accuracy in the predictions. To confirm this observation, we have performed an experimentation using WEKA [23]. We have analyzed the effect of using different attributes for the same DS1 training examples. The first training dataset only includes physico-chemical attributes: H, P and C. A second dataset includes the cited physico-chemical and two attributes related to structural features: SS and SA. The third dataset also includes all the previous attributes and PSSM attributes.

The results are shown in table III. For the experimentation, we have used IB1 WEKA classifier. We can notice an improvement of the results when PSSM attributes are used, with an increment of 5.4 percentage points in accuracy regarding the second dataset and 7 percentage points regarding the first dataset. We can also notice that the algorithm obtains better results with the second set of attributes than with the first set. Motivated by these results, we have included PSSM information in the encoding of our algorithm.

TABLE III. Efficiency of Ib1 WEKA classifier predicting DS1 protein dataset with different sets of attributes.

Method	Attributes	Acc.$_{\mu\pm\sigma}$	Cov.$_{\mu\pm\sigma}$
IB1	H,P,C	0.087$_{\pm0.05}$	0.084$_{\pm0.08}$
IB1	H,P,C,SS,SA	0.115$_{\pm0.07}$	0.112$_{\pm0.09}$
IB1	H,P,C,SS,SA,PSSM	0.169$_{\pm0.08}$	0.164$_{\pm0.12}$

Our experimentation was performed under the same conditions that appeared in [18]. A threshold of 8 angstroms (Å) was established to determine a contact. In order to avoid the effect of learning local contacts, we set the same minimum sequence separation (7 residues) between each pair of amino acids to establish a contact as in the reference work.

Since the algorithm incrementally adds decision rules to a final set of rules, and since the optimal and exact number of rules is unknown, we have performed various experiments varying the numbers of runs of the EA, where to a higher number of runs corresponds a higher number of rules in the final set. The aim of this was to test whether or not a higher number of rules would yield better results. From these, we have concluded that the best results were obtained when the algorithm was run for 1,000 iterations.

Table IV shows the results obtained on DS1. As in [18], we used a 3-fold cross-validation. We have compared our results with the ones reported in [18] using the same data set. The first column of the table reports the sequence length range of each subset of proteins, while the second column represents the number of proteins of each subset. The third column shows the average accuracy rate obtained by our proposal, and finally, the fourth column presents the average accuracy rate obtained by the reference algorithm [18]. Standard deviation for accuracy is also reported. We can notice how the accuracy rate decreases when the length of the sequences increases. It can be noticed that our algorithm obtains better results than [18] in all the cases. Low values of standard deviation show us that our data results are not significantly spread compared to the results obtained by [18]. Positive values of X_d are achieved in all the cases. Therefore our predictor improves the performance of a random predictor (negative values of X_d).

TABLE IV. Efficiency of our method predicting DS1 protein data set.

		Our proposal	[18]
Protein length	#prot.	Acc.$_{\mu\pm\sigma}$	Acc.$_{\mu\pm\sigma}$
$L \leq 100$	65	0.61$_{\pm 0.25}$	0.26$_{\pm 0.39}$
$100 \leq L < 170$	57	0.25$_{\pm 0.15}$	0.21$_{\pm 0.32}$
$170 \leq L < 300$	30	0.20$_{\pm 0.11}$	0.15$_{\pm 0.22}$
$L \geq 300$	21	0.13$_{\pm 0.10}$	0.11$_{\pm 0.15}$
All proteins	173	0.29$_{\pm 0.15}$	0.18$_{\pm 0.32}$

The following is an example of a rule $R_{i,j}$ generated by our algorithm:

$$\begin{aligned} &if\ H_i \in [0.52, 0.92]\ and\ P_i \in [-1.00, -0.93]\ and \\ &C_i = 0\ and\ SS_i = 2\ and\ SA_i = 0\ and \\ &H_j \in [0.32, 0.82]\ and\ P_{j+1} \in [-0.41, -0.01]\ and \\ &C_{j+1} = 0\ and\ SS_{j+1} = 2\ and\ SA_{j+2} = 1\ and \\ &PSSM_i^K \in [-0.97, -0.23]\ then\ contact \end{aligned} \tag{9}$$

If we inspect this rule, we can notice that it indicates that if the hydrophobicity of amino acid in position i is between 0.52 and 0.92, the SA value of amino acid in position $j + 2$ is equal to 1 and PSSM value in position i for amino acid K is between -0.97 and -0.23 among other requirements, a contact is established. We can notice that rules generated by our proposal are characterized by a high interpretability, which would allow experts in the field to easily inspect the results for further validation.

IV. Conclusion

We have implemented an efficient decision rule-based system for the protein residue-residue contact prediction. This system is based on a multi-objective evolutionary algorithm. The specified decision rules are based on biochemical properties (H, P and C) and 1D structural features (SS and SA) of the amino acids, as well as evolutionary information in form of PSSM values. The rules can be interpreted by experts in the field. Moreover, an efficient structure included in the algorithm achieves a reduction of time complexity of the application by means of a fast evaluation of examples from the dataset. As for future work, our algorithm must be tested with higher and novel protein data sets.

