Evolutionary decision rules for predicting protein contact maps

Alfonso Eduardo Márquez-Chamorro,
Gualberto Asencio-Cortés,
Federico Divina,
Jesus Salvador Aguilar-Ruiz

Abstract Protein structure prediction is currently one of the main open challenges in Bioinformatics. The protein contact map is an useful, and commonly used, representation for protein 3D structure and represents binary proximities (contact or non-contact) between each pair of amino acids of a protein. In this work, we propose a multi-objective evolutionary approach for contact map prediction based on physico-chemical properties of amino acids. The evolutionary algorithm produces a set of decision rules that identifies contacts between amino acids. The rules obtained by the algorithm impose a set of conditions based on amino acid properties to predict contacts. We present results obtained by our approach on four different protein data sets. A statistical study was also performed to extract valid conclusions from the set of prediction rules generated by our algorithm. Results obtained confirm the validity of our proposal.

Keywords Protein structure prediction · Contact map · Multi-objective evolutionary computation · Residue–residue contact

1 Introduction

The Protein Structure Prediction (PSP) problem consists in determining the three-dimensional model of a protein, using only information contained in its amino acid sequence. The PSP problem is one of the most important open problems in computational biology [66]. This is because the 3D structures determine the protein function. It follows that knowing the 3D structure of a protein would be of enormous help for designing new drugs for diseases such as cancer or Alzheimer's. Although there exist experimental methods for determining protein structures, e.g., X-ray crystallography and nuclear magnetic resonance, such techniques are very expensive and present limitations with the structures of certain proteins [27, 38, 51]. In addition to this, the great number of protein sequences whose three-dimensional structures must be determined make computational methods for protein structure prediction an essential tool. However, the accuracy achieved by the most recent and relevant proposals in the literature is up to approximately 35 % [44] and clearly must be improved.

The first thing one has to decide when using a computational method is how to represent the data. In the PSP literature, there are three main data structures to represent a protein 3D structure: torsion angles, distance maps and contact maps. Torsion angles represent the values of the flexible angles of a protein molecule. Torsion angles are based on the assumption of constant bond lengths and some constant bond angles between atoms. This representation is based on three torsion angles in the protein backbone plus the angles in protein sidechains. This is a simplification of the real situation where the supposed constant bond lengths and angles depend on the environment of atoms. Examples of recent proposals that predict protein torsion angles are Faraggi et al. [17] and Furuta et al. [21].

On the other hand, distance maps represent the distances between reference atoms of each pair of protein residues. Examples of methods that predict protein distance maps are [4, 36].

Contact maps are the most commonly used structure in the PSP literature. In a nutshell, a contact map represents binary proximities between each pair of protein residues,

A. E. Marquez-Chamorro (✉) · G. Asencio-Cortes · F. Divina · J. S. Aguilar-Ruiz
Pablo de Olavide University Seville, Seville, Spain
e-mail: amarcha@upo.es

which are predicted by residue–residue contact predictors. The prediction of contact maps is a very important problem. For instance, there is a competition dedicated to contact map predictors in CASP [44], but there is no competitions in CASP for distance map or torsion angle predictors. More details about contact maps are given in Sect. 2.

In addition to this, some proposals discretize the distances between atoms, providing an intermediate representation between contact and distance maps. For instance, Walsh et al. [58] which uses 4-class distance maps.

As already mentioned, in PSP, the prediction of the 3D structure of a protein must be based on characteristics of the amino acids forming its sequence. Some commonly used features are the physico-chemical properties of residues. Usually the properties that are used are hydrophobicity, polarity, charge and residue size, as well as the properties of the AAindex repository [32], which contains currently 544 amino acid properties. On the other hand, predictors often use secondary structures (commonly from DSSP [31] or PSIPRED [28]), solvent accessibility [41], evolutionary information (commonly the Position Specific Scoring Matrix (PSSM) from PSIBLAST [2]) and contact orders (usually CO [47], RCO [33], CN [34] or the most recent RWCO [52]). Some authors also used topological measures of the protein molecule like the recursive convex hull [53].

Several types of approaches have been proposed in the literature with the aim of computationally solving the PSP problem and they will be detailed in Sect. 2. Within such proposals, evolutionary algorithms (EAs) have proven to achieve excellent results, see, for instance [8]. EAs have become popular as robust and effective methods for solving optimization problems. In particular, EAs have shown the capacity of finding suboptimal solutions in search spaces when the search space is characterized by high dimensionality. This is the case of the protein folding problem, where the set of possible folding rules of a protein determine the search space.

In this paper, we propose a method based on an EA to predict contact maps. In particular, the EA adopted is a multi-objective EA (MOEA), which bases its prediction on three physico-chemical properties (hydrophobicity, polarity and charge), on solvent accessibility and on secondary structure. We used an evolutionary approach based on the Strength Pareto Evolutionary Algorithm (SPEA) [67]. Our algorithm generates a set of decision rules that predicts contacts between amino acids. In particular, each rule imposes a set of conditions on some specific amino acids properties that, if satisfied, predict a contact.

The rest of the paper is organized as follows: in Sect. 2 we provide a general description of the main concepts regarding contact maps and their prediction, and we review the state-of-the-art of contact map prediction. In Sect. 3, we define the elements, procedures and evaluation measures used by our prediction method. In Sect. 4, we detail the predictions performed and the protein datasets that we used, we discuss the achieved results and we analyze the predicted rules. Finally, in Sect. 5, we describe the main conclusions of the work and we outline approaches for future studies.

2 Contact map prediction

The contact map of a protein sequence is a square matrix of order L, where L is the number of amino acids in the sequence. The 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, or 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) [44], a threshold that is also adopted in this paper. In order to measure the distance between two amino acids, it is necessary to use a reference atom of each amino acid, the most commonly used being the alpha carbon and the beta carbon of amino acids [15]. In our method, we use the beta carbon (with the exception of glycine, which has no beta carbon, and for which its alpha carbon is used).

Usually contacts between amino acids are divided and predicted by groups according to their sequence separation. Sequence separation between amino acids a_i and a_j, where i and j represent the positions of the residues in the sequence, is $|i - j|$. Based on the separations, contacts are classified into three classes: short, medium and long range. In short range, a minimum separation of six residues is used to consider a contact, whereas in medium and long range, the minimum separations are 12 and 24, respectively.

Contact maps present several advantages with respect to other representations. For instance, unlike 3D models of proteins, contact maps, as well as distance maps, have the desirable property of being insensitive to rotation or translation of the protein molecule. On the other hand, given a contact map of a protein, it is possible to reconstruct a 3D model of the protein backbone, solving the Molecular Distance Geometry Problem (MDGP) [39]. This can be done in different ways, e.g., using quadratic potential GO model [3] or using tools like FT-COMAR [56, 57]. It is also possible to obtain the coordinates of all protein atoms from the protein backbone using tools like SCWRL, IRECS, SCAP, SCATD or SCCOMP [19] or the recent tool SIDEpro [46]. Contact maps, as protein

structure representation, are also useful to compare protein structures, using the maximum contact map overlap [13].

Many different approaches for contact map prediction have been proposed in the literature, the three mostly used approaches being those based on artificial neural networks (ANNs) [41, 54, 63], evolutionary algorithms (EAs) [8, 9, 30] and support vector machines (SVMs) [42, 62].

