Molecular Biosystems. 2014; 10: 1063-1071

Improving enzyme regulatory protein classification by means of SVM-RFE feature selection

Carlos Fernandez-Lozano[a], Enrique Fernández-Blanco[a], Kirtan Dave[b], Nieves Pedreira[a], Marcos Gestal[a], Julián Dorado[a] and Cristian R. Munteanu[a]

[a] *Computer Science Faculty, Dept. of Information and Communication Technologies, University of A Coruña, Campus de Elviña s/n, 15071 A Coruña, Spain. E-mail: carlos.fernandez@udc.es; Fax: +34-981167160; Tel: +34-981167000, Ext. 1302*
[b] *G.H. Patel P.G. Dept. of Computer Science and Technology, Sardar Patel University, Vallabh Vidyanagar, Gujarat, India*

Abstract
Enzyme regulation proteins are very important due to their involvement in many biological processes that sustain life. The complexity of these proteins, the impossibility of identifying direct quantification molecular properties associated with the regulation of enzymatic activities, and their structural diversity creates the necessity for new theoretical methods that can predict the enzyme regulatory function of new proteins. The current work presents the first classification model that predicts protein enzyme regulators using the Markov mean properties. These protein descriptors encode the topological information of the amino acid into contact networks based on amino acid distances and physicochemical properties. MInD-Prot software calculated these molecular descriptors for 2415 protein chains (350 enzyme regulators) using five atom physicochemical properties (Mulliken electronegativity, Kang–Jhon polarizability, vdW area, atom contribution to P) and the protein 3D regions. The best classification models to predict enzyme regulators have been obtained with machine learning algorithms from Weka using 18 features. K* has been demonstrated to be the most accurate algorithm for this protein function classification. Wrapper Subset Evaluator and SVM-RFE approaches were used to perform a feature subset selection with the best results obtained from SVM-RFE. Classification performance employing all the available features can be reached using only the 8 most relevant features selected by SVM-RFE. Thus, the current work has demonstrated the possibility of predicting new molecular targets involved in enzyme regulation using fast theoretical algorithms.

Introduction

Enzymes are large biological molecules responsible for the thousands of chemical interconversions that sustain life.[1,2] This paper is focused on enzymes, which are proteins with a significant influence on metabolic reactions. Usually, the influence on those metabolic reactions is reflected in a great accelerating rate and specificity of reactions.

This enzyme influence is very important for life reactions, for example, the same reactions without enzymes are among the slowest that have ever been measured, some with half-times approaching the age of the earth. Enzymes are needed in most chemical reactions in a biological cell, and should occur at rates sufficient for life; thus, this difference provides a measure of the importance and proficiencies of enzymes as catalysts and their relative susceptibilities to inhibition by transition-state analogue inhibitors.[3]

Furthermore the set of enzymes made in a cell determines which metabolic pathways occur in that cell. It is important to note that only a small portion of the enzyme (less than 4 amino acids) is related to the catalysis in a direct way.[4] The region that contains these catalytic residues, binds the substrate and then carries out the reaction is known as the active site.

The enzymes that catalyse chemical reactions are regulated enzymes. Some examples of enzyme regulators are cyclase regulators, enzyme activators, enzyme inhibitors and kinase regulators.

The experimental method of characterizing the proteins that act as enzyme regulators is expensive and time-consuming, and is impossible to apply to test thousand of proteins with other functions or without any known function. Therefore, theoretical methods are useful to predict protein functions such as enzyme regulation. In order to create quantitative prediction models for a specific function of proteins, the molecular information should be encoded into specific numbers (molecular descriptors) using any type of information available such as molecular topology, 3D protein conformation, and atom/amino acid physicochemical properties. These numbers are unique for a specific protein and they can be used to obtain classification models using machine learning techniques.

Examples in this regard are the molecular descriptors based on electrostatic potential that have been used to predict enzyme class,[5] DNA-cleavage protein activity,[6] protein–protein interactions in parasites,[7,8] drug–protein interactions[9] or lipid-binding proteins.[10] The classifier represents a Quantitative Structure–Activity Relationship (QSAR)[11] between the protein 3D structure and the biological activity/function. The QSAR models[12] for drugs have been intensively used for a large spectrum of studies such as target searching,[13–15] and antifungal,[16] antiviral[17] and antimalarial[18] activity. Other types of QSAR models used the structures of peptides,[19] proteins[20] and DNA promoters.[21]

The aim of this study is to demonstrate the possibility of encoding protein chain 3D structure information into new molecular descriptors using molecular topology and atom physicochemical properties in order to obtain QSAR classification models that can predict enzyme regulatory function for new peptides.

Materials and methods

Fig. 1 shows the steps performed to predict proteins related to the enzyme regulation process.

