Journal of Structural Biology 189 (2015) 9–19

Contents lists available at ScienceDirect

Journal of Structural Biology

journal homepage: www.elsevier.com/locate/yjsbi

Improvement of protein binding sites prediction by selecting amino acid residues' features

Georgina Mirceva *, Andrea Kulakov

Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, Skopje, Macedonia

ARTICLE INFO

Article history:
Received 17 April 2014
Received in revised form 30 August 2014
Accepted 23 November 2014
Available online 3 December 2014

Keywords:
Protein binding site
Protein function
Protein interaction
Feature selection technique (FST)
Feature transformation technique

ABSTRACT

One of the main focuses of bioinformatics community is the study of the relationship between the structure of the protein molecules and their functions. In the literature, there are various methods that consider different protein-derived information for predicting protein functions. In our research, we focus on predicting the protein binding sites, which could be used to functionally annotate the protein structures. In this paper we consider a set of sixteen amino acid residues' features, and by applying various feature selection techniques we estimate their significance. Although the number of features in our case is not high, we perform feature selection in order to improve the prediction power and time complexity of the prediction models. The results show that by applying proper feature selection technique, the predictive performance of the classification algorithms is improved, i.e., by considering the most relevant features we induce more accurate models than if we consider the entire set of features. Furthermore, the model complexity, as well as the training and testing times are decreased by performing feature selection. We also compare our approach with several existing methods for protein binding sites prediction. The results demonstrate that the existing methods considered in this research are specific and applicable to the group of proteins for which the model was developed, while our approach is more generic and can be applied to a wider class of proteins.

1. Introduction

Protein molecules constitute an important part of the cells in living organism due to their involvement in various essential processes in the cells. Each protein has particular functions in a cell. High-throughput technologies provide vast amount of data stored in protein databases. This data could be used to annotate proteins whose functions are not discovered yet. Various methods could be used for annotating protein structures in experimental manner. However, these methods are labour intensive, expensive and time-consuming. Subsequently, protein annotation could not be performed at a speed comparable to that of discovery of new protein structures.

Since many research groups work on protein function prediction, it was necessary to define a standard for unified representation of the knowledge about proteins' annotations, thus Gene Ontology (GO) (The Gene Ontology Consortium, 2008) was introduced. GO is a controlled and structured vocabulary of the protein annotation terms, which are divided in three groups: molecular function, biological process and cellular component. For each annotation an evidence code is stored, which indicates the manner in which the annotation was discovered. In Du Plessis et al. (2011), an analysis of the evidence codes from GO is performed. This analysis shows that as of April 2010, 98.08% of the annotations are computationally discovered and are not curated, 0.7% are computationally inferred and are curated, while only 0.61% of the annotations are experimentally discovered. From this fact, the importance of computational methods in predicting protein functions is understandable.

In the literature, there is a range of computational methods for annotating protein structures. We categorize the computational methods for annotating protein structures into six main groups. The first group of methods inspects the homology in protein sequences, since homologous proteins are more likely to share common functions. Therefore, comparison of a query sequence with the known protein sequences (Altschul et al., 1990) can be performed in order to identify homologous proteins. Nevertheless, some newly discovered sequences will not have a homolog among the known proteins, thus other approaches are often required. The second group of methods (Sigrist et al., 2010) annotates protein

* Corresponding author at: ul. "Rugjer Boshkovikj" 16, P.O. Box 393, 1000 Skopje, Macedonia. Fax: +389 2 3088 222.

E-mail addresses: georgina.mirceva@finki.ukim.mk (G. Mirceva), andrea.kulakov@finki.ukim.mk (A. Kulakov).

http://dx.doi.org/10.1016/j.jsb.2014.11.007
1047-8477/

structures based on signatures (motifs) found in their sequences. The third group of methods determines protein annotations based on structure similarity. Protein structures have higher conservation than sequences, so structurally similar proteins are more likely to have similar functions. There are many methods for protein structure retrieval, like the methods proposed in Holm and Sander (1993), Shindyalov and Bourne (1998), Ye and Godzik (2004), etc. The fourth group of methods annotates protein structures by detecting the protein binding sites (Tuncbag et al., 2009) based on the amino acids residues' features. A recent publication (Lu et al., 2013) presents the most widely used features for binding sites prediction, and reviews the latest methods for binding sites prediction. The fifth group of methods (Panchenko et al., 2004) annotates protein structures based on the conserved parts of the sequences/structures that do not change throughout evolution. Protein Interactions by Structural Matching (PRISM) method (Keskin et al., 2008) considers both sequence and structure conservation to identify the binding sites of the template structures. Then, the binding sites of the query are determined by structural matching with the template structures. The sixth group of methods annotates protein structures using protein–protein interaction networks (Sharan et al., 2007).

In our research we focus on developing methods for protein binding site prediction. There are various methods used for this purpose, like distance-based methods (Mihel et al., 2008; Ofran and Rost, 2003; PRINT, 2013), methods that examine the sequence and/or structure conservation (Aytuna et al., 2005; Capra and Singh, 2007; Jones and Thornton, 1997), methods based on identifying pockets (An et al., 2005; Hendlich et al., 1997; Laskowski, 1995), etc. Also, there are methods that combine various information. For example, ConCavity (Capra et al., 2009) considers both sequence conservation and 3D structure to make more accurate predictions.

Accessible Surface Area (ASA) (Shrake and Rupley, 1973), Relative ASA (RASA), depth index (DPX) (Pintar et al., 2003), protrusion index (CX) (Pintar et al., 2002) and hydrophobicity (Kyte and Doolittle, 1982) are the most widely used amino acid residues' features. Since an amino acid residue is constituted of several atoms, we can extract several features regarding ASA and RASA by summing the characteristics over different sets of atoms (all atoms, backbone atoms, side-chain atoms, polar atoms or non-polar atoms), as in Mihel et al. (2008). Also, DPX and CX of an amino acid residue could be calculated as an average, maximum or minimum of the DPXs or CXs of the atoms that constitute the residue (Mihel et al., 2008).

In Mirceva and Kulakov (2012a,b), the models for protein binding sites prediction are induced by considering only total ASA, average DPX, average CX and hydrophobicity. Besides these four features, we can consider some additional features of the amino acid residues, and then by using an appropriate technique we can select the most valuable set of features. Although the number of features is not very high, the number of samples (amino acid residues) that would be used is huge, thus reducing the dimensionality of the dataset is an important issue. By applying appropriate feature selection and transformation techniques, besides decreasing model's complexity and reducing training and testing times, also the prediction power of the models could be increased due to elimination of the irrelevant features. Furthermore, the models with lower complexity are more easily interpretable.

From the feature transformation techniques, we apply Principal component analysis (PCA) (Abdi and Williams, 2010; Pearson, 1901) to transform the original features into new ones, and then we perform reduction. The feature selection techniques (FSTs) generally can be divided as filter and wrapper techniques (Kohavi and John, 1997). The former are not related to the model induction method and do not optimize the final criterion, while in wrapper techniques the induction method is used to build models using different subsets of features, and the optimal subset is chosen to maximize the final criterion. There are embedded schemes where the induction algorithm has embedded FSTs. In this research we focus on the filter and wrapper techniques. From the filter FSTs, in this research we consider several techniques that independently rank the features (Hunt et al., 1966; Kira and Rendell, 1992; Liu and Setiono, 1995; Quinlan, 1993), and then we consider the features with highest ranks. However, considering the top n features do not mean that the optimal subset of n features is chosen. Namely, the top ranked features could have high dependency with the class attribute, but may also have high dependency between themselves too. Therefore, we can use techniques that evaluate subsets of features. We identify three categories of these techniques, i.e., exponential, sequential and randomized. We consider the exhaustive search, which is an exponential method. This methodology is, however, time intensive and could be used if the number of features is not high, as in our case. In order to avoid examination of all subsets of features, a heuristic approach (Pearl, 1984) could be used. We use several sequential techniques that sequentially add or remove features, but they can get stuck in a local minima. Using randomization, with genetic algorithms (Goldberg, 1989) we can escape local minima. Another popular method is the minimal-Redundancy-Maximal-Relevance (mRMR) technique (Peng et al., 2005), which was recently used for feature selection in protein disulphide bond prediction (Niu et al., 2013) and protein–protein interaction prediction (Liu et al., 2013). From the filter techniques, we also use the technique given in (Guyon et al., 2002).