Regarding the ANNs approaches, Xue et al. proposed SPINE-2D [63] that consists of two neural networks using one and two layers, respectively. These networks use 34 features as input, including PSSM from PSIBLAST [2], seven physico-chemical properties of amino acids, including hydrophobicity, volume and polarizability and secondary structure from the DSSP secondary-structure assignment program [31]. Tegge et al. [54] proposed NNcon which uses 2D-Recursive Neural Network (2D-RNN) models to predict both general residue–residue contacts and specific beta contacts (i.e., pairs of residues in beta sheets). They combine general and specific contact maps to produce predictions. Lippi et al. [41] proposed a novel hybrid architecture based on neural and Markov logic networks with grounding-specific weights, to predict contacts between β-strand residues. Multiple alignment profiles, secondary structure and solvent accessibility in two states were used as input.

As far as EAs are concerned, Chen and Li [9] proposed an ensemble of genetic algorithm (GA) classifiers to predict long-range contacts. The individuals of the GA include three amino acid windows and 20 properties obtained from HSSP database [14] for each residue in such windows. The method uses the sequence profile centers, that is, the average sequence profiles of residue pairs belonging to the same contact class or non-contact class. Judy et al. [30] propose a MOEA, representing protein structures by torsion angles. They modified the classical algorithm Pareto Archived Evolutionary Strategy (PAES) [11, 12], introducing two immune inspired operators: vaccination and immune selection. Their algorithm, named MI-PAES, uses adaptive probabilities of crossover, mutation and immune operation and a geometric annealing schedule in the immune operator. Calvo et al. [8] also proposed a MOEA, called Pitagoras-PSP. This algorithm uses an evolutionary ab initio approach based on PAES. The algorithm predicts protein backbone and side-chain torsion angles and it uses an energy function as fitness function. Mutation operators maintain values of torsion angles in feasible ranges according to secondary structure of residues and rotamer libraries.

With respect to SVM approaches, Wu et al. [62] developed a composite set of nine SVM-based contact predictors that are used in I-TASSER [50] simulation in combination with sparse template contact restraints. They used the original energy function of I-TASSER and contact predictions generated by extended versions of SVMSEQ [61]. Lo et al. [42] proposed a hierarchical scheme for contact prediction, with an application in membrane proteins. This approach consists of two levels: in the first level, contact residues are predicted from sequences, while in the second one their pairing relationships are further predicted. The statistical analyses on contact propensities are combined with evolutionary profile, relative solvent accessibility and helical features.

Apart from these three main approaches, there are other important approaches that try to address the residue–residue contact prediction problem. Li et al. [40] developed ProC_S3, based on a set of Random Forest algorithm based models using 1287 sequence-based features. Marks et al. [43] use a global model of maximum entropy constrained by correlated mutations from multiple sequence alignments. Finally, they reconstruct protein 3D models using distance geometry and simulated annealing. On the other hand, ensemble approaches, which combine several predictors, are also recently applied for contact map prediction [16, 22, 64], as well as nearest neighbor-based algorithms [1], Hidden Markov Models [7], integer linear optimization [48, 49, 60], sparse inverse covariance [29] and template-based approaches [5, 59].

3 Methods

Before describing our algorithm, this section presents a brief introduction to multi-objective optimization problems and related concepts.

A Multi-objective optimization problem requires the optimization of a set of objectives, usually in conflict with each other. The existence of multiple objectives poses a fundamental difference with the single objective problems: typically, there will not be a single solution, but a set of solutions that can present different clashes between the values of the objectives to optimize. We can define a multi-objective optimization problem in this way: let $(f_1(x), f_2(x), \ldots, f_n(x))$ be a set of functions to be optimized, where $x = (x_1, \ldots, x_p)$ is a vector of decision variables belonging to a universe X and $f_i(x)$ is an arbitrary linear or non-linear function, $1 \leq i \leq n$. Therefore, the problem consists of finding the x that provides the best compromise value for all $f_i(x)$.

To solve the above problem, we should defined some criteria to determine which solutions are considered of good quality and which are not. To this aim, the concept of dominance is generally used. A solution x is said to be not dominated *iff* there is not another solution y such that: $f_i(y) \leq f_i(x)$ for all $i = 1, \ldots, n$ and $f_i(y) < f_i(x)$ for some i. From this, it follows that the best solutions are those that are not dominated. This set is called Pareto front.

We have applied these concepts to the PSP problem. In this article, we consider two objectives to be optimized separately, which are defined in Sect. 5. In order to do so, we have implemented a MOEA, called MECoMaP (Multiobjective Evolutionary Contact Map Predictor), based on a SPEA. This algorithm uses an external population with non-dominated solutions, which is obtained at the end of every generation. The algorithm is based on the strength concept. 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 of this section.

Each individual of the population represents a decision rule. In particular, rules are based on some specific amino acid properties. Basically they specify a set of conditions on each property that, if satisfied, predict a contact between two amino acids. It is known that amino acid properties play an important role in the PSP problem [24]. Several PSP methods were proposed that relied on amino acids properties, e.g., hydrophobicity and polarity were employed in HP models [55]. In our approach, the decision rules evolved by our algorithm, base the prediction on three physico-chemical properties: hydrophobicity, polarity and charge of the residues. It has been shown that such properties have certain relevance in PSP. In addition to these properties, we also make use of two structural features of proteins: secondary structure prediction (SS) and solvent accessibility (SA). We selected the Kyte-Doolittle hydropathy profile [37] for hydrophobicity, the Grantham's profile [23] for polarity and Klein's scale for net charge [35]. In Table 1, we can appreciate the property values for each amino acid according to the cited scales, normalized between -1 and 1 for hydrophobicity and polarity.

Secondary structure prediction consists of predicting the location of alpha-helices, beta-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 [28].

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 to obtain the actual SA of each residue as described in [6]. In order to predict the RSA, 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 RSA, where lower values mean a buried state and higher values represent exposed states.

In the following, we address the various solutions adopted for what regards the representation, the genetic operators and the fitness function used by the EA.

3.1 Encoding

We represent a protein sequence by $s_1, \ldots, s_L$, where L is the sequence length and s_i $(1 \leq i \leq L)$ is an amino acid. Each individual in our algorithm represents a decision rule which determines whether amino acids s_i and s_j are in contact, with $1 \leq i < j \leq L$. For this purpose, we include in each individual properties of two windows of $\pm$ 3 residues centered around the two target amino acids s_i and s_j. Therefore, one window is relative to amino acids s_{i-3}, s_{i-2}, s_{i-1}, s_i, s_{i+1}, s_{i+2}, s_{i+3} and the other one is relative to amino acids s_{j-3}, s_{j-2}, s_{j-1}, s_j, s_{j+1}, s_{j+2}, s_{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 Eq. 1.

Table 1 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

$$Q_k = H_{\min}, H_{\max}, P_{\min}, P_{\max}, C, \mathrm{SS}, \mathrm{SA} \quad (1)$$

where

$-1 \leq H_{\min} < H_{\max} \leq 1$

$-1 \leq P_{\min} < P_{\max} \leq 1$

$C \in \{-1, 0, 1\}$

$\mathrm{SS} \in \{-1, 0, 1, 2\}$

$\mathrm{SA} \in \{-1, 0, 1, 2, 3, 4\}$

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

$$R_{i,j} = \{Q_k\} \quad (2)$$

Given a test sequence $t_1, \ldots, t_{L'}$, where L' is the test sequence length, and a pair of amino acids t_a and t_b ($1 \leq a < b \leq L'$), the algorithm predicts a contact between these amino acids if there exist any rule $R_{i,j}$ ($1 \leq i < j \leq L$) that covers the pair (t_a, t_b).