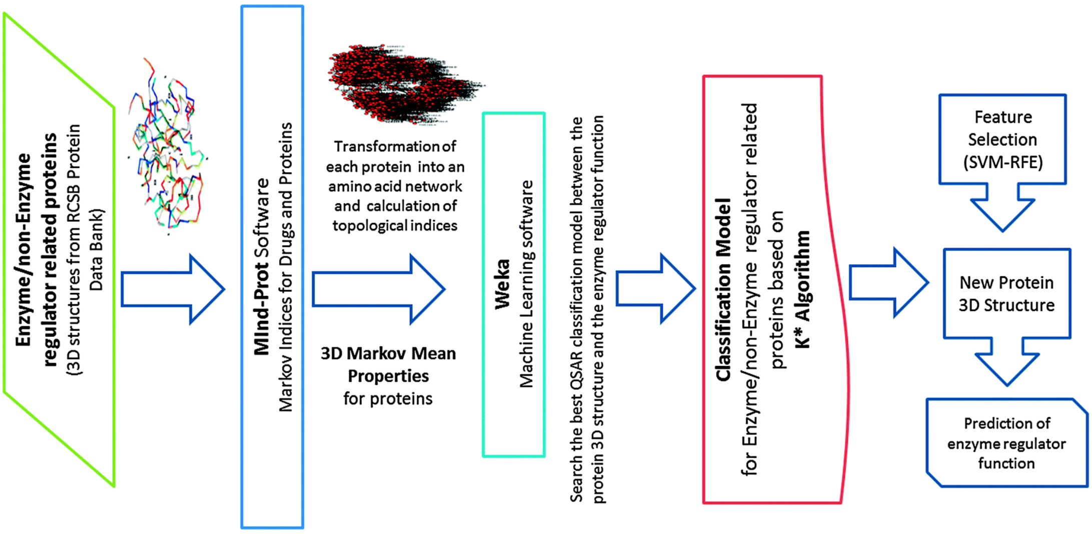

Enzyme/non-Enzyme regulator related proteins
(3D structures from RCSB Protein Data Bank)
MInd-Prot Software
Markov Indices for Drugs and Proteins
Transformation of each protein into an amino acid network and calculation of topological indices
3D Markov Mean Properties for proteins
Weka
Machine Learning software
Search the best QSAR classification model between the protein 3D structure and the enzyme regulator function
Classification Model
for Enzyme/non-Enzyme regulator related proteins based on
K* Algorithm
Feature Selection (SVM-RFE)
New Protein 3D Structure
Prediction of enzyme regulator function

Fig. 1 Flow chart of the methodology using 3D Markov mean properties for proteins and machine learning technologies to obtain an enzyme regulator/non-enzyme regulator related classification model.

In the first step, the dataset was created from two well-known classes of protein chains: one of them related to enzyme regulatory function and the other not related to this function. These protein chains can be checked at the Gene Ontology[22] website GO:0030234.

The MInd-Prot[23] tool turns this information into molecular descriptions that will be used in the next step as inputs in the Weka[24] suite. This suite includes several machine learning algorithms[25] for solving data mining problems related to different fields such as SNPs,[26] gene identification,[27] phenotype–genotype mapping,[28] microarrays,[29] and so on. These Weka algorithms will be used to search for the best classification method to classify a new 3D protein structure related to the enzyme regulation function. Finally, this classification model will allow to get a QSAR[30] model from the new 3D structure and the enzyme regulation function.

Protein set

To obtain the protein 3D structure, the authors have used two different databases containing a total number of 2415 samples: out of those, 350 samples correspond to proteins whose chains are identified as enzyme regulators (positive group) and 2065 chains are protein chains non-related to any enzyme regulatory function (negative group). The PDBs for the positive group have been downloaded from the Protein Databank,[31] the "Enzyme Regulator" list (GO ID30234[22]) obtained with the "Molecular Function Browser (GO)" in the "Advanced Search Interface" (protein identity cutoff = 30%, downloaded on November 13th, 2012).

The negative chain group was selected from the PISCES CulledPDB[32] list of proteins. This database was downloaded on November 16th, 2012 from http://dunbrack.fccc.edu/PISCES.php. In order to generate a better set, the selection was made with an identity of less than 20% (similarity among two different sequences), resolution of 1.6 Å and R-factor of 0.25. These parameters in the selection guarantee that the proteins do not have any other possible biological function. The PDB files of the negative groups were downloaded from the same Protein Databank. The protein chains from the positive group that were present in the negative group were eliminated from the latter.

Therefore, the dataset is composed of two different subsets, one with a biological function related to enzyme regulation and another without this function. Using these data, the authors have developed a classification model for this property, explained in the next section. Those protein chains are characterised by numerical properties that will be used in the classification model (features). This conversion from a 3D protein structure to molecular descriptors was carried out with the MInD-Prot tool.

Markov mean properties

MInD-Prot[23] is a Python software for the calculation of the mean properties of the Markov indices (molecular descriptors) for drugs (simple/medium molecules) and proteins (macromolecules). The inputs for this tool are the PDB/FASTA files for proteins and the SMILE codes for drugs. In the current study, the protein chains have been turned into contact networks by using the information from the 3D coordinates of the amino acids (PDB files). Therefore, the nodes are the alpha-carbon atoms of each amino acid and the links are defined by a cutoff geometrical distance.

The tool is able to calculate Markov Mean Properties (MP) using different molecule physicochemical properties in order to encode specific molecular information in addition to the topology. The algorithm is a modification of the Markov Chains (MC) method, called MARCH-INSIDE (MI), introduced by Gonzalez-Diaz et al.[33–35] The node weights for each amino acid, such as the physicochemical properties, are calculated as a sum of all atomic properties from each type of amino acid. Thus, the tool uses four types of properties such as Mulliken Electronegativity (EM), Kang–Jhon Polarizability (PKJ), van der Waals area (vdWA)[36] and Atom Contribution to P (AC2P).[37]

The contact network representation of a protein chain is a static model with the amino acids distributed spatially, with specific 3D Cartesian coordinates (x_i, y_i, z_i) for each Cα atoms (the network nodes). A Euclidean distance cutoff (r_{off}) of 7 Å between two Cα atoms is used to obtain the amino acid contact network for each protein chain. Thus, all the amino acids at an Euclidean distance less than r_{off} are connected ($\alpha_{ij} = 1$ elements in the connectivity matrix A). Each amino acid has a different contribution to interactions with other molecules for the enzyme regulation function and depends on the type of the amino acid and the 3D position. Therefore, the 3D structure of the protein is virtually divided into spherical spatial regions (R): core (c), inner (i), middle (m) and surface (s). Each region is calculated as a percentage of the longest distance r_{max} with respect to the protein chain geometrical center: c between 0% and 25%, i between 25% and 50%, m between 50% and 75%, and s between 75% and 100%. The total region (t) is considered as the entire protein chain space (0% to 100%). The tool uses the Markov Chain theory to calculate the probabilities of interaction between any two amino acids placed at a topological distance k (0–5). These values are averaged by all k values for each region R. Thus, it is possible to calculate k-averaged parameters (MP_R) for the amino acids contained in a specific region (R = c, i, m, s, t)[38–42] and for a specific physicochemical property.