In this paper, we induce models for protein binding sites prediction in three steps. In the first step, we extract several features for each amino acid residue. In the second step, a range of FSTs are used to identify the most valuable features. Finally, in the third step, prediction models are induced by using several classification methods (Freund and Mason, 1999; Friedman et al., 1997; Gama, 2004; Huang et al., 2008; John and Langley, 1995; Kohavi, 1996; Quinlan, 1993; Senge and Hüllermeier, 2011). With applying appropriate FSTs we expect that the prediction models would be improved. Also, we make comparison with several existing methods for protein binding site prediction that are proposed in An et al. (2005), Aytuna et al. (2005), Capra et al. (2009), Capra and Singh (2007), Hendlich et al. (1997), Jones and Thornton (1997), Keskin et al. (2008), Laskowski (1995), Mihel et al. (2008), Ofran and Rost (2003), PRINT (2013).

The rest of this paper is organized as follows. In Section 2, we present our approach for protein binding sites prediction. Section 3 shows some results of the evaluation of our approach. Also, we compare our approach with several existing methods for protein binding sites prediction. Finally, in Section 4 we conclude the paper and point out possibilities for further improvements.

2. Materials and methods

2.1. Extraction of the amino acid residues' features

Protein chains are constructed from amino acid residues that contain several atoms. First, we extract several features for each atom, and then we calculate the features for each residue based on the features of its atoms. We consider the following features: Accessible Surface Area (ASA) (Shrake and Rupley, 1973), Relative ASA (RASA), depth index (DPX) (Pintar et al., 2003), protrusion index (CX) (Pintar et al., 2002) and hydrophobicity (Kyte and Doolittle, 1982). We use the Protein Structure and Interaction Analyzer (PSAIA) software (Mihel et al., 2008) for extracting these features of the amino acid residues.

ASA is introduced in Lee and Richards (1971), and is calculated using the rolling ball algorithm (Shrake and Rupley, 1973) where a probe sphere with a predefined radius (typically 1.4 Å) is used to estimate the surface of an atom that is accessible to the probe sphere. For each amino acid residue we calculate its total ASA, main-chain ASA, side-chain ASA, polar ASA and non-polar ASA, by summing ASA over various set of atoms (Mihel et al., 2008).

Many residues are hidden in the protein interior and could not be reached by the probe sphere. These residues could not be part of a binding site, so we filter only the surface amino acid residues. We consider that a given residue is at the protein surface if at least 5% of its surface is accessible by the probe sphere (Chothia, 1976).

Different amino acids have different number of atoms, thus ASA would be higher for some amino acids. Therefore, RASA could be used, which is a ratio between the ASA of a residue and the standard ASA (Hubbard and Thornton, 1993) of the corresponding amino acid. As for ASA, we extract the total RASA, main-chain RASA, side-chain RASA, polar RASA and non-polar RASA (Mihel et al., 2008).

DPX (Pintar et al., 2003) of an atom shows how far the atom is from its nearby atom that is accessible to the probe sphere. The depth index of the atoms that are accessible to the probe sphere is zero, and is greater than zero for the other atoms.

CX is introduced in Pintar et al. (2002), and it indicates the density of the region where a given atom is located. The number of non-hydrogen atoms N_{atom} within a sphere of radius $R = 10$ Å around the atom is calculated. The volume around the atom occupied by the protein is estimated as $V_{int} = N_{atom} * V_{atom}$, where V_{atom} is the mean volume of an atom (20.1 Å^3). The difference between the entire volume of the sphere and the volume occupied by the protein is denoted as V_{ext}. Finally, the protrusion index is calculated as V_{ext}/V_{int}.

For each amino acid residue we calculate the average, minimum and maximum of the DPXs and CXs of its atoms. Minimum DPX is zero for all surface residues, so we discard this feature.

Hydrophobicity (Kyte and Doolittle, 1982) indicates the hydrophobic properties of the amino acids. Namely, hydrophobic amino acids are more commonly found in the protein interior, and hydrophilic amino acids are typically located near the protein surface. We consider the hydrophobicity scale introduced by Kyte and Doolittle (1982).

2.2. Dataset description

As a standard of truth, we use the Biomolecular Interaction Network Database (BIND) (Bader et al., 2001) that holds knowledge regarding the protein binding sites that are determined experimentally. Since the number of known protein structures is huge and there is a large redundancy among them, therefore usually only the representative protein chains with low sequence similarity are considered. The test dataset is formed by the residues of 3530 chains that have less than 10% sequence similarity between themselves, using the criterion given in Chandonia et al. (2004). Using the same criterion, the training dataset is formed by the residues of 633 chains that are not previously used in forming the test dataset and have less than 20% sequence similarity. Next, we filter the surface residues as described before, and thus we obtain 115,579 residues in the training and 625,939 in the test dataset. From the training residues, only 15,696 (around 13.58%) are part of binding sites according to BIND (Bader et al., 2001), meaning that the non-binding sites class is dominant. In order to prevent inducing models that are biased toward the dominant class, we balance the training dataset by down sampling the dataset to 27% of its original size without replacement of the samples and by following uniform distribution of the class attribute. The test dataset, named B3530, remains unbalanced, thus for evaluation purposes some appropriate evaluation measure should be used. After balancing, the features are normalized in the interval [0,1]. The test dataset B3530 is used for evaluation of our approach using various FSTs combined with various classifiers. In Table 1 we provide summary statistics of the datasets used in this research.

We compare our approach with the methods proposed in An et al. (2005), Aytuna et al. (2005), Capra et al. (2009), Capra and Singh (2007), Hendlich et al. (1997), Jones and Thornton (1997), Keskin et al. (2008), Laskowski (1995), Mihel et al. (2008), Ofran and Rost (2003), PRINT (2013). For evaluation of the methods given in Aytuna et al. (2005), Jones and Thornton (1997), Mihel et al. (2008), Ofran and Rost (2003), we use the PSAIA software (Mihel et al., 2008), while for the other methods we use the pre-calculated predictions available at PRISM, PRINT and ConCavity websites. We use two datasets in the comparison. B1549 test dataset is formed by considering the chains from B3530 for which we obtained predictions by the methods given in Aytuna et al. (2005), Capra et al. (2009), Capra and Singh (2007), Jones and Thornton (1997), Keskin et al. (2008), Mihel et al. (2008), Ofran and Rost (2003), PRINT (2013).

The other dataset used in the comparison includes the knowledge stored in the LigASite v7.0 database (Dessailly et al., 2008), which contains biologically relevant binding sites in proteins with known apo-structures. This database contains both the redundant and non-redundant (<25% sequence similarity) sets. In this research we take into account only the chains in the non-redundant set in order to consider the most representative chains that are not homologous. The test dataset named L213 is obtained by considering the residues of the 213 chains from the non-redundant set for which we have predictions using all examined methods, while the residues of the remaining 329 chains from the non-redundant set are members of the training dataset. On this training dataset, we perform balancing by down sampling the dataset to 20% of its original size without replacement of the samples and by following uniform distribution of the class attribute. Also, we perform normalization as described before. L213 test dataset is used in the comparison with all examined methods.

2.3. Feature selection and transformation

In this paper we use several techniques for feature selection and transformation in order to reduce the number of features. This task is critical in our case since the number of samples is high, so reducing the dimensionality (complexity) of the problem is of high importance. Furthermore, by selecting the most relevant features, the predictive performances of the models could be improved, and also the training and testing times as well as model complexity could be decreased.