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

$$H_{\min} \leq H(t_k) \leq H_{\max} \quad (3)$$

$$P_{\min} \leq P(t_k) \leq P_{\max} \quad (4)$$

$$C(t_k) = C \quad (5)$$

$$\mathrm{SS}(t_k) = \mathrm{SS} \quad (6)$$

$$\mathrm{SA}(t_k) = \mathrm{SA} \quad (7)$$

where $H(t_k)$ is the hydrophobicity of the amino acid t_k, $P(t_k)$ its polarity, $C(t_k)$ its charge, $\mathrm{SS}(t_k)$ its secondary structure and $\mathrm{SA}(t_k)$ its solvent accessibility.

3.2 Genetic operators and fitness functions

The algorithm starts with a randomly initialized population and is run for a maximum number of generations. If the fitness of the best individual does not increase over 20 generations, the algorithm is stopped and a solution is provided. In order to obtain the next generation, individuals are selected with a tournament selection mechanism of size two. Crossover and mutation are then applied in order to generate offsprings.

Various crossover operators have been tested. In particular, we have tested the performances of one-point, two-points, uniform and BLX-α crossovers. These cross-over operators act at the level of the amino acid properties. For instance, one-point crossover 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. BLX-α crossover creates a new offspring $R_{i,j}$, where the values of the elements of Q_k (for each $k \in \{i-3, i-2, i-1, i, i+1, i+2, i+3, j-3, j-2, j-1, j, j+1, j+2, j+3\}$) are mutated within an interval delimited by the maximum and minimum values of the two parent individuals for the same element of Q_k. An α value is also selected to calculate this interval. In our case, we set the α value for the crossover to 0.1. This parameter must be higher or equal than 0. This crossover operator can be seen as a linear combination of the two parents. After having performed several runs of the algorithm, the best results were obtained when the two-point crossover was used, which was then used as standard crossover in the algorithm.

We have applied two different mutation operators. The first operator, called Gaussian operator, mutates an element of Q_k of an individual $R_{i,j}$, where $k \in \{i-3, i-2, i-1, i, i+1, i+2, i+3, j-3, j-2, j-1, j, j+1, j+2, j+3\}$, following a Gaussian distribution. The value of this element is increased or decreased with a probability of 0.5. If the values of a mutated individual are not within the allowed ranges for each properties, the mutation is discarded. 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 varies its range to all the allowed values. 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. This type of mutation is applied with a 0.1 probability. The parameter setting of the algorithm are shown in Table 2. This setting was determined after several preliminary runs.

The aim of the algorithm is to find both general and precise rules for identifying residue–residue contacts. The fitness of an individual is equal to the number of individuals that it dominates. We consider two objectives to be optimized, rule coverage and rule accuracy. Thus, an individual dominates another according to its values of rule coverage and rule accuracy. Rule coverage represents the proportion

Table 2 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

of contacts covered by each rule and rule accuracy evaluates the correctly predicted contacts rate by each rule. Therefore, Rule coverage $= C/C_t$ and Rule 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 set of non-dominated individuals are included in a external archive and they will be selected for the next generation.

An execution of the algorithm provides as a result a set of rules. If the algorithm is run several times, the final prediction model will consist of all the rules obtained at each execution. In other words, each time the algorithm is run, a number of rules are added to the final solution.

This is done in an incremental way: first, the best individual, according to its F-measure, is selected and added to the solution S. Then the next best individual is added to S, and the F-measure of S is calculated. This process is repeated until the addition of a rule causes the F-measure of S to decrease. The F-measure is defined as in Eq. 8:

$$F\text{-measure} = 2 \cdot \frac{\text{Rule coverage} \cdot \text{Rule accuracy}}{\text{Rule coverage} + \text{Rule accuracy}} \tag{8}$$

Repeated or redundant rules are not included in the final solution. Each pair of rules $(R_{a,b}, R'_{c,d})$ is checked. 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 9.

$$\begin{aligned} &H_{\min} \geq H'_{\min} \wedge H_{\max} \leq H'_{\max} \\ &P_{\min} \geq P'_{\min} \wedge P_{\max} \leq P'_{\max} \\ &C = C' \\ &\mathrm{SS} = \mathrm{SS}' \\ &\mathrm{SA} = \mathrm{SA}' \end{aligned} \tag{9}$$

The pseudocode of MECoMaP is shown in Algorithm 1. The evolutionary process is repeated *numIt* times where *numIt* is the number of iterations. The algorithm 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' is formed by the non-dominated individuals (external population A) and by the selected individuals with the binary tournament selection operator. The last 50 % of the individuals in P' is created using the 2-point crossover operator. Mutation is applied to the whole population, except to the Pareto front individuals, at the end of the evolutionary process. This process is repeated a maximum number of generations *maxGen*. At the end of each iteration a set of best individuals is stored in *Results* set. Hence, the *Results* set is formed in an incrementally way, as we have said before, and constitutes the output of our algorithm.

Algorithm 1 MECoMaP Algorithm for Contact Map Prediction

INPUT set of protein subsequences M, maximum number of iterations $numIt$, maximum number of generations $maxGen$, size of the population S.
OUTPUT set of generated rules *Results*.

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

4 Experimentation

In this section we will present the results obtained by MECoMaP on four different datasets. MECoMaP was 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.

The first protein data set (DS1) consists of 173 non-redundant proteins with sequence identity less than 25 % and was obtained from [18]. As in [18], four subsets have been obtained according to the sequence length (L_s): $L_s < 100$, $100 \leq L_s < 170$, $170 \leq L_s < 300$, $L_s \geq 300$. The minimum and maximum lengths of proteins are 31 and 753 amino acids, respectively. DS1 contains 240,501 positive examples and 5,034,050 negative examples (non-contacts).

The second data set (DS2), with 53 non-redundant and non-homologous globulin proteins, is detailed in [10]. In this case, proteins are classified according to their SCOP class [45] as described in [10]. Alpha proteins contain only alpha helical secondary structure. Beta proteins contain

only beta-sheet secondary structure. Alpha/beta proteins contain alternating α-helical and β-sheet secondary structure elements. This structure is known as a TIM barrel. In alpha/beta proteins, the alpha helical and beta sheet regions occur in independent regions of the molecule. Small proteins are referred to the size of the protein and are usually dominated by metal ligand or disulfide bridges. Finally, Coiled-coil proteins refers to a structural motif in which alpha helices are coiled together. The sequence identity of DS2 dataset is also lower than 25 %. DS2 is formed by a total of 30,546 contacts and 356,528 non-contacts.

The third data set is presented in [65]. This data set (DS3) includes 48 non-homologous proteins. DS3 is divided into five subsets according to L_s: $L_s < 100$, $100 \leq L_s < 200$, $200 \leq L_s < 300$, $300 \leq L_s < 400$, $L_s \geq 400$. DS3 contains 10,498 positive examples and 367,299 negative examples.

The fourth data set (DS4), is detailed in [29]. A total of 150 non-homologous proteins are contained in this data set. The sequence length of the proteins varies between 50 and 275 amino acids. DS4 is formed by 225,352 positive examples and 3,194,288 negative examples.