The following algorithm is used to calculate the indices for each physicochemical property:

Calculation of the squared connectivity matrix of Cα atoms (A) using the 3D coordinates from PDBs; $n \times n$ matrix, n = the number of amino acids in the protein chain, α_{ij} = elements with values of 1 for connected amino acid pairs and 0 for the non-connected ones.

Calculation of the weighted matrix (W) by adding the values of the physicochemical property for each type of connected amino acid (w_j elements from vector w as amino acid weight vector).

Calculation of the interaction probability matrix (${}^{1}\Pi$) obtained by the normalisation of W.

Calculation of similar interaction probability matrices (${}^{k}\Pi$) for other k steps of interactions (k = 0–5), for a specific molecular property.

The matrices ${}^{k}\Pi$ are used to calculate the 3D Markov mean properties corresponding to the entire protein chain, ${}^{k}MP_t$, for a specific k (see eqn (1)); the central matrix ${}^{k}\Pi$ is multiplied from the left by the probability vector ${}^{0}p$ for all amino acids without considering the network connectivity; the result is multiplied from the right by the vector of the amino acid weights (w); the values correspond to elements from 1 to n.

The MPs corresponding to a specific 3D region R (c, i, m, s) are obtained using the same formula by multiplying only the values that correspond to the amino acids in an R.

In the final step, ${}^{k}\mathrm{MP}_{\mathrm{R}}$ are averaged for all k values resulting in Markov Mean Properties MP_{R} (see eqn (2)).

In conclusion, MInD-Prot[23] calculates for each protein chain 20 molecular descriptors MP_{R} that correspond to 4 types of physicochemical properties, and averaged for all the k values into 5 regions R: EM_{R}, $\mathrm{PKJ}_{\mathrm{R}}$, $\mathrm{vdWA}_{\mathrm{R}}$ and $\mathrm{AC2P}_{\mathrm{R}}$.

$$
\begin{aligned}
{}^{k}\mathrm{MP}_{\mathrm{t}} &= \left[{}^{0}p(w_1)\, {}^{0}p(w_2) \ldots {}^{0}p(w_n) \right] \cdot \begin{bmatrix} {}^{1}p_{1,1} & {}^{1}p_{1,2} & \cdot & {}^{1}p_{1,n} \\ {}^{1}p_{2,1} & {}^{1}p_{2,2} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ {}^{1}p_{1,n} & {}^{1}p_{n,2} & \cdot & {}^{1}p_{n,n} \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ w_n \end{bmatrix} \\
&= \sum_{j=1}^{n} {}^{k}p(w_j)\, w w_j
\end{aligned}
\tag{1}
$$

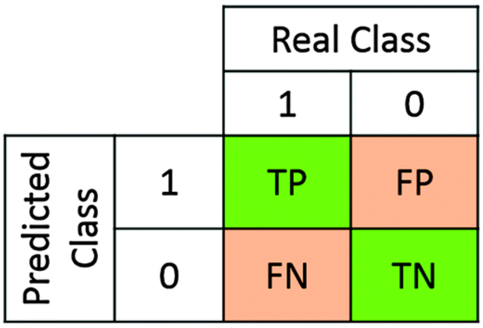
$$
\mathrm{MP}_{\mathrm{R}} = \sum_{k=0}^{5} {}^{k}\mathrm{MP}_{\mathrm{R}} \tag{2}
$$

The 3D structure and physicochemical property information of the protein chains encoded into these indices were used as input for the machine learning methods from Weka[24] in order to find the best QSAR classification model that can predict the enzyme regulatory protein chains. Additional information about the transformation of the sequence database into molecular descriptors and the input for the classification models can be found as ESI.†

Classification methods

With the aim to minimize influence of the configuration of training and validation dataset, the authors have applied in this paper the different classification methods using the well-known 10-fold cross-validation technique to split data.[43] This technique splits the dataset into 10 random equal-size subsets, 9 of which are chosen 10 times to train the models and the remaining set is used to test them. Notice that, each time, a random subset is chosen to be the test set.

The result from the test phase for a two-class problem is usually presented by using a confusion matrix (see Fig. 2), which provides a good number of different measures that help to understand the results of the classification method based on a comparison between the class provided by the classificatory model with the real or actual class: true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN).



		Real Class	
		1	0
Predicted Class	1	TP	FP
	0	FN	TN

Fig. 2 Confusion matrix.

Some of the most commonly used measures within the machine learning field are accuracy, precision and recall. The first one, accuracy (eqn (3)), establishes the percentage of correctly labelled samples, the precision (eqn (4)) is the fraction of all samples positive-labelled that really are positive, while recall (eqn (5)) is the fraction of all positive samples that have been detected by the classification model.

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN) \tag{3}$$

$$\text{Precision} = TP/(TP + FP) \tag{4}$$

$$\text{Recall} = TP/(TP + FN) \tag{5}$$

These metrics present major drawbacks when the dataset is not balanced and there is a more representative class than the other. In these situations with very skewed classes the use of metrics like F-score (eqn (6)) should be better.[45]

$$\text{F\_1 Score} = 2(\text{precision}\cdot\text{recall})/(\text{precision} + \text{recall}) \tag{6}$$

The Receiver Operating Characteristic (ROC) curve is a comparison between two operating characteristics, usually a true positive rate and a false positive rate as the criterion changes. The ROC curve is a plot that represents the performance of a binary classifier as its discrimination threshold is varied. Accuracy and ROC measurements help to select better models and discard the worst ones independently of the cost context or the class distribution. Thus, the analysis of the accuracy and ROC are an estimation of cost-benefit of diagnostic decision-making and allow an easy comparison between different models.