2.3.1. Feature transformation technique

In the literature, various feature transformation techniques are provided. From this category, we use the Principal Component Analysis (PCA) (Abdi and Williams, 2010; Pearson, 1901) in order

Table 1
Summary statistics of the datasets. For training datasets, we present the characteristics before balancing.

Training dataset	BIND		LigASite
#Chains	633		329
#Samples	115,579		67,522
Binding samples	13.58%		9.72%
Test dataset	**B3530**	**B1549**	**L213**
#Chains	3530	1549	213
#Samples	625,939	277,735	37,886
Binding samples	14.74%	16.42%	11.30%

to reduce the number of features. PCA transforms the original correlated features into novel features that are not correlated, and they correspond to the eigenvectors of the covariance matrix. The features' reduction is performed by considering the eigenvectors with highest eigenvalues. In this way we do not apply classical feature ranking, but we make ranking and reduction of the features obtained with the PCA transformation. We made tests by considering eight and four eigenvectors to cover 97% and 85% of the variance of the original samples.

2.3.2. Feature selection techniques

Instead of transforming the original features into novel features and reducing the number of features by filtering the novel features with highest relevance, we may apply some feature selection technique (FSTs) where the selection process is performed over the original features. In the literature, a range of FSTs are provided, and in this paper we take into account the techniques that are most commonly used. In Table A.1 given in the Appendix these techniques are systematically categorized.

The feature selection techniques (FSTs) could be divided into filter and wrapper techniques (Kohavi and John, 1997). Wrapper techniques consider some classification method for model induction in order to identify the most appropriate set of features by maximizing the final objective function (ex. classification accuracy), while filter techniques optimize some other objective function (ex. correlation between the features and the class attribute). Further, the filter techniques could be categorized into techniques that rank the features independently and techniques that evaluate subsets of features. The former category aims to find the features that have highest dependency with the class attribute, while the later additionally inspects the redundancy among the features in the subset of features. Next, we give a short description of FSTs used in this research.

2.3.2.1. Filter techniques that individually rank the features. We use several techniques to individually rank the features by using different measures. First, we use the Chi-Square test (Liu and Setiono, 1995) to measure the dependency between the inspected feature and the class attribute. Chi-Square test could be used for discrete features, so we apply discretization by using the criterion proposed in Fayyad and Irani (1993). The discretized data is also used for the other FSTs that require discrete attributes.

We also use the information gain (Hunt et al., 1966; Quinlan, 1993) and gain ratio (Quinlan, 1993) to rank the features. The information gain indicates the drop of the entropy after selecting a feature, but it favors the features with higher number of values. Therefore, in the gain ratio, the information gain is normalized by dividing it with the entropy of the feature.

We also use the Relief technique (Kira and Rendell, 1992), where instance based learning is used to estimate the features' weights (ranks). The procedure presented in Kira and Rendell (1992) could be used for binary problems, and considers one nearest neighbor from each class. We use the procedure given in Kononenko (1994) that considers k-nearest neighbors from each class, and could be used for multi-class problems. Using this technique, the distances with the neighbors within the same class are penalized, while the distances with the neighbors of the other classes are favored. Additionally, these distances could be weighed by using weights $\exp(-(j/\sigma)^2)$, where j is the rank of the neighbor. In this research we use $\sigma = 2$. We consider 10 nearest neighbors from each class, and we apply feature selection with and without weighting.

2.3.2.2. Filter techniques that evaluate subsets of features. Previous techniques independently rank the features and aim to maximize the relevance of the features, but do not inspect their redundancy. Other filter techniques evaluate subsets of features by using some measure. In the techniques described below, the Pearson's correlation coefficient (PCC) (Hall, 1999) is used to evaluate the subsets of features. Since the class attribute is nominal, a separate binary attribute is formed for each value of the class attribute, and the correlation is calculated by averaging the correlations between the inspected feature and these binary attributes.

We identify three categories of filter techniques that evaluate subsets of features, i.e., exponential, sequential and randomized. We use the exhaustive search, where all subsets of features are examined. This search belongs to the first category because the number of subsets increases exponentially as the number of features increases.

To avoid brute-force examination of all subsets, a heuristic search (Pearl, 1984) could be used. Therefore, we use the greedy stepwise approach that belongs to the sequential techniques because it adds or removes the features sequentially. This search could be performed in a forward or backward direction. In a forward selection, the initial subset is initialized to an empty set. The superset that provides the highest increase in PCC is chosen as a current optimal set. This procedure is recursively repeated until some termination criterion is satisfied. In this research, the procedure stops when the set contains all features, and the features are ranked according to the order of inclusion. The backward elimination goes in a reverse direction, where the initial set contains all the features and sequential elimination is performed by removing the most irrelevant features. We also use the best-first search that allows adding or removing a feature in each step by constraining the intensity of backtracking. We restrict the backtracking by terminating when the number of consecutive non-improving sets becomes five, and we start with an empty set of features. By using these approaches, we can avoid evaluation of all subsets of features, but we could get stuck in a local minima. This could be escaped by using randomization techniques.

From the category of randomization techniques, we use a genetic algorithm (GA) (Goldberg, 1989), which considers a generation with several solutions (subsets of features in this case), and evolve them to the optimal solutions. A new generation of solutions is formed by combining the solutions from the previous generation (by making a crossover), where the solutions with higher fitness values (PCC in this case) have higher probabilities to be considered for making a new generation. In GA, mutations may also occur, which means that the features could be added or removed randomly, thus escaping local minima. We consider two probabilities of crossover, i.e., 0.6 and 1, and for both values the same subsets of features are selected.

It has been observed in recent literature that the minimum-Redundancy-Maximum-Relevance (mRMR) (Peng et al., 2005) technique is being increasingly used for feature selection. The mRMR technique is based on mutual information instead of correlation. The method maximizes an objective function $\Phi(D,R)$ in order to maximize the relevance D and to minimize the redundancy R in the same time. In Ding and Peng (2005), two schemes are provided, i.e., Mutual Information Difference (MID) and Mutual Information Quotient (MIQ). In MID, the objective function that is maximized is $\Phi(D,R) = D - R$, while in MIQ the objective function is $\Phi(D,R) = D/R$. In this research we use both schemes. Since mRMR is applicable for nominal features, therefore we discretize the features in ten intervals with equal widths.

We also use the technique used in Guyon et al. (2002), where a recursive backward elimination of the features is performed. For each inspected subset of features, a prediction model is induced by using the SVM classifier. Then, the ranks of the features are calculated based on the weights obtained by the SVM models induced for each subset of features.

2.3.2.3. Wrapper techniques. Wrapper techniques (Kohavi and John, 1997) consider some model induction method to find the optimal subset of features by optimizing the final objective function. In this research, a forward selection of the features is performed in order to find the optimal subset of features. We use the following classifiers for model induction: C4.5 Tree (Quinlan, 1993), Alternating Decision Tree (ADTree) (Freund and Mason, 1999), Naïve Bayes (John and Langley, 1995), Naïve Bayes Tree (NBTree) (Kohavi, 1996), Bayesian Network (BayesNet) (Friedman et al., 1997) and K-Nearest Neighbours (KNN) (Aha et al., 1991). We apply repeated 2-fold cross validation over the training data to evaluate the significance of a given subset of features.

2.3.3. Software and experimental setup

For mRMR we use the implementation provided at http://penglab.janelia.org/proj/mRMR/ (Accessed March 12, 2013), while for the other FSTs we use the implementation provided in the Weka software (Hall et al., 2009). We use the default settings, except if it is otherwise stated, and we want to note that the test samples are not used for feature selection.

2.4. Induction of the prediction models

2.4.1. Classical classification methods

Next, we induce models for protein binding sites prediction. For this purpose we consider several classical (crisp) classification methods, i.e., C4.5 Tree (Quinlan, 1993), Alternating Decision Tree (ADTree) (Freund and Mason, 1999), Functional Trees (FTree) (Gama, 2004), Naïve Bayes (John and Langley, 1995), Naïve Bayes Tree (NBTree) (Kohavi, 1996) and Bayesian Network (BayesNet) (Friedman et al., 1997).