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

Before presenting the results obtained by our algorithm on the datasets, we present results of two preliminary studies. The first study was conducted to determine the distribution of the number of contacts according to the sequence separation between the pairs of amino acids. In this study, we have used the protein data set DS1. The result of this study is represented in Fig. 1. The *X*-axis represents the different possible values of sequence separations (the number of residues between those that are in contact) and the *Y*-axis represents the number of residue–residue contacts. The study concludes that the vast majority of contact occurrences (97 %) are established with a sequence separation lower than 140 amino acids. Therefore, we discard all the possible contacts with a sequence separation higher than 140 during the training phase in all the experimentations. Using this constraint, a considerable waste of computational time is avoided. Similar contact distributions were obtained for the other datasets.

Three statistical measures are used in CASP competitions [44] (coverage, accuracy and X_d) to evaluate the performance of protein structure predictors. The coverage and accuracy, defined in this context, have a different meaning to that expressed in Sect. 5. In particular, in CASP, coverage indicates what percentage of contacts

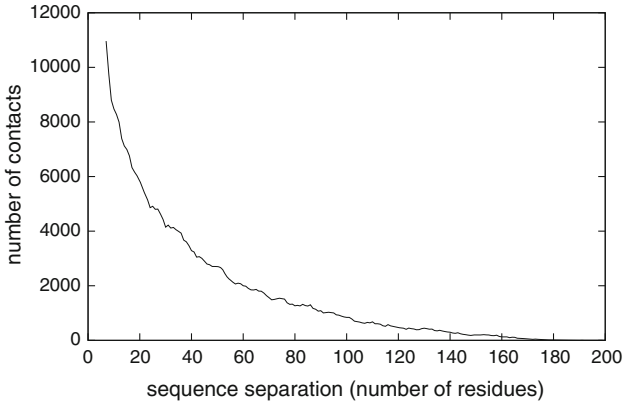

Fig. 1 Sequence separation vs. number of contacts

have been correctly identified. Accuracy reflects the number of correctly predicted contacts. X_d represents the distribution accuracy of the predicted contacts. X_d is defined by Eq. 10

$$X_d = \sum_{i=1}^{15} \frac{P_i - P_a}{i} \tag{10}$$

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 $4i$.

The other preliminary experiment we present is aimed at verifying if the encoding adopted by MECoMaP provides enough information to perform a good classification. To this aim, we have compared the results obtained by MECoMaP with those obtained by five well-known classifiers: Naive Bayes (NB), C4.5 classifier tree, Nearest Neighbor approach with $k = 1$ (IB1), Neural Network (NN) and Support Vector Machine (SVM). For this experimentation we have used DS1 ($L_s < 100$), DS2, DS3 and DS4. We have set the same experimental conditions in all the cases: a sequence separation of 6 amino acids and a threefold cross-validation was performed, as cited in [18]. From all the extracted data, we have built four files in ARFF format, with all the training data information. The positive class (contact) is represented with 1 and the negative class (no contact) is represented with 0. The total data sets consists of 123, 949 instances with 6, 922 positive and 117, 027 negative cases (contacts and no contacts respectively) for DS1, 171, 916 instances with 5, 530 positive cases and 166,386 negative cases for DS2 and 55, 988 instances, 18, 486 contacts and 1, 119, 751 no contacts for DS3 and 44, 444 positive cases and 1, 512, 823 negative cases for DS4. We have used the WEKA [26] implementation of C4.5 (J48), Naive Bayes (NB), IB1, Multilayer Perceptron (Neural Network) and Sequential minimal optimization algorithm (SMO) which represents a Support

vector machine approach. We set the parameters of the algorithms by default, except the parameter buidingLogisticModel which is set to true and -E flag of kernel which is set to 2.0. These settings belong to SMO approach.

Table 3 shows the results of this experiment. The obtained results are within normal values of accuracy and coverage rates for the contact map prediction [10]. These results confirm that our encoding provides enough information for a good performance of a learning classifier. Furthermore, we can also notice that MECoMaP achieved the best results for this experiment in the majority of the cases. High values of coverage are achieved by NB for DS2 and DS4; however, the accuracy rate is significantly low in these cases, so these results are overcome on average by our algorithm. On the other hand, NN achieves higher values of accuracy than MECoMaP for DS3, but its coverage is much lower than the coverage obtained by our method. In addition, we performed a statistical test (Friedman test) on the results shown in Table 3 and we found that the differences of accuracy values for DS2 and DS4 are statistically significant.

In order to further verify the correct performance of our approach, in Fig. 2 we show the different Pareto fronts for ten generations (from generation 10 to 100 with an interval of 10) of an execution of the algorithm on DS1 ($L_s < 100$) data set. Each different symbol represents an individual of the Pareto front in different generations. The *X*-axis represents the coverage and the *Y*-axis shows the accuracy rate. These two measures are the two objectives subject of optimization. We can notice how the quality of individuals improve with the generations. This result confirms that the algorithm is optimizing the two objectives.

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 else performed various experiments varying the numbers of runs of the EA, where to a higher number of runs corresponds a higher number of rules. The aim of this was to test whether 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.

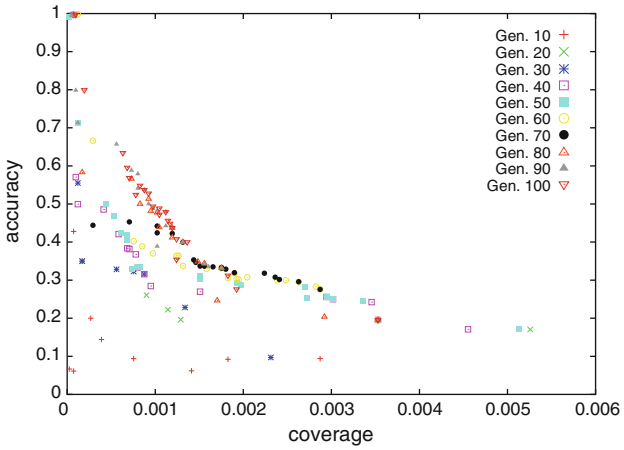

Fig. 2 First Pareto fronts for an execution in different generations

Table 4 shows the results for the first experimentation using DS1 data set. We used a sequence separation of seven residues and a threefold cross-validation as in [18]. We have compared our results with the ones shown 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 MECoMaP, 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 size of the proteins increases. This is due to the fact that, generally, ab initio methods only work well with peptides lower than 150 amino acids [20]. We obtain better results than [18] for proteins whose sequence length is lower than or equal to 100. We have obtained the same accuracy rates for the second subset and similar accuracy rates for the third and fourth group.