The Area Under Receiver Operating Characteristic Curve (AUROC)[44] is one of the most commonly used ways to measure the performance of a diagnostic test: the larger the area (closer to 1), the more accurate the diagnostic test is.

All those measurements were used as a tool to compare the different techniques that the authors have compared within the frame of this work in order to perform the main task, which is to develop a model that can distinguish enzyme regulatory proteins among SVM-RFE; this model was first presented by Guyon et al.[62] in order to select genes within a cancer classification problem. The method ranks all features within the original data according to some score function that will be assigned by means of a training set of a SVM with a linear kernel. Subsequently, it uses that score to drop the feature (or features) with the lowest scores. Thus it provides information about what are the most relevant features within a dataset (those that remain included in the feature set). This elimination process is repeated until the best classification accuracy is reached.

The authors have performed several experiments in order to select the best models. The classification implementations used in those tests were the ones included in the well-known machine learning library Weka.[24] More specifically, the authors have used: AdaBoost (AB),[46] MultiLayer Perceptron (MLP),[47,48] Naïve Bayes (NB),[49] Random Forest (RF),[50] J48[45] (the Weka implementation of c4.5 algorithm) and K*. Among all these classifiers, K* [51] was chosen because of the reasons presented in the next point.

The K* algorithm attempts to offer an efficient approach in problems that deal with missing values, real valued features or symbolic features. K* is included in Weka's instance-based learning algorithms, which are intended to make the most of some of the benefits of the use of entropy as a distance measure instead of the traditional Euclidean distance used within traditional Instance Based Learning[52] algorithms.

The traditional Instance Learning-based methods make comparisons between test samples and a previously annotated database of instances. In order to perform this comparison, the algorithm needs to use a similarity function, since one of the most simple ways to address this comparison is the use of nearest neighbour algorithms,[53] which usually are based on the Euclidean distance between the test instance and the annotated samples (the test instance will be labelled as the class of the closest annotated sample). Another option is performing the comparison between one sample and a subset of the k nearest samples of the training set, such as the KNN algorithm (in this case, a particular instance will be labelled with the most common class among this subset). By using this kind of comparison, some previous work provides correct classifications using noisy data or is able to manage either non-relevant or symbolic values.[52,54–56]

As previously stated, the K* algorithm differs from this kind of instance-based algorithms because it does not use an Euclidean measure approach: K* uses an entropy-based distance function, extracted from the information theory,[57,58] in order to compute the similarity between two different samples. In short, the entropy can be defined as a measure about how unsorted the data are. It allows a better approach to address problems related to missing and real feature values.

Feature selection: SVM-RFE

Support Vector Machines Recursive Feature Elimination (SVM-RFE) is one of the most successful classification algorithms based on the foundations of statistical learning theory used by Vapnik in the 70s when he proposed SVMs.

In this kind of problem, the SVM-based techniques are aimed at finding the hyperplane that allows discrimination between positive and negative samples. Furthermore, they are also aimed at maximising the distance between this hyperplane and the different samples to allow a better generalization,[59] as shown in Fig. 3. SVM also introduces the key-concept of kernel: it is a function that provides data with a higher dimensionality, which allows convertion of the original data from non-linearly separable to linearly separable. It yields very good results when dealing with high-dimensional data.[60,61]

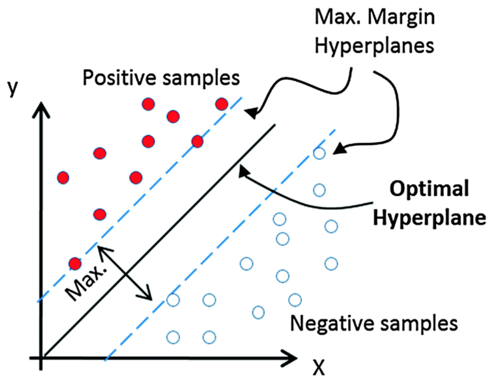


Fig. 3 SVM schema: optimal hyperplane and max. margin hyperplanes (support vectors).

In this work, the R Statistical Package[63] has been used to conduct SVM-RFE with linear regression functions (more specifically the Caret[64] and pROC[65] packages). As SVM-RFE states, the Caret package (an acronym of Classification and Regression Training) provides a set of functions to perform a backward selection of features based on score ranking.

Results and discussion

Dataset description

As noted previously, trials described in this section were developed by using a database composed of 2415 samples. Each and every one of those samples was labelled with one of the groups that make up the dataset. Specifically, the dataset was composed of 350 samples corresponding to enzyme regulatory proteins and 2065 samples of proteins with no enzyme regulatory function. This kind of unbalanced data is not the most suitable to be used as inputs for learning algorithms because the results would present a high sensitivity and low specificity because the learning algorithms would tend to classify most of samples as part of the most common group.[66] To avoid this situation, a pre-processing phase must be used to get a more balanced dataset, in this case by means of the synthetic minority oversampling technique (SMOTE),[67] a technique included within Weka. In short, SMOTE provides a more balanced dataset using an expansion of the lower class by creating new samples, interpolating other minority-class samples. After this pre-processing, the final dataset is composed of 1750 positive samples and 2065 samples of negative or non-enzyme regulatory proteins.

This resultant dataset was processed by MInD-Prot[23] to obtain the 20 Markov Mean Properties for each sample chain, which will be used as features by the classification techniques. Those 20 features can be classified into 4 subsets depending on the physicochemical properties: Mulliken Electronegativity (EM), Kang–Jhon Polarizability (PKJ), van der Waals area (vdWA) and Atom Contribution to P (AC2P). Table 1 shows the performance over validation data of the most common machine learning algorithms included in Weka, used with the standard/recommended configurations.