2.4.2. Fuzzy classification methods

However, the classical classification methods are sensitive to small changes in the data (the features of the amino acid residues in our case) that may arise during evolution. To overcome this, besides the classical classification algorithms, we also consider two existing fuzzy-based classifiers, i.e., the bottom-up (Huang et al., 2008) and top-down (Senge and Hüllermeier, 2011) methods for inducing Fuzzy Pattern Trees (FPTs). In Mirceva and Kulakov (2012a,b) and Naumoski et al. (2012), these methods are used for protein binding sites prediction, and discovering the diatoms' indicating preferences in water ecosystems. We feel that there is a need to explain these models, as they are relatively new and less known than the classical methods. First, we describe the bottom-up approach (Huang et al., 2008), and then we point out the differences with the top-down approach (Senge and Hüllermeier, 2011).

2.4.2.1. Bottom-up fuzzy pattern trees. First, the dataset is fuzzified by using some fuzzy membership function (FMF). In this research we use the triangular, trapezoidal and Gaussian FMFs. With fuzzification, each feature is labeled with a predefined number of fuzzy terms. For induction of the fuzzy models we use sets of four features and we set the number of fuzzy terms per feature to 5, thus we have 20 different fuzzy terms. For each fuzzy term a separate tree, named primitive tree, is induced thus obtaining 20 primitive trees (trees at the lowest level). These primitive trees could be used as models, but they are too simple and cannot provide accurate predictions. Therefore, these primitive trees are aggregated in order to provide more accurate model. For that purpose, first, by using the Root-Mean Squared Error (RMSE) we calculate the similarity between the membership values of a given fuzzy term and the inspected class. Then, the primitive tree with highest similarity is aggregated with the other primitive trees by using fuzzy aggregation operators, thus several candidate trees are obtained. From these trees, the tree with the highest similarity is identified and it is further aggregated with the remaining primitive trees. This procedure is repeated until the tree's depth becomes 5. In the process for aggregating trees we consider the AND, OR, MAX and MIN fuzzy aggregation operators. In this way, a separate FPT is induced for each class (two classes in our case). During testing, the test sample is compared with the two FPTs, and it is classified in the class that corresponds to the tree for which highest similarity is obtained. In Naumoski et al. (2012), the procedure for inducing bottom-up FPTs is described in details.

2.4.2.2. Top-down fuzzy pattern trees. In the top-down FPTs (Senge and Hüllermeier, 2011), two major changes are introduced in order to enhance the bottom-up FPTs (Huang et al., 2008). First, the direction of the model induction is inverted, and second, the stop criterion is adjusted according to the problem's complexity. With the second improvement the model induction stops when the increase of the similarity in two consecutive levels is lower than 25%, thus the complexity of the model is adapted according to the difficulty of the learning problem. Further, the top-down FPT method considers various FMFs and applies the most appropriate one.

2.4.3. Software and experimental setup

For the classical classification methods we use their implementations provided in the Weka software (Hall et al., 2009), while for the fuzzy classification methods (Huang et al., 2008; Senge and Hüllermeier, 2011) we use the implementations provided by their authors. In the model induction, we use the default settings for the parameters, and we want to mention that the test samples are not used for tuning the models' parameters.

3. Results and discussion

The test datasets are not balanced, so we must use an evaluation measure that is suitable for unbalanced datasets. We use the Area under the ROC curve (AUC-ROC) measure, which attains values in the interval [0,1]. Higher AUC-ROC corresponds to better prediction power. Also, we use the Sensitivity (True positive rate) and Specificity (True negative rate) evaluation measures in order to get better insight about the prediction power of the models.

In the first analyses, we use the balanced BIND training dataset for model induction, and B3530 test dataset for validation of the models. First, we compare the models obtained using all features and the models obtained using total ASA, average DPX, average CX and hydrophobicity. The results of this analysis provided in Table 2 show that generally when the entire set of features is considered, the prediction power of the model is decreased (except for ADTree and BayesNet). This proves that considering more features does not mean increasing the prediction power. Also, the training and testing times, as well as model's complexity increase as the number of features increases. For example, the first C4.5 tree (using all features) is induced in 54 s and has 3015 nodes, while the second C4.5 tree is induced in 11 s and has 243 nodes. The testing for all test samples using these C4.5 trees lasts 13 and 5 s respectively.

We applied the feature selection and transformation techniques explained above. Table A.2 given in the Appendix provides details regarding the selected features with each technique. Total ASA, non-polar ASA, total RASA, the three variants of CX and hydrophobicity are the features selected by most of the techniques. In Table A.3 given in the Appendix we provide details about the time required for each technique. Feature selection lasts longer when Relief technique is applied, which is expected since each sample is compared with all other samples in order to find its nearest neighbors. The selection using SVM lasts longer because for each

Table 2
AUC-ROC obtained using various classifiers and different sets of features. For the feature selection techniques that rank the features, we consider the top 4 features. The bolded values are the highest values for AUC-ROC achieved by each classifier. NA denotes that we were not able to obtain the result due to memory complexity.

Set of features	C4.5	NB	NBTree	ADTree	FTree	BayesNet
All features	0.564	0.565	0.576	**0.562**	0.582	0.579
tASA, avgDPX, avgCX, hydrophobicity	0.587	0.567	0.586	0.546	0.589	0.576
PCA	0.577	**0.574**	0.570	0.551	0.581	0.572
ChiSquared	0.568	0.565	0.572	0.560	0.576	0.575
InfoGain	0.568	0.565	0.572	0.560	0.576	0.575
GainRatio	0.564	0.560	0.566	0.552	0.578	0.572
Relief unweighted	0.582	0.554	**0.589**	0.546	0.584	**0.588**
Relief weighted	0.561	0.552	0.568	0.552	0.566	0.568
Exhaustive/Best-first/Genetic	0.583	0.567	0.583	**0.562**	0.589	0.587
Forward selection	0.577	0.567	0.571	0.552	0.578	0.579
Backward elimination	0.577	0.567	0.571	0.552	0.578	0.579
mRMR (MID)	**0.590**	0.555	0.586	0.546	**0.591**	0.586
mRMR (MIQ)	0.566	0.558	0.568	0.541	0.569	0.563
SVM	0.583	0.567	0.570	0.553	0.585	0.580
Wrapper_C4.5	0.583	0.565	0.578	0.546	0.585	**0.588**
Wrapper_ADTree	0.569	0.542	0.567	0.560	0.571	0.562
Wrapper_NB	0.568	0.565	0.570	0.561	0.573	0.573
Wrapper_NBTree	0.585	0.547	**0.589**	0.546	0.590	0.587
Wrapper_BayesNet	0.587	0.565	0.577	**0.562**	0.589	**0.588**
Wrapper_KNN	0.586	0.548	**0.589**	0.546	0.589	0.587

Set of features	Bottom-up FPT			Top-down FPT
	Triang.	Trapez.	Gauss.	
All features	NA	NA	NA	NA
tASA, avgDPX, avgCX, hydrophobicity	0.541	0.563	0.541	**0.586**
PCA	0.555	0.531	0.545	0.572
ChiSquared	**0.569**	**0.564**	0.567	0.578
InfoGain	**0.569**	**0.564**	0.567	0.578
GainRatio	0.561	0.563	0.543	0.569
Relief unweighted	0.541	0.535	0.541	0.568
Relief weighted	0.547	0.530	0.548	0.568
Exhaustive/Best-first/Genetic	NA	NA	NA	NA
Forward selection	0.567	**0.564**	0.568	0.575
Backward elimination	0.567	**0.564**	0.568	0.575
mRMR (MID)	0.554	0.549	0.556	0.560
mRMR (MIQ)	0.555	0.559	0.555	0.547
SVM	0.565	**0.564**	0.566	0.579
Wrapper_C4.5	0.541	0.562	0.542	0.569
Wrapper_ADTree	0.536	0.561	0.554	0.558
Wrapper_NB	0.557	0.562	0.558	0.564
Wrapper_NBTree	0.541	0.531	0.541	0.576
Wrapper_BayesNet	0.545	**0.564**	**0.570**	0.585
Wrapper_KNN	0.541	0.533	0.541	0.570

inspected subset, a SVM model is induced. The wrapper techniques take significantly more time than the others, since for each examined subset of features separate models are induced.

