Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

Predicting GPCR and Enzymes Function with a Global Approach Based on LCS

Luiz Melo Romão
Universidade da Região de Joinville
Departamento de Informática
Joinville - SC - Brasil
luizmromao@gmail.com

Julio César Nievola
Pontifícia Universidade Católica do Paraná
Pós-Graduação em Informática
Curitiba - PR - Brasil
nievola@pucpr.ppgia.br

***Abstract*—The families of G-Protein Coupled Receptor (GPCR) and enzymes are among the main protein family. They represent to the scientific and medical communities, a significant target for bioactive and drug discovery programs. The model of classification of enzymes and GPCR is characterized by its hierarchical structure in format of tree and this makes more difficult its prediction. In this work we propose an adapted version of Learning Classifier Systems (LCS) data mining algorithm which tends to be more efficient than statistical methods based on homology used in tool such as PSI-BLAST. Hence, a new global model approach, called HLCS (Hierarchical Learning Classifier System) is used to predict the function of enzymes and GPCR, respecting its organizational structure of classes throughout the model development. The HLCS is expressed as a set of IF-THEN classification rules, which have the advantage of representing comprehensible knowledge to biologist users. The HLCS is evaluated with eight datasets from enzymes and GPCR, and compared with a Global Naive Bayes algorithm, named GMNB. In the tests realized the HLCS outperformed the GMNB in the databases of the GPCR proteins group type.**

***Index Terms*—Prediction of protein function, Hierarchical Classification, Learning Classifier Systems.**

I. Introduction

In order to develop a drug it is necessary to select a protein associated with the disease so it is interesting to therapeutically affect their function and expression. Hence, to succeed in their experiments, scientific and medical communities are dependent on the quality of the databases used.

The current protein databases are formed mainly by experiments performed with the FASTA [1] or PSI-BLAST [2] tools, which use statistics based on the homology method. In this method, a new amino acid sequence is compared with other sequences in a database. The more similar sequence has its function inferred for the new sequence. However, analysis done on the mapping of protein functions using the method of homology, showed that in about 40% of the cases, a sequence has no significant similarity with a protein already characterized. Since these tools are not sensitive enough to find other similarities between these proteins, wrong observations may be propagated through the databases at the same speed with which new sequences have been analysed.

With this, other ways of predicting protein function have been investigated as alternatives to the homology method. The work of [3] shows that the combination of some protein information can be effective in predicting protein function when used in Data Mining methods, such as classification.

The complexity of this type of application comes from the protein's organization, which has the label of the classes hierarchically structured This type of problems exists when one or more classes are divided into subclasses or grouped into superclasses arranged in a hierarchical structure, like a tree or a directed acyclic graph (DAG). Another difficulty in the predictive process is directly related to the depth of the hierarchy. Normally, the predictive performance decreases with increasing depth (specificity), whereas the amount of specific examples is smaller, which makes the model training process and prediction of the sample more difficult.

In this work we present the algorithm Hierarchical Learning Classifier System (HLCS) that presents a global solution for the classification of GPCR and enzymes function. The HLCS is based on Learning Classifier System (LCS), which is a method that generates its results in a set of rules in the IF-THEN format, which, according to [4] is more understandable than models such as neural networks, support vector machines and others.

The remainder of this paper is organized as follows: Section 2 discusses the hierarchical classification concept and how to distinguish hierarchical problems. Section 3 discusses the problem of predicting protein functions and their databases. The HLCS architecture and the operation of each of its components are described in Section 4. Section 5 demonstrates the computational results achieved and Section 6 presents the conclusions of this study and the possible directions for future research.

II. Hierarchical Classification

A classification problem is defined by a set of examples where each example is described by a set of predictive attributes associated with a class attribute. The classification task of data mining consists of building, in a training phase, a classification model that maps each example t_i to a class c $\in$ C of the target application domain, with i = 1, 2, ..., n, where n represents the number of examples in the training set [5]. However, several classification scenarios have real problems that are much more complex. Cases such as text categorization, web content search, prediction of protein function, among

others, are problems in which the class label is hierarchically structured .