Table 1 Classification model result. K* yields the best results

	Accuracy	*F*-measure	AUROC	No. of features
AB[a]	0.637	0.622	0.627	20
MLP[b]	0.637	0.622	0.627	20
NB[c]	0.625	0.632	0.645	20
RF[d]	0.839	0.84	0.917	20
J48[e]	0.733	0.734	0.766	20
K* [f]	0.867	0.867	0.948	20

[a] weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W. [b] functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -5990607817048210779. [c] bayes.NaiveBayes -D 5995231201785697655. [d] trees.RandomForest '-I 10 -K 0 -S 1' 4216839470751428698. [e] trees.J48 '-C 0.25 -M 2' -217733168393644444. [f] lazy.kstar -B 20 -M a.

Reference model

The first experiment, as always, was used to choose the most suitable classification model using all the available information. As mentioned in the "Classification methods" section, in order to minimise the influence of the randomness of the partitions among training and testing sets, a 10-fold cross validation was used to perform the experiments and to choose the best classification technique. In the "Dataset description" section, it is also explained that in order to obtain a more balanced training data, the SMOTE algorithm was applied to the original dataset. In Table 1, there is a summary of the results and measures obtained for accuracy, F-measure and AUROC values for the validation dataset and the total number of features used to obtain the classification model.

The validation phase was performed using the classification model obtained with the resampled data in the training phase and applied to the original data. K* yields the best results, as shown in Table 1, with values of accuracy of 0.867 and AUROC over 0.9 for validation. Comparing these results with the rest of the tested techniques, we found that it improved significantly the results offered by AB (0.637 and 0.627), MLP (0.637 and 0.627), NB (0.625 and 0.645) and J48 (0.733 and 0.766). Moreover, it also improves the results of RF (0.839 and 0.917), but with a minor improvement level. AUROC plots for these models are represented in Fig. 4.

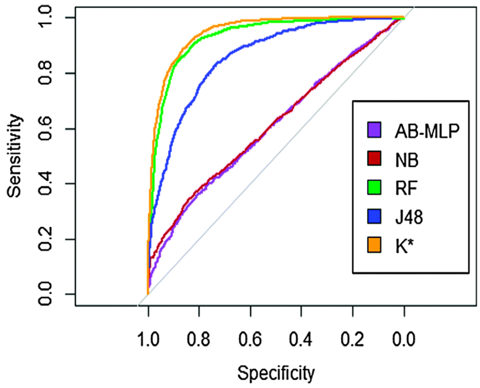


Fig. 4 AUROC for reference models.

Feature subset selection

Once the reference model and the best classification technique were chosen based on the results of the previous section, it was also of interest to determine the main features to discriminate between the proteins with enzyme regulator properties.

The first (and simplest) approach is a grouping of the features based on the physicochemical properties. New models were developed using the K* algorithm and only the selected group of variables. Table 2 and Fig. 5 show the results obtained when all the features are used and the results for each one of the physicochemical groups (EM, PKJ, vdWA and AC2P).

Table 2 Feature selection grouping by type of physicochemical property using K*

Features	Accuracy	F-measure	ROC area	Features
All features	0.867	0.867	0.948	20
EM	0.812	0.812	0.905	5
PKJ	0.844	0.843	0.936	5
vdWA	0.821	0.821	0.912	5
AC2P	0.823	0.823	0.916	5

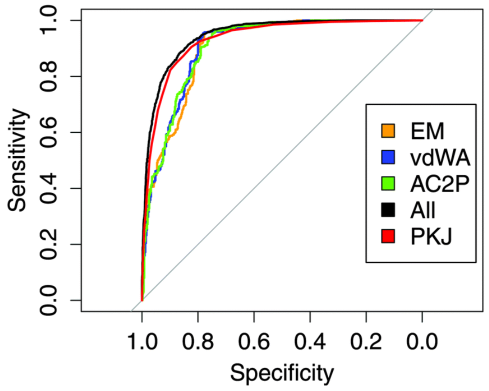


Fig. 5 AUROC for feature selection grouping by type of physicochemical property using K*.

According to these results, the best classification is provided by the PKJ subset. This result is also close to the one yielded by the complete model and the result of the other subsets. Thus, these results seem to indicate that more than one of these features can be used to detect protein enzyme regulator behaviour. Any of these subsets yields an accuracy of over 80% and an AUROC of 90%, which is significantly high, showing a good classification result. The best results (both for accuracy and AUROC) were obtained with the Kang–Jhon polarizability properties and these results are very close to the results obtained using all the features.

Therefore, Table 2 shows that successful classifications can be performed with good results using only five features from the original dataset. Thus, now the question should be: is there another different subset of five (or less) features that yields better results than those in Table 2? In the next two subsections we will try to answer this question.

Wrapper subset evaluator. The first approach to address this question is based on the use of the Wrapper Subset Evaluator provided by the Weka suite.[68] This algorithm makes an exhaustive search within the set of features to determine the most relevant.

It evaluates feature sets by means of the learning scheme, in which case the K* algorithm is the one that provides the most suitable results. Furthermore, a cross validation scheme is used to estimate the accuracy of the learning scheme for a set of features. As in the previous test, the AUROC measure is used. Over each fold, the Wrapper Subset Evaluator will select the most representative set of variables, so the final set of variables selected as the most representative will be the sum of all the variables selected over all the different folds.