Acknowledgment

This research was supported by the Junta de Andalucia, Project of Excellence P07-TIC-02611 and by the Spanish MEC under project TIN2011-28956-C02-01.

References

[1] AN. Tegge, Z. Wang, J. Eickholt and J. Cheng, *NNcon: Improved Protein Contact Map Prediction Using 2D-Recursive Neural Networks*, Nucleic Acids Research, 37(2), 515-518, 2009.

[2] J. Cheng and P. Baldi, *Improved residue contact prediction using support vector machines and a large feature set*, Bioinformatics, 8, 113, 2007.

[3] N. Gupta, N. Mangal and S. Biswas, *Evolution and Similarity Evaluation of Protein Structures in Contact Map Space*, Proteins: Structure, Function, and Bioinformatics, 59, 196-204, 2005.

[4] Y. Zhang, *I-TASSER: fully automated protein structure prediction in CASP8, Proteins: Structure, Function, and Bioinformatics*, 77, 100-113, 2009.

[5] J. Gu and PE. Bourne, *Structural Bioinformatics (Methods of Biochemical Analysis)*, Wiley-Blackwell, 2003.

[6] B. Monastyrskyy, K. Fidelis, A. Tramontano and A. Kryshtafovych, *Evaluation of residue-residue contact predictions in casp9*, Proteins, 79 Suppl 10, 119–125, 2011.

[7] E. Zitzler and L. Thiele, *An evolutionary algorithm for multiobjective optimization: The strength Pareto Approach*, Technical Report 43, Zürich,Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), 1998.

[8] R. Unger and J. Moult, *Genetic algorithms for protein folding simulations*, J. Mol. Biol., 231, 75-81, 1993.

[9] J. Kyte and R. Doolittle, *A simple method for displaying the hydropathic character of a protein*, J. J. Mol. Bio. 157, 105–132, 1982.

[10] R. Grantham, *Amino acid difference formula to help explain protein evolution*, J. J. Mol. Bio., 185, 862–864, 1974.

[11] P. Klein, M, Kanehisa and C. De Lisi, *Prediction of protein function from sequence properties: Discriminant analysis of a data base*, Biochim. Biophys., 787, 221–226, 1984.

[12] D. Jones, *Protein secondary structure prediction based on position-specific scoring matrices*, Journal of Molecular Biology, 292, 195–202, 1999.

[13] W. Kabsch and C. Sander, *Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features*, Biopolymers, 22(12), p. 2577-2637, 1983.

[14] M. Stout, J. Bacardit, J. Hirst and N. Krasnogor, *Prediction of recursive convex hull class assignments for protein residues*, Bioinformatics, 24(7), p. 916-923, 2009.

[15] SF. Altschul, TL. Madden, AA. Schäffer, J. Zhang, Z. Zhang, W. Miller and DJ. Lipman, *Gapped blast and psi-blast: a new generation of protein database search programs*, Nucleic Acids Res 25(17), 3389–3402, 1997.

[16] G. Adelson-Velskii and EM. Landis, *An algorithm for the organization of information*, Proceedings of the USSR Academy of Sciences, 146, p. 263–266 (Russian). English translation: Ricci, M.J. in Soviet Math, 3, p. 1259–1263, 1962.

[17] R. Giraldez, JS. Aguilar-Ruiz and JC. Riquelme, *Knowledge-based Fast Evaluation for Evolutionary Learning*, IEEE Transactions on Systems, Man and Cybernetics, Part C, Vol 35(2) p. 254-261, 2005.

[18] P. Fariselli, O. Olmea, A. Valencia and R. Casadio, *Prediction of contact map with neural networks and correlated mutations*, Protein Engineering 14, 133–154, 2001.

[19] B. Monastyrskyy, K. Fidelis, A. Tramontano and A. Kryshtafovych, *Evaluation of residue-residue contact predictions in casp9*, Proteins: Structure, Function, and Bioinformatics 79(S10), 119–125, 2011.

[20] P. Fariselli and R. Casadio, *A neural network based predictor of residue contacts in proteins*, Protein Engineering, 12, 15-21, 1999.

[21] Y. Zhao and G. Karypis, *Prediction of Contact Maps Using Support Vector Machines*, Proceedings of Third IEEE Symposium on Bioinformatics and Bioengineering, BIBE 2003, 26-33, 2003.

[22] B. Xue, E. Faraggi and Y. Zhou, *Predicting residue-residue contact maps by a two-layer, integrated neural-network method*, Proteins, 76(1), 176–183, 2009.

[23] M. Hall, E. Frank, G. Holmes, P. Reutemann and I. Witten, *The weka data mining software: An update.*, SIGKDD Explorations 11(1), 10-18, 2009.