The concept of hierarchical classification is proposed in [6] as a specific type of structured classification problem, where the output of the classification algorithm is defined by a class taxonomy. The class taxonomy, was explored in [7], as a hierarchical concept defined over a partially established set $(C, \prec)$, where C is a finite set that enumerates all the concepts of class in the application and the $\prec$ relation represents the relationship "*IS-A*". Thus, "*IS-A*" is defined in [6] relationship as asymmetric, anti-reflexive and transitive Thus, "*IS-A*" is defined in [6] relationship as asymmetric, anti-reflexive and transitive, as follows:

- The only one greatest element "R" is the root of the tree;
- $\forall c_i, c_j \in C$, if $c_i \prec c_j$ then $c_j \not\prec c_i$;
- $\forall c_i \in C$, $c_i \not\prec c_i$;
- $\forall c_i, c_j, c_k \in C$, $c_i \prec c_j$ and $c_j \prec c_k$ imply $c_i \prec c_k$.

In this manner, any classification problem with a class structure that satisfies the four properties mentioned in the "*IS-A*" hierarchy, can be considered as a hierarchical classification problem.

The hierarchical classification problems can be differentiated using three specific criteria:

- **Type of Structure:** The structure of a hierarchical classification problem can be organized in tree or DAG.
- **How deep the classification in the hierarchy is performed:** The classification method can be implemented so that whenever a leaf node will be classified, called the Mandatory Leaf-Node Prediction, or one can stop the sorting on any node at any level of hierarchy, called Non-Mandatory Leaf Node Prediction.
- **Type of Algorithmic Approach:** Basically, the hierarchical classification of algorithms can be classified into local and global.

With respect to the last item, it is important to make clear that most of the solutions that work with hierarchical problems use the local model in the training phase to build the classification model. This model trains a binary classifier for each node of the class hierarchy. In this case, it is necessary to use N independent local classifiers, one for each class except the root node. Therefore, the number of classifiers to be trained can be very large in situations where there are many classes. Moreover, in using the local approach, the technique can provide inconsistent results, because there is no guarantee that the class hierarchy will be respected.

In the global approach, a single classification model is built from the training set, taking into account the hierarchy of classes as a whole during a single execution of the classifier algorithm. In the global approach, the fact that the algorithm maintains hierarchical relationships between classes during the phases of training and testing makes the outcome of the prediction easier to understand.

According to [6], the global approach is still underexploited in the literature and it deserves more investigation because it builds a singular coherent classification model. Even though a single model produced by the global approach will tend to be more complex (larger) than each of the many classification models produced by the local-model approach, intuitively the single global model will tend to be much simpler (smaller) than the entire hierarchy of local classification models.

III. Proteins and DataBases

Proteins are the main components of the cell, and perform almost all functions related to cell activity. They consist of long strings or chains of amino acids, also called polypeptide chains that fold into a number of different structures [8]. According to [5], the prediction of protein function links biological functions to proteins. This knowledge can help researchers better understand diseases, drug development, and preventive medicine, among others.

The two databases used in this article involve the families of G-Protein Coupled Receptor (GPCR) and Enzymes. The protein functional classes are given unique hierarchical indexes by [9] in the case of GPCRs and by Enzyme Commission Codes [10] in the case of enzymes. According to [11], GPCR divides the superfamily into six classes. Each superfamily can be organized into a hierarchy of classes, class subfamilies, class subfamily subfamilies and types.