Table 3 shows the variables selected by the wrapper algorithm (and the number of folds where they appear). It should be noted that a total of six variables are selected in all the folds (EM_i, PKJ_i, PKJ_m, PKJ_t, $vdWA_c$ and $AC2P_m$), so they could be considered the most relevant ones. On the other hand, there are other features that are never selected or marked as relevant (EM_t and $vdWA_t$)

Table 3 Feature selection by means of Wrapper Subset Evaluator

Feature	% folds where feature appears
EM_c	40
EM_i	100
EM_m	70
EM_s	50
EM_t	0
PKJ_c	90
PKJ_i	100
PKJ_m	100
PKJ_s	90
PKJ_t	100
$vdWA_c$	100
$vdWA_i$	70
$vdWA_m$	60
$vdWA_s$	90
$vdWA_t$	0
$AC2P_c$	80
$AC2P_i$	80
$AC2P_m$	100
$AC2P_s$	40
$AC2P_t$	20

Once the wrapper selected the most relevant features, it is time to check their performance, shown in Table 4.

Table 4 Classification results for variables selected by means of Wrapper Subset approach. Best results achieved with 18 variables from the original

10CV-validation				
Feature	Accuracy	F-measure	ROC area	Features
All available features (reference)	0.867	0.867	0.948	20
Selected at least in one fold	0.872	0.872	0.950	18
Selected in 100% folds	0.841	0.841	0.934	6
Selected in ≥ 90% folds	0.863	0.863	0.944	9
Selected in ≥ 80% folds	0.870	0.870	0.945	11

Using this approach, the results achieved using all the variables can be improved with a lower number of features provided by the wrapper subset algorithm (18 variables that appear at least in one fold or 11 variables that appear in more than 80% of the folds along the process). Furthermore, the results of the classification model shown in Table 4 (which selected features by means of physicochemical properties) are also improved although a great number of features were used.

SVM-RFE. The results provided by Wrapper Subset Evaluator are good (in comparison with the reference models) but present two major drawbacks. Firstly, the wrapper approach makes an exhaustive search because it proves all the possible combinations of features to select the most suitable one, which implies that the time needed for its execution is high (almost 5 days in the computing server used). Secondly, the number of variables finally selected is also quite high to get results similar to the reference models.

In the case of SVM-RFE, the computation time needed to finish the test was reduced to only six hours in the same computing server and as we will see the results are improved using a lower number of features.

Table 5 and Fig. 6 show how SVM-RFE establishes a ranking between the variables depending on their contribution to the overall accuracy of the classification method. A lower value for the order column represents a bigger contribution of that variable to the classification task.

Table 5 Feature selection by means of SVM-RFE

Feature	Accumulated accuracy	Order
PKJ_t	0.7577	1
$AC2P_t$	0.7962	2
EM_t	0.8505	3
EM_s	0.8569	4
$wdWA_s$	0.8659	5
PKJ_s	0.8610	6
$wdWA_t$	0.8624	7
$AC2P_s$	0.8964	8
PKJ_m	0.9019	9
PKJ_i	0.9099	10
$AC2P_i$	0.9148	11
EM_i	0.9135	12
$wdWA_i$	0.9190	13
$wdWA_m$	0.9162	14
$AC2P_m$	0.9201	15
EM_m	0.9272	16
$AC2P_c$	0.9288	17
PKJ_c	0.9275	18
EM_c	0.9291	19
$wdWA_s$	0.9280	20

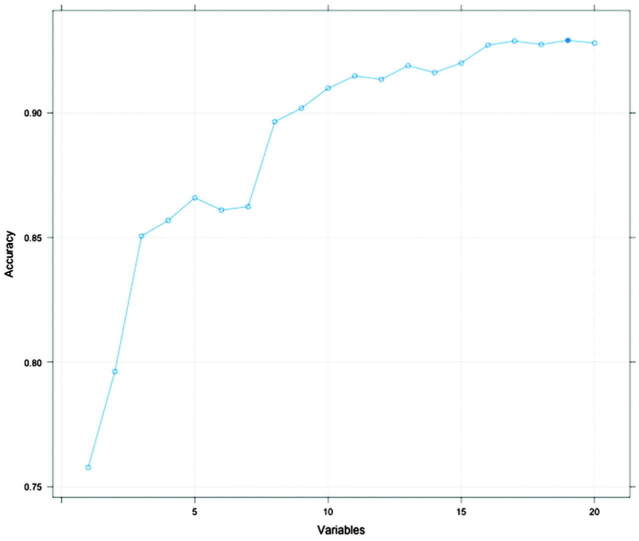


Fig. 6 SVM-RFE: evolution of accuracy level.

It should be noted that the R package for SVM-RFE uses values for accuracy within the reduced feature of the elimination process, so this value will be used for comparisons.

According to the variable ranking established by SVM-RFE, the accuracy levels of the classification performed with PKJ properties can be reached using only 3 features (instead of 5): PKJ_t, $AC2P_t$ and EM_t. The accuracy reached with SVM-RFE is 0.8505 instead of 0.844 reached with PKJ properties.

Furthermore, adding only two more features (EM_s and $wdWA_s$) accuracy value reaches 0.8659, a very similar value to that obtained by K* using all the available features (0.867) and obviously higher than the accuracy provided by the classification using the five available PKJ properties. It can be noted that these features contain information from the entire protein chain (total region). This can be explained by the complexity of the regulation of enzymes.

Comparing these results with the results obtained with Wrapper Subset Evaluator (see Table 4) SVM-RFE can achieve (or improve) them using only the first eight variables selected (instead of 18 or 11 variables selected with the Wrapper approach).

Finally, if we use these 8 most relevant variables from SVM-RFE we will able to improve the results employing all the features (0.8964 vs. 0.867).

Conclusions

The current work presents the first classification model to predict enzyme regulation function-related proteins. This classification model was obtained by means of the Markov mean properties calculated with the MInD-Prot tool.

The dataset contains a total of 2415 samples, out of which 350 correspond to positive samples, that is, to enzyme regulator function proteins. The dataset information is composed of the topological information of the amino acid contact networks of the proteins, the atom physicochemical properties (Mulliken electronegativity, Kang–Jhon polarizability, van der Waals area, atom contribution to P) and the protein 3D regions.