We made simulations by filtering the top 4, 8 and 10 features, and the results indicate that the additional features do not always improve the predictions. Therefore, in the next analyses, for the techniques that rank the features we consider only the top 4 features. Using C4.5, the most accurate model is obtained using the MID scheme of mRMR. Also the sets of features identified by some wrapper techniques, and the set of features that is used in previous papers are appropriate in combination with C4.5. Using InfoGain and GainRatio for estimating features' significance, we do not obtain the best C4.5 trees, even though C4.5 uses gain ratio to select the best feature in each node. This is a result of the feature selection where we rank the features and find the most optimal feature that should be examined in the root node of the tree. The top ranked features have high gain ratio at the root node, but later in lower nodes they could turn to be irrelevant since they may have high correlation with the features that are examined in the upper nodes toward the root of the tree. Using the NB classifier, lower AUC-ROC is obtained. The highest AUC-ROC using NB is achieved on the sets of features obtained by PCA, since NB presumes features' independency and PCA transforms the original correlated features into new non-correlated features. NBTree classifier achieves highest AUC-ROC using the sets of features identified by Wrapper_KNN, Relief unweighted and Wrapper_NBTree. ADTree classifier induced models with lower prediction power. We can mention that for the other classifiers using wrapper FSTs, the worst model is obtained using Wrapper_ADTree. However, using ADTree classifier, Wrapper_ADTree shows better selection performances than most of the other wrappers. FTree models generally show best prediction performances. Using the FTree classifier and considering the top 4 features identified by the MID scheme of mRMR, the highest AUC-ROC is obtained. Also, FTree obtained better results on the set of features selected by the exhaustive search, best-first search, genetic algorithm and some wrappers. Regarding BayesNet classifier, best results are obtained by using the features selected by the corresponding wrapper (Wrapper_BayesNet). Generally, the top-down FPT method induces more accurate models than the bottom-up approach. However, with the top-down approach the best results are obtained by using the set of features used in previous papers. Namely, none of the FSTs found a better set of 4 features appropriate for this method. Regarding the bottom-up approach, by using FSTs more optimal sets of features are identified than the set of four features that was used previously. Regarding mRMR, MID proved as a better scheme than MIQ. Unweighted Relief obtained better results than weighted Relief. Regarding wrapper FSTs, if we use the model induction method in the feature

selection, more powerful models are induced. The highest AUC-ROC of 0.591 is obtained using FTree classifier and the top 4 features selected by the MID scheme of mRMR. The second best model is obtained by the C4.5 classifier using the same set of features. In almost all simulations, higher precision is achieved using a subset of features instead of using the entire set of features. Generally, by using FSTs we enhanced the classifiers.

Tables A.4 and A.5 given in the Appendix present the times required for training and testing by using various classifiers and different sets of features. These tests are made on a machine with Intel Core 2 Duo CPU on 2.1 GHz and 4 GB RAM. It can be seen that the induction of ADTree, top-down FPT and NBTree models takes longer than the induction of the other models. On the other side, the testing time is longer when FTrees and bottom-up FPTs are used.

In order to get a clearer picture regarding the overall ranks of the classifiers and FSTs, we performed overall ranking using the Friedman and Quade tests (García et al., 2010). Since for the FPT based methods we induced models by using sets of 4 features, therefore only these sets are considered in the ranking. The results of the ranking are shown in Figs. 1 and 2. Lower value for the rank indicates better overall ranking. It can be seen that the set of 4 features identified by Wrapper_BayesNet has the best rank, while the set selected by the Relief weighted technique has the worst rank. Regarding classifiers, FTree has the best prediction power, while the bottom-up FPT with trapezoidal FMF is the worst classifier.

We also made tests by using different number of features in order to investigate the benefits of using feature selection techniques. In this analysis, we consider the Wrapper_BayesNet and FST based on the SVM classifier, which are the best FSTs according to the results shown on Fig. 1, while for the model induction we use the FTree and BayesNet classifiers that attained best ranks

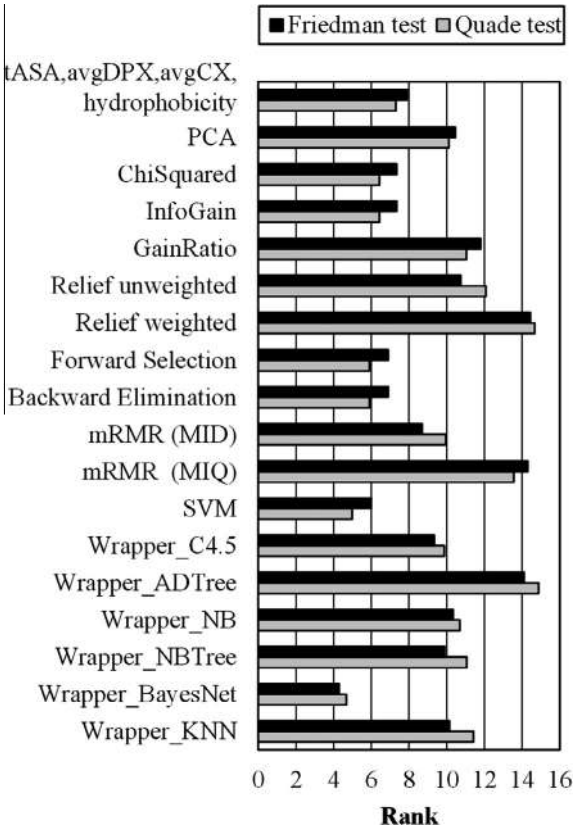

Fig.1. The ranks of the sets of four features obtained by using the Friedman and Quade tests. Lower value indicates better ranking.

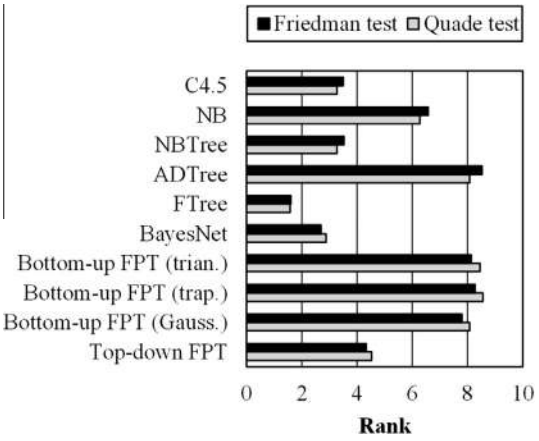

Fig.2. The ranks of the classifiers obtained by using the Friedman and Quade tests. Lower value indicates better ranking.

according to Fig. 2. We induced models by using the highly ranked features, and different tests were made by considering various number of features. In this analysis we use the B1549 test dataset for evaluating the models. Figs. A.1 and A.2 show the ratio of the training/testing time by using various number of features and the training/testing time by using all (16) features, while Fig. A.3 presents the values for AUC-ROC of the models obtained by using different number of features. These tests are made on a machine with 8 processors on 2.27 GHz and 8 GB RAM. The results show that by using the FTree and BayesNet classifiers the training and testing times linearly increase with the inclusion of additional features. Regarding the time required for testing the models, by using the FTree classifier the increase of the time required for testing the query samples significantly increases with the increase of the number of features. On the other side, the results for AUC-ROC show that with the rise of the number of features up to 4 or 6, there is an increase of the predictive performance of the models, while by using more than 6 features, generally lower values for AUC-ROC are obtained. From this, we can conclude that the optimal number of features could be between 4 and 6. The analysis showed that by applying proper feature selection technique before generating the prediction model, we can improve the model and time complexity (training and testing times), as well as the predictive performances (AUC-ROC) of the models as a result of considering the most relevant features.