The enzyme nomenclature scheme was developed starting in 1955, when the International Congress of Biochemistry in Brussels set up an Enzyme Commission. The first version was published in 1961. The current sixth edition, published by the International Union of Biochemistry and Molecular Biology in 1992, contains 3196 different enzymes classified in a hierarchical tree structure. The Enzyme Commission number (EC number) is a numerical classification scheme for enzymes, based on the chemical reactions they catalyze [10]. As a system of enzyme nomenclature, every EC number is associated with a recommended name for the respective enzyme. For example, the code for the tripeptide aminopeptidases is "EC 3.4.11.4", whose components indicate the following groups of enzymes:

- EC 3 enzymes are hydrolases (enzymes that use water to break up other molecule)
- EC 3.4 are hydrolases that act on peptide bonds
- EC 3.4.11 are those hydrolases that cleave off the amino-terminal amino acid from a polypeptide
- EC 3.4.11.4 are those that cleave off the amino-terminal end from a tripeptide

IV. The Proposed HLCS

Conceived in 1975 by John Holland [12], the Learning Classifier System (LCS) consists of a set of rules called classifiers. The LCS develops a model of intelligent decision-making, using two biological metaphors, evolution and learning, where learning guides the evolutionary component to move in the direction of the best rules.

The LCS has been used with great success in several areas like robotics [13], environment navigation [14], [15], function approximation [16], data mining [17] and others. And their main approaches are XCS [14], [18], ACS [19] and UCS [20].

However, the topic of this work, hierarchical classification problems like protein function, has not been directly addressed by these and neither other approach of LCS.

The purpose of this paper is to present a global Hierarchical Learning Classifier System (HLCS) - based on the LCS model - to predict the function of proteins. In our previous work [21] we have already presented a solution for predicting protein function using LCS. However, it presents a local type approach, that is, it does not take into account class hierarchy during model training.

In order to work with the class hierarchy, this new version presents a specific component for this task that is the evaluation component of the classifiers. This component has the task of analysing the predictions of classifiers considering the class hierarchy. In addition to this, the HLCS architecture consists of the following modules: population of classifiers, GA component, performance component and credit assignment component, which interacts internally.

The details of each one the HLCS components follow below.

A. Classifier Population and Evaluation Component

The size of the population of classifiers ($SizePop$) is defined by the HLCS algorithmic settings. The set of all classifiers is the predictive model. Each classifier C_i ($0 < i \leq SizePop$) of the HLCS comprises: a n set of conditions (where n=number of attributes of an training instance), the class value and the classifier quality measure.

$$C_i\text{=}[(Cond_0 \text{ and...and } Cond_n)(ClassValue)(Q_{classifier})]$$

Each condition has three parameters: OP, VL, A/I, where: OP: operator relation (= or !=), VL: condition value and A/I: the choice of an active or inactive attribute, which determines whether the condition will be used in the classifier or not.

In order to form the initial population of classifiers, the HLCS randomly chooses an instance of the training base as a model. For each attribute of an instance a condition in the classifier is created. At the beginning, the conditions start with the operator relation (OP) "=". The condition value (VL) receives the value attribute of the instance and whether the condition will be active (A) or inactive (I) is randomly determined.

The last step in the creation of the initial population of classifiers is define the quality of the classifier. ($Q_{classifier}$). To calculate the classifier quality two factors are considered: the percentage of positive classes predicted ($recall$) and the hierarchical control evaluation of the classifier ($evaluation\_h$). The evaluation represents the predictive ability of the classifier, considering not only the class in question, but all the class antecedents in the hierarchy. This process is performed by the evaluation component.

The evaluation is a way of considering the predictions made by the classifiers in the hierarchy of the problem. Principally, in the case of the prediction of protein function, is very important in biological terms, the knowledge of all classes that are part of a function, from the root to the most specific class. Based on this principle, this evaluation is responsible for promoting the classifiers that are close to their main goal, taking into consideration the quality of the classifier that predict at least some kind of antecedent class from the real class.

Through the evaluation component, the HLCS is able to verify whether an prediction is correct, partially correct or incorrect. With this model, it is possible to make the reward given to the classifier more dynamic, according to its prediction.