First of all, several classification methods were tested using all the information available (composed of a total of 20 features). As result of these tests, K* seems to be the algorithm that yields the best results. Secondly, we tried to perform the classification using a more reduced set of features, so we proposed Wrapper Subset Evaluator and SVM-RFE approaches to establish which of the original features provides more information to the final model.

A Wrapper Subset Evaluator approach was tested, obtaining good results. However, it presents two major drawbacks: the time needed to perform all the calculations and the high number of variables needed to get similar results to those obtained with all the features.

Finally, SVM-RFE selected features to improve the classification results using all the available data with only 8 out of 20 features calculated with MInD-Prot. Good results can be obtained using only 3 out of the initial 20 features. Therefore, these results can help to predict enzyme regulation function-related proteins using only a reduced amount of molecular information encoded into the protein 3D structure. Therefore, with the new predictions it is possible to search for new molecular targets involved in diverse diseases.

Feature selection by means of SVM-RFE also allows comparisons of how the variables provide more information to the final classification task.

Acknowledgements

Cristian R. Munteanu acknowledges the funding support for a research position by the “Isidro Parga Pondal” Programme from Xunta de Galicia, Spain (supported by the European Social Fund). This work is supported by the General Directorate of Culture, Education and University Management of Xunta de Galicia (Ref. 10SIN105004PR) and “Collaborative Project on Medical Informatics (CIMED)” PI13/00280 funded by the Carlos III Health Institute. The authors also thank the Galician Supercomputing Centre (CESGA) for the provision of computational support.

Notes and references

1. A. Smith, S. P. Datta, G. H. Smith, P. N. Campbell, R. Bentley and H. McKenzie, Oxford dictionary of biochemistry and molecular biology, Oxford University Press, 2000 .
2. C. M. Grisham and R. H. Garrett, Biochemistry, Saunders College Pub., Philadelphia, 1999, pp. 426–427 .
3. R. Wolfenden and M. J. Snider, Acc. Chem. Res., 2001, 34, 938–945 .
4. K. E. Neet, Methods Enzymol., 1995, 249, 519–567 .
5. C. R. Munteanu, H. Gonzalez-Diaz and A. L. Magalhaes, J. Theor. Biol., 2008, 254, 476–482 .
6. C. R. Munteanu, J. M. Vazquez, J. Dorado, A. P. Sierra, A. Sanchez-Gonzalez, F. J. Prado-Prado and H. Gonzalez-Diaz, J. Proteome Res., 2009, 8, 5219–5228 .
7. Y. Rodriguez-Soca, C. R. Munteanu, J. Dorado, A. Pazos, F. J. Prado-Prado and H. Gonzalez-Diaz, J. Proteome Res., 2010, 9, 1182–1190 .
8. Y. Rodriguez-Soca, C. R. Munteanu, J. Dorado, J. Rabuñal, A. Pazos and H. González-Díaz, Polymer, 2010, 51, 264–273 .
9. H. Gonzalez-Diaz, F. Prado-Prado, X. Garcia-Mera, N. Alonso, P. Abeijon, O. Caamano, M. Yanez, C. R. Munteanu, A. Pazos, M. A. Dea-Ayuela, M. T. Gomez-Munoz, M. M. Garijo, J. Sansano and F. M. Ubeira, J. Proteome Res., 2011, 10, 1698–1718 .
10. H. Gonzalez-Diaz, C. R. Munteanu, L. Postelnicu, F. Prado-Prado, M. Gestal and A. Pazos, Mol. BioSyst., 2012, 8, 851–862 .
11. S. Archer, NIDA Res. Monogr., 1978, 86–102 .
12. Recent Advances in QSAR Studies: Methods and Applications, ed. T. Puzyn, J. Leszczynski and M. T. D. Cronin, Springer, 2010 .
13. V. Aparna, J. Jeevan, M. Ravi, G. R. Desiraju and B. Gopalakrishnan, Bioorg. Med. Chem. Lett., 2006, 16, 1014–1020 .
14. A. Speck-Planche and V. V. Kleandrova, Curr. Top. Med. Chem., Netherlands, 2012, vol. 12, pp. 1734–1747 .
15. A. Speck-Planche, V. V. Kleandrova, F. Luan and M. N. Cordeiro, Bioorg. Med. Chem., Elsevier Ltd, England, 2012, vol. 20, pp. 4848–4855 .
16. H. Gonzalez-Diaz, F. J. Prado-Prado, L. Santana and E. Uriarte, Bioorg. Med. Chem., 2006, 14, 5973–5980 .
17. F. J. Prado-Prado, I. Garcia, X. Garcia-Mera and H. Gonzalez-Diaz, Chemom. Intell. Lab. Syst., 2011, 107, 227–233 .