In order to get better insight into the predictive performances of the models, we also analyzed the Sensitivity and Specificity of the models obtained by using various FSTs and model induction methods. This analysis is performed over B1549 test dataset by using the 4 features with highest ranks. The results of this analysis, which are presented on Fig. 3, show that by using the NBTree and BayesNet classifiers higher Sensitivity is obtained in predicting the protein binding sites as a result of the lower number of false negatives. This means that with these classifiers, higher fraction of the residues that are part of binding regions are correctly detected. For example, by using the NBTree classifier and the set of 4 features selected by the Forward selection technique more than 74% of the residues that are part of binding region are correctly identified. On the other side, with this model the Specificity is around 40.5%, which means that 40.5% of the residues that are not part of binding site are correctly identified, as a result of the higher number of false positives. By using the C4.5 and FTree classifiers higher Specificity is obtained, which means that with these methods the number of false positives is smaller. The model generated by using the FTree classifier and the set of features selected by the MID scheme of mRMR has Sensitivity of 49.3% and Specificity

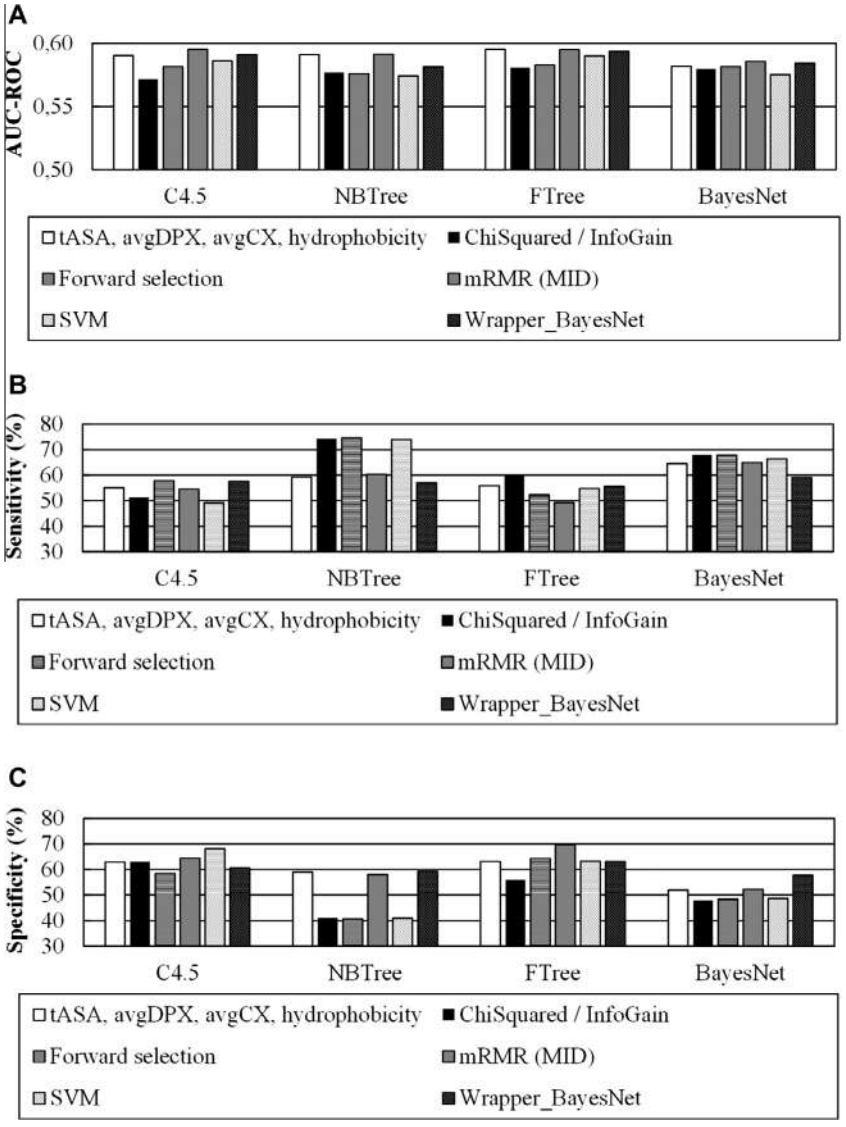

Fig.3. The results obtained using various classifiers and different sets of features: (a) AUC-ROC, (b) Sensitivity and (c) Specificity.

of 69.7%. Since we are interested in predicting the protein binding sites, the best option is to use a model that attain higher Sensitivity in order to find the candidate residues that form binding regions, and then to verify these candidates by using some other more sophisticated methods.

Next, we compare our approach with several existing methods for protein binding sites prediction. We consider distance-based (Mihel et al., 2008; Ofran and Rost, 2003; PRINT, 2013) and conservation based methods (Aytuna et al., 2005; Capra and Singh, 2007; Jones and Thornton, 1997), as well as pocket finding methods (An et al., 2005; Hendlich et al., 1997; Laskowski, 1995). Also, we make tests by using the ConCavity method (Capra et al., 2009) that combines some pocket finding algorithm with the Jensen-Shannon divergence (JSD) method for estimating sequence conservation (Capra and Singh, 2007). Additionally, we consider the PRISM method (Keskin et al., 2008) that identifies the binding sites of a given query by structural matching with the template structures whose binding sites are determined based on sequence and structure conservation (Aytuna et al., 2005). Regarding our approach, we consider the FSTs and classification methods that are better ranked in the analysis performed above. In Table 3 we present the results for AUC-ROC, Sensitivity and Specificity obtained on B1549 and L213 test datasets.

Our approach shows better performances than the JSD sequence conservation-based method (Capra and Singh, 2007). The results show that the examined existing methods behave differently on various datasets. The existing distance-based and conservation-based methods, except JSD, achieved high Sensitivity and high Specificity for predicting the binding sites stored in the BIND database, and low Sensitivity and high Specificity on the LigASite database. On the other side, the methods that are based on pocket finding obtained high Sensitivity and high Specificity on the LigASite database and very low Specificity on the BIND database. This fact shows that the former methods accurately predict the binding sites from the BIND database, while the latter methods make better predictions of the binding sites in the LigASite database. This is because these methods are concentrated on a specific group of interactions, and thus are appropriate for one group of proteins, but not in general. On the other side, our approach attained comparable results on all datasets and thus proved as very stable. Our approach is general, since we use training dataset from various proteins without focusing on specific interactions.

Table 3
The results for AUC-ROC, Sensitivity and Specificity obtained using various methods. NA denotes that we do not have predictions on this dataset. Type 3DM denotes 3D matching; D denotes distance-based; C denotes conservation-based and P denotes pocket finding.