Therefore, for each prediction, the HLCS achieves the evaluation, ($evaluation\_h(i)$ where: $0 < i \leq$ number of instances) of the classifier as follows:

$$evaluation\_h(i) = \begin{cases} 1, & \text{if Correct} \\ 0, & \text{if Incorrect} \\ \frac{nodes\_common}{level\_real\_class}, & \text{if Partially Correct} \end{cases} \quad (1)$$

The correct prediction occurs when the predicted class is equal to the real class, as shown in Figure 1. In the examples, the black box represents the predicted class and the black oval the real class. In the case of the figure example, according to Equation 1, the value of the classifier evaluation is 1.

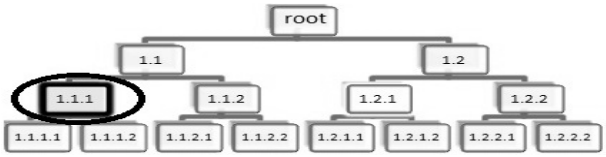

Fig. 1. Example of correct prediction

The incorrect prediction occurs when the predicted class is different from the real class and all of its antecedents, as shown in Figure 2. In the case of the figure example, according to Equation 1, the value of the classifier evaluation is 0.

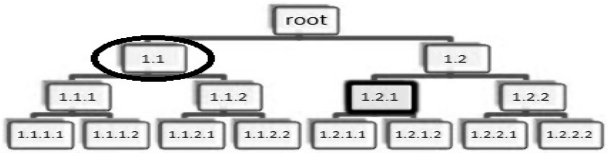

Fig. 2. Example of incorrect prediction

The prediction is considered partially correct when the algorithm misses the real class but hits at least one antecedent of this class, except the root node. In this case, the value incorporated into its quality measure will vary based on the error, taking into account the number of nodes in common with the root node and the level of real class, as shown in Equation 1. In Figure 3, the predicted class (1.2.1) has one antecedents in common with the real class (1.2.2) and, as the real class is in the 2rd level, the value of the classifier evaluation is 0.5.

The final classifier evaluation is then calculated as the sum of the evaluations, as shown in Equation 2, and this value is incorporated into its quality measure.

$$evaluation\_h \leftarrow \sum evaluation\_h(i) \quad (2)$$

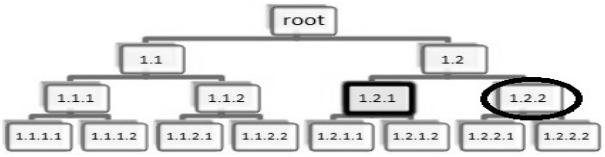

Fig. 3. Example of partially correct prediction

Hence, the classifier quality measure is calculated as follows:

$$Q_{classifier} \leftarrow \frac{TP}{(TP+FN)} * evaluation\_h \qquad (3)$$

where:

- TP: (True Positive) is the number of instances that satisfy all the active attributes of the classifier, and where the predicted class is equal to the real class;
- FN: (False Negative) is the number of instances that do not satisfy all the active attributes of the classifier where the predicted class is equal to the real class.

This process is then repeated for all classifiers of the population.

B. Performance Component

After creating the initial population, the process of learning and development of classifiers begins. In the first stage, the HLCS randomly chooses an training instance and compares with the classifiers of population. The comparison is made between the attributes of the training instance and the conditions of the classifier. These classifiers, whose active conditions are equal to the training instance attributes, form a action set, and are conducted to the performance component where they will participate in a competition.

The performance component is used to analyse the classifiers and evaluate the learning process. The classifier that obtains the highest bid ($eBid$) be used to predict the class of the training instance. The calculation of $eBid$ is shown in Equation 4.

$$eBid \leftarrow 1+((\frac{total-actives}{total})*Q_{classifier})*(1+Mod) \quad (4)$$

where:

- $Q_{classifier}$: the classifier quality measure;
- Mod: a random value that represents a modulation characterized by a noise with a normal distribution with mean 0 and variance 1;
- $total$: total classifier conditions;
- $actives$: total classifier active conditions.

In order to analyse the result of the prediction in the training instance and define the reward of the classifier, the credit assignment component is called.