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). Low values of standard deviation show us that our data results are not

Table 3 Average accuracy, coverage and standard deviation values obtained for different Weka classification algorithms for the DS1, DS2, DS3 and DS4 protein data sets with the same experimental settings

Methods	DS1 data set		DS2 data set		DS3 data set		DS4 data set	
	Acc.$_{\mu \pm \sigma}$	Cov.$_{\mu \pm \sigma}$	Acc.$_{\mu \pm \sigma}$	Cov.$_{\mu \pm \sigma}$	Acc.$_{\mu \pm \sigma}$	Cov.$_{\mu \pm \sigma}$	Acc.$_{\mu \pm \sigma}$	Cov.$_{\mu \pm \sigma}$
IB1	0.11 $\pm$ 0.37	0.11 $\pm$ 0.27	0.05 $\pm$ 0.01	0.05 $\pm$ 0.01	0.08 $\pm$ 0.00	0.08 $\pm$ 0.00	0.07 $\pm$ 0.00	0.07 $\pm$ 0.00
J48	0.33 $\pm$ 0.31	0.03 $\pm$ 0.22	0.10 $\pm$ 0.05	0.08 $\pm$ 0.04	0.35 $\pm$ 0.03	0.41 $\pm$ 0.02	0.08 $\pm$ 0.01	0.07 $\pm$ 0.01
NB	0.14 $\pm$ 0.34	0.20 $\pm$ 0.39	0.09 $\pm$ 0.02	0.20 $\pm$ 0.02	0.19 $\pm$ 0.02	0.05 $\pm$ 0.08	0.08 $\pm$ 0.02	0.32 $\pm$ 0.03
NN	0.24 $\pm$ 0.05	0.10 $\pm$ 0.02	0.12 $\pm$ 0.07	0.04 $\pm$ 0.27	0.42 $\pm$ 0.01	0.07 $\pm$ 0.01	0.12 $\pm$ 0.00	0.03 $\pm$ 0.00
SMO	0.14 $\pm$ 0.14	0.03 $\pm$ 0.09	0.04 $\pm$ 0.02	0.06 $\pm$ 0.03	0.08 $\pm$ 0.01	0.09 $\pm$ 0.01	0.04 $\pm$ 0.01	0.09 $\pm$ 0.01
MECoMaP	0.54 $\pm$ 0.24	0.21 $\pm$ 0.18	0.38 $\pm$ 0.09	0.12 $\pm$ 0.01	0.37 $\pm$ 0.08	0.39 $\pm$ 0.02	0.36 $\pm$ 0.09	0.11 $\pm$ 0.03

Table 4 Efficiency of our method predicting DS1 protein data set

Protein length	#prot.	Acc.$_{\mu \pm \sigma}$	Acc.$_{\mu \pm \sigma}$ [18]
$L \leq 100$	65	0.54 $\pm$ 0.24	0.26 $\pm$ 0.39
$100 \leq L < 170$	57	0.21 $\pm$ 0.16	0.21 $\pm$ 0.32
$170 \leq L < 300$	30	0.17 $\pm$ 0.08	0.15 $\pm$ 0.22
$L \geq 300$	21	0.10 $\pm$ 0.05	0.11 $\pm$ 0.15
All proteins	173	0.26 $\pm$ 0.13	0.18 $\pm$ 0.32

significantly spread. In fact, standard deviation achieved by our predictor is lower than that achieved by the reference method [18] for all the protein lengths. Additionally, we performed a statistical test (Friedman test) with these values and we found that these differences are statistically significant.

Table 5 presents the results for the second experimentation using DS2 with a minimum sequence separation of six residues as in [10]. The first column of the table presents the SCOP classification, as was detailed at the beginning of this section. The second column shows the number of proteins of each subset, and the third and fourth columns show the average accuracy rate obtained by MECoMaP and by the algorithm presented in [10], respectively. Standard deviation is also reported. From Table 5, we can infer, as first conclusion, a good performance of our method for the beta proteins prediction. This is because most of the rules generated by our algorithms predict beta sheets. This observation will be validated in a further analysis presented in Sect. 4.1. We also obtain better accuracy than the algorithm proposed in [10], for alpha, small and coil–coil classes. In this cases, we have performed a non-parametric statistical test (Friedman test) on the results. After executing the test, the obtained p value was 0.039 (p value < 0.05), such that the null hypothesis was rejected. Therefore, the results obtained from this test sustain our conclusions, since the differences of the results obtained are statistically significant. We have also obtained positive values of X_d in all the cases, determining a good performance of our algorithm. Low standard deviation values show that the data results are clustered closely around the mean.

A third experiment compares our proposal with a neural network method (RBFNN) proposed in [65]. This method used the protein data set DS3. The authors of this work used an input and hidden layer with 20 nodes; the learning rate is set to 0.01, and the goal rate is set to 0.001. All the experimental results are based on the neural network toolbox of MATLAB version 6.5. The authors used coverage to evaluate the performance. Coverage is calculated using two variables: N_p represent predicted contacts by the algorithm and desired numbers and N_d is the total number of contacts. We used the same experiment settings as in [65]. Table 6 shows the results of this experimentation. The third column represents the coverage rate of our algorithm and the forth column represents the coverage rate of RBFNN. As we can see from this table, the average coverage is largely improved by our method in all the cases. Only the third subset is poorly predicted. This is due to the fact that only one protein is used as training set in this case and it seems to be insufficient to build an effective knowledge model.

Table 5 Efficiency of our method predicting DS2 protein data set

SCOP class	#prot.	Acc.	Acc. [10]
Alpha	11	0.30 $\pm$ 0.13	0.24 $\pm$ 0.13
Beta	10	0.42 $\pm$ 0.12	0.38 $\pm$ 0.16
$a + b$	15	0.38 $\pm$ 0.09	0.45 $\pm$ 0.10
a/b	7	0.22 $\pm$ 0.05	0.37 $\pm$ 0.07
Small	4	0.50 $\pm$ 0.01	0.36 $\pm$ 0.07
Coil–coil	1	0.25 $\pm$ 0.00	0.22 $\pm$ 0.00
All proteins	48	0.38 $\pm$ 0.08	0.37 $\pm$ 0.14

At the end of the execution, the program generates a resulting contact map for each protein test. In Fig. 3, we show an example for the protein 5PTI from DS1 data set. We can appreciate that the lower triangular (predicted contacts) is largely similar to the upper one (real contacts).

4.1 Analysis of predicting rules

In order to further evaluate the results obtained by MECoMaP, we have statistically analyzed the set of rules obtained on DS1 with $L_s < 100$. Similar results were found for other L_s. A set of 10,244 rules were extracted by the algorithm after an execution of 1,000 iterations. With this study, we want to analyze the properties of the amino acids that are predicted to be in contact. This would allow us to draw conclusions about the influence that these properties have on the folding problem.

First, we have analyzed the properties of the amino acid i in the rules set. The histograms in Figs. 4 and 5 show the relative frequency of hydrophobicity and polarity values for the amino acid i. The properties values have been discretized into five groups in intervals of 0.5 from -1 to 1 for the hydrophobicity and polarity. In Figs. 4 and 5 we added in each group of hydrophobicity and polarity, respectively, the rules whose interval $[H_{\min}, H_{\max}]$ is totally or partially included. Therefore, note that the same rule could be included in one or more groups. Although all the study is referred to residue i, the amino acid j presents similar behavior.

These rules indicate that a vast majority of amino acids in contact have high values of hydrophobicity. Furthermore, a high percentage of contacts have non-polar

Table 6 Efficiency of our method predicting DS3 protein data set

Protein length	#prot.	Cov.	Cov. [63]
$L \leq 100$	10	0.41	0.26
$100 \leq L < 200$	13	0.62	0.30
$200 \leq L < 300$	2	0.15	0.31
$300 \leq L < 400$	13	0.29	0.26
$L \geq 400$	9	0.75	0.26
All proteins	48	0.44	0.27

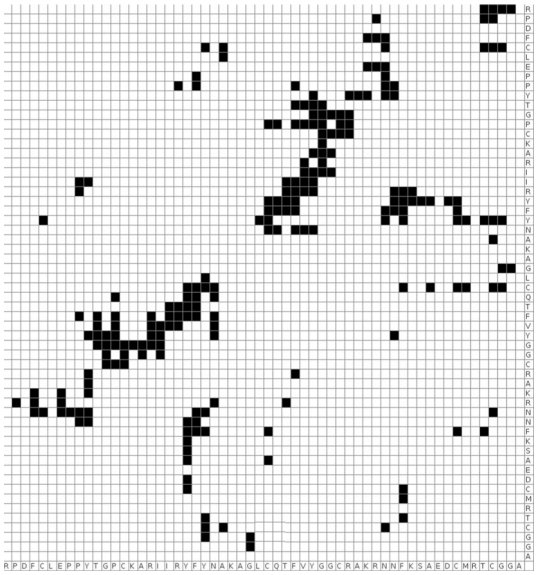

Fig. 3 Contact map of protein 5PTI with a threshold of 8Å

residues. These conclusions were expected because hydrophobic and non-polar amino acids tend to be located in the core of the protein. The core of proteins contains much less space than other protein regions, and contacts among amino acids are more frequent. Therefore, these type of residues have more probabilities to be in contact [25]. We have not observed any clear conclusion regarding the net charge of amino acids i and j individually. Although amino acids with opposite charges are supposed to be in contact [25], this condition seems to be irrelevant in our rule set.