Method	Type	Reference
Our approach		This
PRISM	3DM	Keskin et al. (2008)
Atom nucleus distance	D	Ofran and Rost (2003)
PIADA	D	Mihel et al. (2008)
PRINT	D	PRINT (2013)
ASA change	C	Jones and Thornton (1997)
Van der Waals distance	C	Aytuna et al. (2005)
JSD	C	Capra and Singh (2007)
LigSite	P	Hendlich et al. (1997)
PocketFinder	P	An et al. (2005)
Surfnet	P	Laskowski (1995)
ConCavity LigSite	C + P	Capra et al. (2009)
ConCavity PocketFinder	C + P	Capra et al. (2009)
ConCavity Surfnet	C + P	Capra et al. (2009)

Test dataset	B1549			L213		
Method	AUC-ROC	Sensitivity	Specificity	AUC-ROC	Sensitivity	Specificity
Our approach	0.595	0.546	0.645	0.619	0.457	0.782
PRISM	0.787	0.878	0.695	0.538	0.372	0.705
Atom nucleus distance	0.833	0.845	0.821	0.530	0.295	0.765
PIADA	0.832	0.848	0.816	0.531	0.300	0.763
PRINT	0.775	0.926	0.623	0.548	0.476	0.619
ASA change	0.820	0.864	0.776	0.543	0.353	0.732
Van der Waals distance	0.840	0.822	0.859	0.525	0.256	0.795
JSD	0.537	0.496	0.578	0.608	0.679	0.538
LigSite	NA	NA	NA	0.744	0.664	0.824
PocketFinder	NA	NA	NA	0.773	0.796	0.751
Surfnet	NA	NA	NA	0.741	0.731	0.751
ConCavity LigSite	0.587	0.657	0.518	0.800	0.781	0.820
ConCavity PocketFinder	NA	NA	NA	0.809	0.771	0.847
ConCavity Surfnet	NA	NA	NA	0.791	0.676	0.907

Our approach could attain better prediction power if it is used for building models for specific group of proteins and interactions. In this way, our approach is self-adaptable, since in the induction of the model for particular group of proteins/interactions, the FST will filter the most significant features that are important for that specific group. On the other side, the examined existing methods could not be adapted for inducing separate models that are adjusted for various proteins. In order to prove this, we made models for predicting the binding sites of the protein chains that belong to the fold TIM beta/alpha-barrel (51,350) in the SCOP hierarchy, which is the fold with highest number of chains in L213 test dataset. For this purpose we use 2.03 version of the SCOP database (Murzin et al., 1995). The test dataset is formed by considering the 3752 residues of the 17 protein chains from L213 dataset that belong to the inspected fold (51,350), while the training dataset is generated from the 2934 residues of the remaining 14 protein chains from the non-redundant LigASite set that belong to the same fold. The training dataset is balanced by down sampling the set to 20% of its original size without replacement of the samples and by following uniform distribution of the class attribute. After balancing, the features are normalized in the interval [0,1]. The predictive performances of the best model are: AUC-ROC = 0.716, Sensitivity = 61.80% and Specificity = 81.41%. From this, it is evident that if the proposed approach is used on a more specific group of proteins, then the predictive power improves. Moreover, by the usage of FST, the most relevant features are identified for the inspected group of proteins.

Finally, we made an analysis of the predictions for the protein chains from SCOP fold TIM beta/alpha-barrel (51,350) whose functions are not discovered yet according to the Gene Ontology annotations from 10 July 2014. The results presented in Tables A.6 and A.7 in the Appendix show that for these protein chains (1OF8 chain A and 1OF8 chain B) there are fewer false negatives and false positives than by using the existing methods. From these results we can conclude that the residues that are incorrectly classified as negatives with our approach are also misclassified with the existing methods. In future, by analyzing the parts of the protein structure where false negatives occur, the disadvantages of the methods can be identified in order to improve them. Also, these predictions potentially could be used for determination of the functions of these protein chains.

4. Conclusion

In our previous work, a set of four features of the amino acid residues was considered for predicting the protein binding sites. However, computational selection of the most relevant features was not performed. In order to consider the most significant features, in this paper we applied various feature selection and transformation techniques before the model induction. Generally the techniques that evaluate subsets of features appeared to be a better choice, since they consider both the relevance and the redundancy of the features. The wrapper techniques showed better performances when the corresponding induction method is considered. The results showed that the predictive power and time complexity are improved when an appropriate FST is used to select the optimal subset of features. By using the Friedman and Quade tests, we also performed ranking of the FSTs and classification methods in order to get better picture about their overall prediction power. The results show that Wrapper_BayesNet selects the most optimal subset of features, while the FTrees classifier achieves best prediction power.

Additionally, we compared our approach with several existing methods for protein binding sites prediction. The results show that the examined existing methods behave in different manner on different sets of proteins, while our approach is stable. This means that the existing methods are suitable for making predictions for

a given group of proteins, but are not able to make general decisions. Furthermore, the examined existing methods are not adaptable, meaning that they can be used only for particular set of proteins. Our models are trained using various proteins, but if the models are focused on a specific group of proteins, then the prediction power will be much better. In order to prove this, we generated a model for predicting the binding sites of the protein chains from the SCOP fold TIM beta/alpha-barrel (51,350) and the results showed that our approach attains significantly better results if it is applied on a specific group of proteins. Our approach provide self-adaptability since by using FST the most relevant features for the particular group of proteins are automatically chosen. Additionally, our approach could be used to build a general prediction model that is a cascade of separate sub-models, where each sub-model will self-adapt for the corresponding types of proteins.

We identified several directions for further improvements. We plan to include additional amino acid residues' features and to apply additional FSTs. Regarding classifiers, we would continue searching for other classification methods that may provide more powerful models. Also, we will try to build a general cascade prediction model, which is a cascade of sub-models that are self-adapted to cover specific groups of proteins.

Acknowledgments

This work was partially financed by the Faculty of Computer Science and Engineering at the "Ss. Cyril and Methodius University in Skopje", Skopje, R. Macedonia.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.jsb.2014.11.007.