18. A. R. Katritzky, O. V. Kulshyn, I. Stoyanova-Slavova, D. A. Dobchev, M. Kuanar, D. C. Fara and M. Karelson, Bioorg. Med. Chem., 2006, 14, 2333–2357 .
19. O. Ivanciuc, Curr. Proteomics, 2009, 6, 289–302 .
20. H. González-Díaz, F. Prado-Prado, L. G. Pérez-Montoto, A. Duardo-Sánchez and A. López-Díaz, Curr. Proteomics, 2009, 6, 214–227 .
21. H. Gonzalez-Diaz, A. Perez-Bello, E. Uriarte and Y. Gonzalez-Diaz, Bioorg. Med. Chem. Lett., 2006, 16, 547–553 .
22. S. Carbon, A. Ireland, C. J. Mungall, S. Shu, B. Marshall and S. Lewis, Bioinformatics, 2009, 25, 288–289 .
23. C. R. Munteanu and H. González-Díaz, MInD-Prot - Markov Indices for Drugs and Proteins, Register No.: 03/2012/1051 (SC-228-12), (2012), Santiago de Compostela, Spain.
24. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. A. Witten, SIGKDD Explorations, 2009, 11, 10–18 .
25. I. H. W. a. E. Frank, Data Mining Practical Machine Learning Tools and Techniques, Kaufmann, San Francisco, 2nd edn, 2005 .
26. V. Aguiar-Pulido, J. A. Seoane, J. R. Rabunal, J. Dorado, A. Pazos and C. R. Munteanu, Molecules, 2010, 15, 4875–4889 .
27. W. S. Hayes and M. Borodovsky, Genome Res., 1998, 8, 1154–1171 .
28. K. Prank, E. Schulze, O. Eckert, T. W. Nattkemper, M. Bettendorf, C. Maser-Gluth, T. J. Sejnowski, A. Grote, E. Penner, A. von Zur Muhlen and G. Brabant, Eur. J. Endocrinol., 2005, 153, 301–305 .
29. B. K. Lavine, C. E. Davidson and W. S. Rayens, Comb. Chem. High Throughput Screening, 2004, 7, 115–131 .
30. J. Devillers and A. T. Balaban, Topological Indices and Related Descriptors in QSAR and QSPR, Gordon and Breach, The Netherlands, 1999 .
31. H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne, Nucleic Acids Res., 2000, 28, 235–242 .
32. G. Wang and J. R. L. Dunbrack, Bioinformatics, 2003, 19, 1589–1591 .
33. H. Gonzalez-Diaz, F. Romaris, A. Duardo-Sanchez, L. G. Perez-Montoto, F. Prado-Prado, G. Patlewicz and F. M. Ubeira, Curr. Pharm. Des., 2010, 16, 2737–2764
34. H. Gonzalez-Diaz, A. Duardo-Sanchez, F. M. Ubeira, F. Prado-Prado, L. G. Perez-Montoto, R. Concu, G. Podda and B. Shen, Curr. Drug Metab., 2010, 11, 379–406
35. H. Gonzalez-Diaz, F. Prado-Prado and F. M. Ubeira, Curr. Top. Med. Chem., 2008, 8, 1676–1690 .
36. R. Todeschini and V. Consonni, Handbook of Molecular Descriptors, Wiley-VCH, 2002 .
37. T. J. Hou and X. J. Xu, J. Chem. Inf. Comput. Sci., 2003, 43, 1058–1067 .
38. H. González-Díaz, L. Saiz-Urra, R. Molina, L. Santana and E. Uriarte, J. Proteome Res., 2007, 6, 904–908 .
39. H. Gonzalez-Diaz, L. Saiz-Urra, R. Molina, Y. Gonzalez-Diaz and A. Sanchez-Gonzalez, J. Comput. Chem., 2007, 28, 1042–1048 .
40. H. Gonzalez-Diaz, R. Molina and E. Uriarte, FEBS Lett., 2005, 579, 4297–4301 .
41. R. Concu, G. Podda, E. Uriarte and H. Gonzalez-Diaz, J. Comput. Chem., 2009, 30, 1510–1520 .
42. H. González-Díaz, Y. Pérez-Castillo, G. Podda and E. Uriarte, J. Comput. Chem., 2007, 28, 1990–1995 .
43. G. J. McLachlan, K.-A. Do and C. Ambroise, Analyzing microarray gene expression data, Wiley, 2004 .
44. C. Ferri, J. Hernandez-Orallo and R. Modroiu, Pattern Recognit. Lett., 2009, 30, 27–38 .
45. I. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, 2nd edn, 2005 .
46. H. Liu and R. Setiono, 13th International Conference on Machine Learning, Bari, Italy, 1996 .
47. C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, USA, 1995 .
48. C. M. Bishop, Pattern recognition and machine learning, Springer, 2006 .
49. G. H. John and P. Langley, 11th Conference on Uncertainty in Artificial Intelligence, Montreal, Quebec, 1995.
50. L. Breiman, Mach. Learn., 2001, 45, 5–32 .
51. J. G. Cleary and L. E. Trigg, Machine Learning International Workshop, 1995 .
52. D. W. Aha, D. Kibler and M. K. Albert, Mach. Learn., 1991, 6, 37–66 .
53. T. Cover and P. Hart, IEEE Trans. Inf. Theory, 1967, 13, 21–27 .
54. D. W. Aha, Int. J. Man–Mach. Stud., 1992, 36, 267–287 .
55. D. W. Aha and D. Kibler, Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 1989.
56. S. Cost and S. Salzberg, Mach. Learn., 1993, 10, 57–78 .

57. C. E. Shannon, W. Weaver, R. E. Blahut and B. Hajek, The mathematical theory of communication, University of Illinois Press, Urbana, 1949 .
58. D. J. MacKay, Information theory, inference and learning algorithms, Cambridge University Press, 2003 .
59. C. J. C. Burges, Data Min. Knowl. Disc., 1998, 2, 121–167 .
60. O. Chapelle, P. Haffner and V. N. Vapnik, IEEE Trans. Neural Networ., 1999, 10, 1055–1064 .
61. L. S. Moulin, A. P. Alves Da Silva, M. A. El-Sharkawi and R. J. Marks Ii, IEEE Trans. Power Syst., 2004, 19, 818–825 .
62. I. Guyon, J. Weston, S. Barnhill and V. Vapnik, Mach. Learn., 2002, 46, 389–422 .
63. R Core Team, R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, 2013, http://www.R-project.org/.
64. M. Kuhn, Journal of Statistical Software, 2008, 28, 1–26 .
65. X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez and M. Müller, BMC Bioinf., 2011, 12, 77 .
66. F. Fernández-Navarro, C. Hervás-Martínez and P. Antonio Gutiérrez, Pattern Recogn., 2011, 44, 1821–1833 .
67. N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, J. Artif. Int. Res., 2002, 16, 321–357 .
68. R. Kohavi and G. H. John, Artif. Intell., 1997, 97, 273–324