Figures 6 and 7 represent the relative frequencies of values of solvent accessibility and secondary structure, respectively. As we can see in Fig. 6, lower values of solvent accessibility are the most represented values in the rules. This is due to the fact that amino acids with low values of solvent accessibility are in the core of the protein and are often in contact. We can appreciate from Fig. 7 that a high number of rules (77 %) presents secondary structure values of type E (β-sheets). This is explained by the separation in the sequence constraint which was set to 7. In fact, in this way, intra turn and intra α-helix contacts are

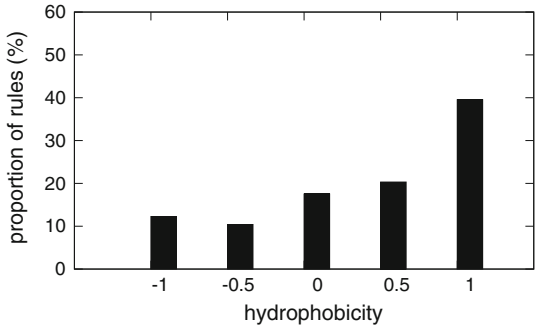

Fig. 4 Relative frequency of hydrophobicity values for amino acid i in our predicted rules

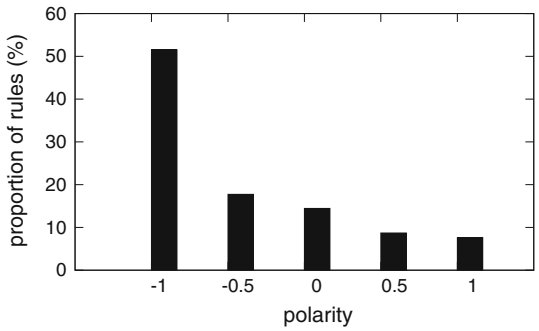

Fig. 5 Relative frequency of polarity values for amino acid i in our predicted rules

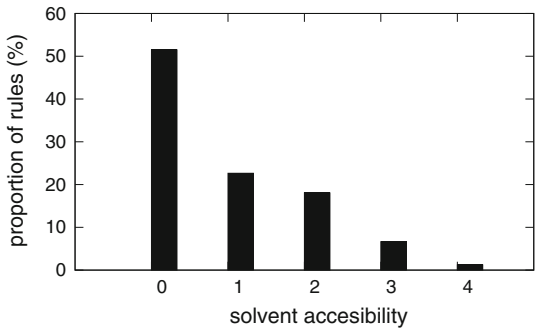

Fig. 6 Relative frequency of solvent accessibility values for amino acid i in our predicted rules

avoided. However, this fact does not affect the long-range β-sheet contacts, and therefore, they predominate in our set of rules.

We have also analysed the relation between the properties of amino acids i and j that are predicted to be in contact. In Figs. 8 and 9 we show the hydrophobicity and polarity regions, respectively, for amino acids i and j covered by our predicted rules. The representation of that regions is based on overlapping translucent rectangles whose area covers the range of hydrophobicities or polarities of amino acids i and j that are included in the rules.

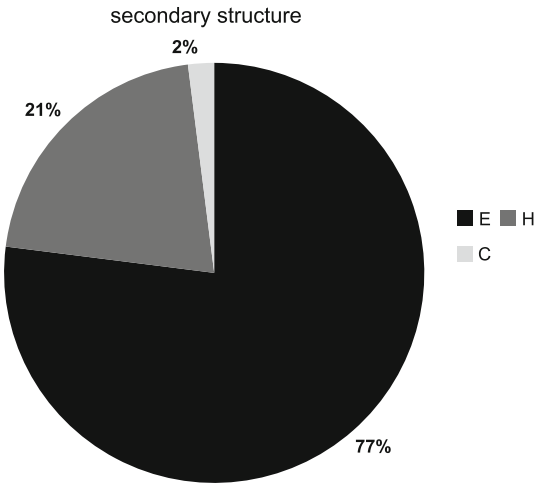

Fig. 7 Relative frequency of secondary structures for amino acid *i* in our predicted rules

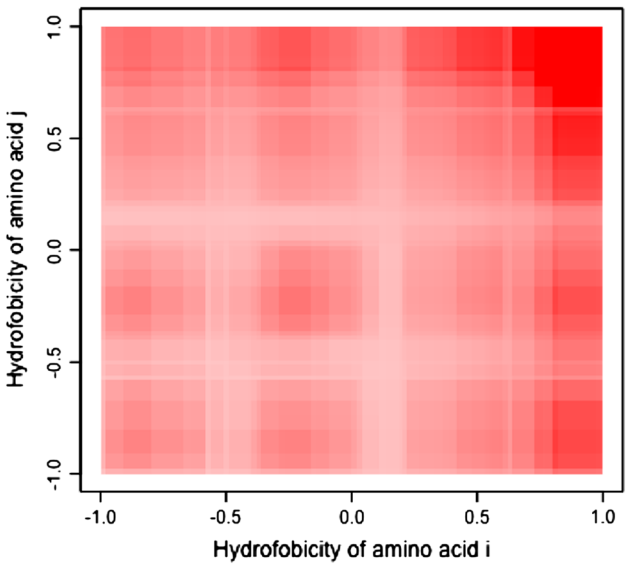

Fig. 8 Hydrophobicity regions for amino acids *i* and *j* covered by our predicted rules

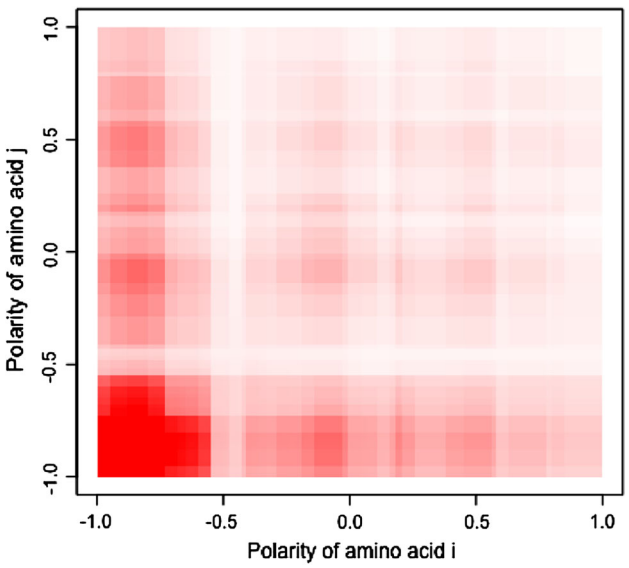

Fig. 9 Polarity regions for amino acids *i* and *j* covered by our predicted rules

From Fig. 8, we appreciate that the obtained rules predict contact between amino acids whose hydrophobicity is high, especially when both amino acids are hydrophobic (values close to 1.0). As we can observe in the Fig. 9, non-polar amino acids (values close to −1.0) are more likely to be in contact, according to our rules. These results are consistent with those obtained in Figs. 4 and 5. In Fig. 10 we show the relative frequency of charge values for amino acids *i* and *j* in the rules. We found that amino acids with charge 0 are often in contact (in 79.3 % of cases) according to the rules.