References

Abdi, H., Williams, L.J., 2010. Principal component analysis. Wiley Interdiscip. Rev. Comput. Stat. 2 (4), 433–459.
Aha, D.W., Kibler, D., Albert, M.K., 1991. Instance-based learning algorithms. Mach. Learn. 6, 37–66.
Altschul, S., Gish, W., Miller, W., Myers, E.W., Lipman, D., 1990. Basic local alignment search tool. J. Mol. Biol. 215 (3), 403–410.
An, J., Totrov, M., Abagyan, R., 2005. Pocketome via comprehensive identification and classification of ligand binding envelopes. Mol. Cell. Proteomics 4 (6), 752–761.
Aytuna, A.S., Gursoy, A., Keskin, O., 2005. Prediction of protein–protein interactions by combining structure and sequence conservation in protein interfaces. Bioinformatics 21 (12), 2850–2855.
Bader, G.D., Donaldson, I., Wolting, C., Ouellette, B.F.F., Pawson, T., Hogue, C.W.V., 2001. BIND: the biomolecular interaction network database. Nucleic Acids Res. 29 (1), 242–245.
Capra, J.A., Singh, M., 2007. Predicting functionally important residues from sequence conservation. Bioinformatics 23 (15), 1875–1882.
Capra, J.A., Laskowski, R.A., Thornton, J.M., Singh, M., Funkhouser, T.A., 2009. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. PLoS Comput. Biol. 5 (12).
Chandonia, J.M., Hon, G., Walker, N.S., Lo Conte, L., Koehl, P., Levitt, M., Brenner, S.E., 2004. The ASTRAL compendium in 2004. Nucleic Acids Res. 32 (Database issue), D189–D192.
Chothia, C., 1976. The nature of the accessible and buried surfaces in proteins. J. Mol. Biol. 105 (1), 1–12.
Dessailly, B.H., Lensink, M.F., Orengo, C.A., Wodak, S.J., 2008. LigASite a database of biologically relevant binding sites in proteins with known apo-structures. Nucleic Acids Res. 36 (Database issue), D667–D673.
Ding, C., Peng, H., 2005. Minimum redundancy feature selection from microarray gene expression data. J. Bioinf. Comput. Biol. 3 (2), 185–205.
Du Plessis, L., Škunca, N., Dessimoz, C., 2011. The what, where, how and why of gene ontology-a primer for bioinformaticians. Briefings Bioinf. 12 (6), 723–735.
Fayyad, U.M., Irani, K.B., 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In: Bajcsy, R. (Ed.), Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI 1993). Morgan Kaufmann, San Francisco, CA, USA, pp. 1022–1027.
Freund, Y., Mason, L., 1999. The alternating decision tree learning algorithm. In: Bratko, I., Dzeroski, S. (Eds.), Proceedings of the 16th International Conference on Machine Learning (ICML 1999). Morgan Kaufmann, San Francisco, CA, USA, pp. 124–133.
Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian network classifiers. Mach. Learn. 29 (2–3), 131–163.
Gama, J., 2004. Functional trees. Mach. Learn. 55 (3), 219–250.
García, S., Fernández, A., Luengo, J., Herrera, F., 2010. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf. Sci. 180 (10), 2044–2064.
Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, first ed. Addison-Wesley, Boston, MA, USA.
Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Mach. Learn. 46 (1–3), 389–422.
Hall, M.A., 1999. Correlation-based Feature Selection for Machine Learning (Ph.D. thesis). University of Waikato, Hamilton, New Zealand.
Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The WEKA data mining software: an update. SIGKDD Explor. 11 (1), 10–18.
Hendlich, M., Rippmann, F., Barnickel, G., 1997. LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. J. Mol. Graph. Model. 15 (6), 359–363.
Holm, L., Sander, C., 1993. Protein structure comparison by alignment of distance matrices. J. Mol. Biol. 233 (1), 123–138.
Huang, Z.H., Gedeon, T.T.D., Nikravesh, M., 2008. Pattern trees induction: a new machine learning method. IEEE Trans. Fuzzy Syst. 16 (4), 958–970.
Hubbard, S.J., Thornton, J.M., 1993. NACCESS, Computer Program. Department of Biochemistry and Molecular Biology, University College London, London, London, UK.
Hunt, E.B., Martin, J., Stone, P., 1966. Experiments in Induction, first ed. Academic Press, New York, USA.
John, G.H., Langley, P., 1995. Estimating continuous distributions in bayesian classifiers. In: Besnard, P., Hanks, S. (Eds.), Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Francisco, CA, USA, pp. 338–345.
Jones, S., Thornton, J.M., 1997. Analysis of protein–protein interaction sites using surface patches. J. Mol. Biol. 272 (1), 121–132.
Keskin, O., Nussinov, R., Gursoy, A., 2008. PRISM: protein–protein interaction prediction by structural matching. Methods Mol. Biol. 484, 505–521.
Kira, K., Rendell, L.A., 1992. The feature selection problem: traditional methods and a new algorithm. In: Swartout, W.R. (Ed.), Proceedings of the 10th National Conference on Artificial Intelligence (AAAI 1992). AAAI Press, Menlo Park, CA, USA, pp. 129–134.
Kohavi, R., 1996. Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In: Simoudis, E., Han, J., Fayyad, U. (Eds.), Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD 1996). AAAI Press, Menlo Park, CA, USA, pp. 202–207.
Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artif. Intell. 97 (1), 273–324.
Kononenko, I., 1994. Estimating attributes: analysis and extensions of RELIEF. In: Bergadano, F., De Raedt, L. (Eds.), Proceedings of the European Conference on Machine Learning (ECML 1994). Springer-Verlag New York, Secaucus, NJ, USA, pp. 171–182.
Kyte, J., Doolittle, R.F., 1982. A simple method for displaying the hydropathic character of a protein. J. Mol. Biol. 157 (1), 105–132.
Laskowski, R., 1995. SURFNET: a program for visualizing molecular surfaces, cavities and intermolecular interactions. J. Mol. Graph. 13 (5), 323–330.
Lee, B., Richards, F.M., 1971. The interpretation of protein structures: estimation of static accessibility. J. Mol. Biol. 55 (3), 379–400.
Liu, H., Setiono, R., 1995. Chi2: feature selection and discretization of numeric attributes. In: Vassilopoulos, J.F. (Ed.), Proceedings of the 7th IEEE International Conference on Tools with Artificial Intelligence. IEEE Computer Society, USA, pp. 388–391.
Liu, L., Lu, W.C., Cai, Y.D., Feng, K.Y., Peng, C., Zhu, Y., 2013. Prediction of protein–protein interactions based on feature selection and data balancing. Protein Pept. Lett. 20 (3), 336–345.
Lu, Y., Wang, X., Chen, X., Zhao, G., 2013. Computational methods for DNA-binding protein and binding residue prediction. Protein Pept. Lett. 20 (3), 346–351.
Mihel, J., Šikić, M., Tomić, S., Jeren, B., Vlahoviček, K., 2008. PSAIA – protein structure and interaction analyzer. BMC Struct. Biol. 8, 21.
Mirceva, G., Kulakov, A., 2012a. Fuzzy pattern trees for predicting protein binding sites. In: Bakeva, V., Gjorjevikj, D., (Eds.), Proceedings of the 9th Conference for Informatics and Information Technology (CIIT 2012), pp. 96–100.
Mirceva, G., Kulakov, A., 2012b. Top-down approach for protein binding sites prediction based on fuzzy pattern trees. In: Markovski, S., Gusev, M., (Eds.), ICT Innovations 2012, Proceeding of the ICT Innovations 2012, Springer-Verlag, Berlin Heidelberg, Germany, AISC, 207, pp. 325–334.
Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C., 1995. Scop: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. 247 (4), 536–540.
Naumoski, A., Mirceva, G., Mitreski, K., 2012. A novel fuzzy based approach for inducing diatom habitat models and discovering diatom indicating properties. Ecol. Inf. 7 (1), 62–70.
Niu, S., Huang, T., Feng, K.Y., He, Z., Cui, W., Gu, L., Li, H., Cai, Y.D., Li, Y., 2013. Inter- and intra-chain disulfide bond prediction based on optimal feature selection. Protein Pept. Lett. 20 (3), 324–335.
Ofran, Y., Rost, B., 2003. Predicted protein–protein interaction sites from local sequence information. FEBS Lett. 544 (1–3), 236–239.

Panchenko, A.R., Kondrashov, F., Bryant, S., 2004. Prediction of functional sites by analysis of sequence and structure conservation. Protein Sci. 13 (4), 884–892.
Pearl, J., 1984. Heuristics: Intelligent Search Strategies for Computer Problem Solving, first ed. Addison-Wesley, Boston, MA, USA.
Pearson, K., 1901. On lines and planes of closest fit to systems of points in space. Philos. Mag. 2 (11), 559–572.
Peng, H.C., Long, F., Ding, C., 2005. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. 27 (8), 1226–1238.
Pintar, A., Carugo, O., Pongor, S., 2002. CX, an algorithm that identifies protruding atoms in proteins. Bioinformatics 8 (7), 980–984.
Pintar, A., Carugo, O., Pongor, S., 2003. DPX: for the analysis of the protein core. Bioinformatics 19 (2), 313–314.
PRINT: Dataset of PRotein Protein INTerfaces. <http://prism.ccbb.ku.edu.tr/interface> (accessed 08.08.13).
Quinlan, R., 1993. C4.5: Programs for Machine Learning, first ed. Morgan Kaufmann Publishers, San Mateo, CA, USA.
Senge, R., Hüllermeier, E., 2011. Top-down induction of fuzzy pattern trees. IEEE Trans. Fuzzy Syst. 19 (2), 241–252.
Sharan, R., Ulitsky, I., Shamir, R., 2007. Network-based prediction of protein function. Mol. Syst. Biol. 3, 88.
Shindyalov, H.N., Bourne, P.E., 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng. 11 (9), 739–747.
Shrake, A., Rupley, J.A., 1973. Environment and exposure to solvent of protein atoms. Lysozyme and insulin. J. Mol. Biol. 79 (2), 351–371.
Sigrist, C.J.A., Cerutti, L., De Castro, E., Langendijk-Genevaux, P.S., Bulliard, V., Bairoch, A., Hulo, N., 2010. PROSITE, a protein domain database for functional characterization and annotation. Nucleic Acids Res. 38 (Database issue), D161–D166.
The Gene Ontology Consortium, 2008. The gene ontology project 2008. Nucleic Acids Res. 36 (Database issue), D440–D444.
Tuncbag, N., Kar, G., Keskin, O., Gursoy, A., Nussinov, R., 2009. A survey of available tools and web servers for analysis of protein–protein interactions and interfaces. Briefings Bioinf. 10 (3), 217–232.
Ye, Y., Godzik, A., 2004. FATCAT: a web server for flexible structure comparison and structure similarity searching. Nucleic Acids Res. 32 (Web Server issue), W582–W585.