Figure 11 shows an example of two resulting rules. If we inspect the first rule, we can infer that, for example, the hydrophobicity value for the amino acid *i* lies between 0.52 and 0.92, the polarity value between −1.0 and −0.93, neutral charge 0, solvent accessibility 0 and secondary structure 2 (β-sheet). Therefore, the amino acid *i* could be L (Lysine) or F (Phenylalanine), which fulfills all these features according to the cited scales. As can be noticed, the produced rules are easily interpretable by experts in the field.

5 Conclusions and future work

In this article, we proposed a multi-objective evolutionary algorithm approach for protein contact map prediction. Our algorithm generates a set of rules for residue–residue contact prediction using a representation based on amino acid properties. The rules forming the final solution express a set of conditions on specific physico-chemical properties of amino acids. As a consequence, such rules can easily be interpreted and analyzed by experts in the field to obtain more insight into the protein folding process.

Our approach has been tested with four different protein data bases which appear in the literature obtaining acceptable results. A statistical study of our set of rules has been performed. Some conclusions about the folding protein can be inferred from the rules. These conclusions are related to the physico-chemical properties of amino acids (hydrophobicity and polarity) and two predicted structural features (SA and SS) used by our approach.

As for future work, we intend to expand this study to other significant amino acid properties, e.g., isoelectric point and steric parameter. Furthermore, we are planning to include evolutionary information like Position-Specific Score Matrix (PSSM). This information must be encoded in the representation of the algorithm. We also intend to

Amino acid j				
	+1	0.8	6.8	0.3
	0	2.5	79.3	5.4
	-1	0.3	3.5	0.8
		-1	0	+1
		Amino acid i		

Fig. 10 Relative frequency (%) of charge values for amino acids i and j in our predicted rules

$$if\ H_i \in [0.52, 0.92]\ and\ P_i \in [-1.00, -0.93]\ and$$
$$C_i = 0\ and\ SS_i = 2\ and\ SA_i = 0\ and$$
$$H_j \in [0.32, 0.82]\ and\ P_{j+1} \in [-0.41, -0.01]\ and$$
$$C_{j+1} = 0\ and\ SS_{j+1} = 2\ and\ SA_{j+2} = 1\ then\ contact$$

$$if\ H_{i-1} \in [0.20, 0.82]\ and\ P_i \in [-0.41, -0.01]\ and$$
$$C_i = 0\ and\ SS_{i+1} = 2\ and\ SA_{i+2} = 1\ and$$
$$H_j \in [0.45, 0.62]\ and\ P_{j+1} \in [-0.73, -0.01]\ and$$
$$and\ SS_j = 2\ and\ SS_{j+1} = 2\ then\ contact$$

$$...$$
$$else\ no\ contact;$$

Fig. 11 Description of two predicted rules obtained from the dataset DS1

study the possibility of using self-adaptable parameters for controlling the genetic operators used in the algorithm. Another future development is the application of the algorithm to larger proteins data set to test the validity of our proposal in these cases, where the resulting rules set could cover more of the search space.

References

1. Abu-Doleh AA, Al-Jarrah OM, Alkhateeb A (2011) Protein contact map prediction using multi-stage hybrid intelligence inference systems. J Biomed Inform
2. Altschul SF, Madden TL, Schffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Res Suppl 25(17):3389–3402
3. Andrew Toona GW (2012) A dynamical approach to contact distance based protein structure determination. J Mol Graph Model 32:75–81
4. Asencio Cortes G, Aguilar-Ruiz JS (2011) Predicting protein distance maps according to physicochemical properties. J Integr Bioinform 8(3):181
5. Ashkenazy H, Unger R, Kliger Y (2011) Hidden conformations in protein structures. Bioinformatics 27(14):1941–1947
6. Bacardit J, Stout M, Hirst J, Valencia A, Smith R, Krasnogor N (2009) Automated alphabet reduction for protein datasets. BMC Bioinform 10:6
7. Bjrkholm P, Daniluk P, Kryshtafovych A, Fidelis K, Andersson R, Hvidsten TR (2009) Using multi-data hidden markov models trained on local neighborhoods of protein structure to predict residue-residue contacts. Bioinformatics 25(10):1264–1270
8. Calvo JC, Ortega J, Anguita M (2011) Pitagoras-psp: including domain knowledge in a multi-objective approach for protein structure prediction. Neurocomputing 74(16):2675–2682
9. Chen P, Li J (2010) Prediction of protein long-range contacts using an ensemble of genetic algorithm classifiers with sequence profile centers. BMC Struct Biol 10(Suppl 1):S2
10. Cheng J, Baldi P (2007) Improved residue contact prediction using support vector machines and a large feature set. Bioinformatics 8:113
11. Cutello V, Narzisi G, Nicosia G (2006) A multi-objective evolutionary approach to the protein structure prediction problem. J R Soc Interface 3(6):139–151
12. Day RO, Zydallis JB, Lamont GB, Pachter R (2002) Solving the protein structure prediction problem through a multiobjective genetic algorithm. Nanotech 2:32–35
13. Di Lena P, Fariselli P, Margara L, Vassura M, Casadio R (2010) Fast overlapping of protein contact maps by alignment of eigenvectors. Bioinformatics 26(18):2250–2258
14. Dodge C, Schneider R, Sander C (1998) The hssp database of protein structure-sequence alignments and family profiles. Nucleic Acids Res Suppl 26(1):313–315
15. Duarte JM, Sathyapriya R, Stehr H, Filippis I, Lappe M (2010) Optimal contact definition for reconstruction of contact maps. BMC Bioinform 11:283
16. Eickholt J, Wang Z, Cheng J (2011) A conformation ensemble approach to protein residue-residue contact. BMC Struct Biol 11:38
17. Faraggi E, Yang Y, Zhang S, Zhou Y (2009) Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. Structure 17(11):1515–1527
18. Fariselli P, Olmea O, Valencia A, Casadio R (2001) Prediction of contact map with neural networks and correlated mutations. Protein Eng Des Sel 14:133–154
19. Faure G, Bornot A, de Brevern AG (2008) Protein contacts, inter-residue interactions and side-chain modelling. Biochimie 90(4):626–639
20. Fernandez M, Paredes A, Ortiz L, Rosas J (2009) Sistema predictor de estructuras de proteinas utilizando dinamica molecular (modypp). Revista Internacional de Sistemas Computacionales y Electronicos 1:6–16
21. Furuta T, Shimizu K, Terada T (2009) Accurate prediction of native tertiary structure of protein using molecular dynamics simulation with the aid of the knowledge of secondary structures. Chem Phys Lett 472(13):134–139
22. Gao X, Bu D, Xu J, Li M (2009) Improving consensus contact prediction via server correlation reduction. BMC Struct Biol 9:28
23. Grantham R (1974) Amino acid difference formula to help explain protein evolution. J Mol Biol 185:862–864
24. Gu J, Bourne P (2003) Structural bioinformatics. Wiley-Blackwell, New Jersey
25. Gupta N, Mangal N, Biswas S (2005) Evolution and similarity evaluation of protein structures in contact map space. Proteins Struct Funct Bioinform 59:196–204
26. Hall M, Frank E, Holmes GBP, Reutemann P, Witten I (2009) The weka data mining software: an update. SIGKDD Explor 11(1):10–18
27. Jaravine V, Ibraghimov I, Yu Orekhov V (2006) Removal of a time barrier for high-resolution multidimensional nmr spectroscopy. Nat Meth 3(8):605–607
28. Jones D (1999) Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 292(2):195–202

29. Jones DT, Buchan DWA, Cozzetto D, Pontil M (2012) Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics 28(2):184–190
30. Judy MV, Ravichandran KS, Murugesan K (2009) A multi-objective evolutionary algorithm for protein structure prediction with immune operators. Comput Methods Biomech Biomed Eng 12(4):407–413
31. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22(12):2577–2637
32. Kawashima S, Pokarowski P, Pokarowska M, Kolinski A, Katayama T, Kanehisa M (2008) Aaindex: amino acid index database, progress report 2008. Nucleic Acids Res Suppl 36(Database issue):D202–D205
33. Kihara D (2005) The effect of long-range interactions on the secondary structure formation of proteins. Protein Sci 14(8):1955–1963
34. Kinjo AR, Horimoto K, Nishikawa K (2005) Predicting absolute contact numbers of native protein structure from amino acid sequence. Proteins 58(1):158–165
35. Klein P, Kanehisa M, DeLisi C (1984) Prediction of protein function from sequence properties: discriminant analysis of a data base. Biochim Biophys 787:221–226
36. Kloczkowski A, Jernigan R, Wu Z, Song G, Yang L, Kolinski A, Pokarowski P (2009) Distance matrix-based approach to protein structure prediction. J Struct Funct Genom 10:67–81
37. Kyte J, Doolittle R (1982) A simple method for displaying the hydropathic character of a protein. J Mol Biol 157:105–132
38. Lattman E (2004) The state of the protein structure initiative. Proteins 54(4):611–615
39. Lavor C, Liberti L, Maculan N, Mucherino A (2012) Recent advances on the discretizable molecular distance geometry problem. Eur J Oper Res 219(3):698–706
40. Li Y, Fang Y, Fang J (2011) Predicting residue-residue contacts using random forest models. Bioinformatics 27(24):3379–3384
41. Lippi M, Frasconi P (2009) Prediction of protein beta-residue contacts by markov logic networks with grounding-specific weights. Bioinformatics 25(18):2326–2333
42. Lo A, Chiu YY, Rdland EA, Lyu PC, Sung TY, Hsu WL (2009) Predicting helix-helix interactions from residue contacts in membrane proteins. Bioinformatics 25(8):996–1003
43. Marks DS, Colwell LJ, Sheridan R, Hopf TA, Pagnani A, Zecchina R, Sander C (2011) Protein 3d structure computed from evolutionary sequence variation. PLoS One 6(12), e28766. doi: 10.1371/journal.pone.0028766
44. Monastyrskyy B, Fidelis K, Tramontano A, Kryshtafovych A (2011) Evaluation of residue-residue contact predictions in casp9. Proteins: Struct Funct Bioinform 79(Suppl 10):119–125
45. Murzin A, Brenner S, Hubbard T, Chothia C (1995) Scop: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247:536–540
46. Nagata K, Randall A, Baldi P (2012) Sidepro: a novel machine learning approach for the fast and accurate prediction of side-chain conformations. Proteins 80(1):142–153
47. Plaxco KW, Simons KT, Baker D (1998) Contact order, transition state placement and the refolding rates of single domain proteins. J Mol Biol 277(4):985–994
48. Rajgaria R, McAllister SR, Floudas CA (2009) Towards accurate residue-residue hydrophobic contact prediction for alpha helical proteins via integer linear optimization. Proteins 74(4):929–947
49. Rajgaria R, Wei Y, Floudas CA (2010) Contact prediction for beta and alpha-beta proteins using integer linear optimization and its impact on the first principles 3d structure prediction method astro-fold. Proteins 78(8):1825–1846
50. Roy A, Kucukural A, Zhang Y (2010) I-tasser: a unified platform for automated protein structure and function prediction. Nat Protoc 5(4):725–738
51. Service, R. Structural biology structural genomics, round 2. Science 307, 15541558 (2005)
52. Song J, Burrage K (2006) Predicting residue-wise contact orders in proteins by support vector regression. BMC Bioinform 7:425
53. Stout M, Bacardit J, Hirst JD, Krasnogor, N (2008) Prediction of recursive convex hull class assignments for protein residues. Bioinformatics 24(7):916–923
54. Tegge AN, Wang Z, Eickholt J, Cheng J (2009) Nncon: improved protein contact map prediction using 2d-recursive neural networks. Nucleic Acids Res Suppl 37(Web Server issue):W515–W518
55. Unger R, Moult J (1993) Genetic algorithms for protein folding simulations. Biochim Biophys 231:75–81
56. Vassura M, Margara L, Di Lena P, Medri F, Fariselli P, Casadio R (2008) Ft-comar: fault tolerant three-dimensional structure reconstruction from protein contact maps. Bioinformatics 24(10):1313–1315
57. Vassura M, Di Lena P, Margara L, Mirto M, Aloisio G, Fariselli P, Casadio R (2011) Blurring contact maps of thousands of proteins: what we can learn by reconstructing 3d structure. BioData Min 4(1):1
58. Walsh I, Bau D, Martin A, Mooney C, Vullo A, Pollastri G (2009) Ab initio and template-based prediction of multi-class distance maps by two-dimensional recursive neural networks. BMC Struct Biol 9(1):5
59. Wang Z, Eickholt J, Cheng J (2010) Multicom: a multi-level combination approach to protein structure prediction and its assessments in casp8. Bioinformatics 26(7):882–888
60. Wei Y, Floudas CA (2011) Enhanced inter-helical residue contact prediction in transmembrane proteins. Chem Eng Sci 66(19):4356–4369
61. Wu S, Zhang Y (2008) A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. Bioinformatics 24(7):924–931
62. Wu S, Szilagyi A, Zhang Y (2011) Improving protein structure prediction using multiple sequence-based contact predictions. Structure 19(8):1182–1191
63. Xue B, Faraggi E, Zhou Y (2009) Predicting residue-residue contact maps by a two-layer, integrated neural-network method. Proteins 76(1):176–183
64. Yang JY, Chen X (2011) A consensus approach to predicting protein contact map via logistic regression. In: Chen J, Wang J, Zelikovsky A (eds) Bioinformatics research and applications—7th international symposium, ISBRA 2011, Changsha, China, May 27–29, 2011. Proceedings, Lecture Notes in Computer Science, vol 6674, pp 136–147. Springer
65. Zhang G, Huang D, Quan Z (2005) Combining a binary input encoding scheme with rbfnn for globulin protein inter-residue contact map prediction. Pattern Recogn Lett 16(10):1543–1553
66. Zhou Y, Duan Y, Yang Y, Faraggi E, Lei H (2011) Trends in template/fragment-free protein structure prediction. Theor Chem Acc: Theory Comput Model (Theor Chim Acta) 128:3–16
67. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evol Comput 3(4):257–271