Algorithm 1 Program HLCS Model

```
1: for i = 1 to Size_Population do
2:     generate initial population();
3:     evaluate fitness initial population();
4: end for
5: for j = 1 to NumberGeneration do
6:     for k = 1 to NumberCompetitions do
7:         id_instance = select random id instance();
8:         for i = 1 to Size_Population do
9:             if compare(instance(id_instance),classifier(i)) then
10:                add classifier(i) to Action Set();;
11:            end if
12:        end for
13:        for i = 1 to SizeActionSet do
14:            evaluate eBid classifier(i);
15:            if classifier(i).eBid > best_eBid then
16:                best_eBid ← classifier(i).eBid
17:                id_best_eBid ← i;
18:            end if
19:        end for
20:        if (compare(class(id_instance),class(id_best_eBid)) then
21:            evaluation_h ← 1;
22:            Q_classifier ← Q_classifier * (1 + PR + evaluation_h)
23:        else
24:            evaluation_h ← number_of_nodes_in_common / level_of_real_class
25:            Q_classifier ← Q_classifier * (1 − PR + evaluation_h)
26:        end if
27:        crossover();
28:        mutation();
29:    end for
30: end for
```

C. Credit Assignment Component

The credit assignment component has the function of analysing the outcome of the classifier prediction of the training instance. The credit assignment is implemented by a modification of bucket brigade algorithm, and its analysis is repassed to the classifier quality measure. If the winner classifier correctly predicts the training class instance, it gets a reward, as shown in Equation (5). Otherwise, the classifier receives a punishment for the prediction error, defined in Equation (6). In the case of an error, the evaluation component will interact to determine the degree of error according to the hierarchy of classes of instance.

$$Q_{classifier} = Q_{classifier} * (1 + PR + evaluation\_h) \quad (5)$$

$$Q_{classifier} = Q_{classifier} * (1 - PR + evaluation\_h) \quad (6)$$

where:

- PR: : the percentage of reward defined in the HLCS settings;
- $evaluation\_h$: Defined in Equation 2.

D. GA Component

The GA component is responsible for creating and modifying the classifiers so that they become more efficient. This component uses Genetic Algorithms (GA), which are based on probabilistic techniques that mimics the process of natural evolution. Through the crossover and mutation genetic operators, the classifiers evolve and their quality improve.

For this purpose, a genetic algorithm is applied to the classifiers in the action set. Two classifiers are selected by a tournament method, recombined, and mutated. If the offspring

classifiers have higher quality, they are inserted in the population while other are deleted to keep the number of classifiers in the population constant.

The Algorithm 1 show the entire procedure for creating the HLCS global model.

V. Computational Results

The experiments were performed with datasets from two different proteins families: Enzymes and GPCRs. These bases were used in the work of [22] and are available at https://sites.google.com/site/carlossillajr/resources. Enzymes are catalysts that accelerate chemical reactions while GPCRs are proteins involved in signalling and are particularly important in medical applications as it is believed that from 40% to 50% of current medical drugs target GPCR activity [11].

Each dataset has four different versions based on different kinds of predictor attributes, and in each dataset the classes to be predicted are hierarchical protein functions. Each type of binary predictor attribute indicates whether or not a "protein signature" (or motif) occurs in a protein [22]. The motifs used in this work were: Interpro Entries, FingerPrints from the Prints database, Prosite Patterns and Pfam. Apart from the presence/absence of several motifs according to the signature method, each protein has two additional attributes: the molecular weight and the sequence length.

Before performing the experiments, the following preprocessing steps were applied to the datasets: (1) Every class with fewer than 10 examples was merged with its parent class. If after this merge the class still had fewer than 10 examples, this process would be repeated recursively until the examples would be labeled to the Root class. (2) All examples whose most specific class was the Root class were removed. (3) A class blind discretization algorithm based on equal-frequency binning (using 20 bins) was applied to the molecular weight and sequence length attributes, which were the only two continuous attributes in each dataset. Table I shows the main characteristics of datasets after the preprocessing steps which are detailed in [22]. In all datasets, each protein (example) is assigned, at most, to one class at each level of the hierarchy.

TABLE I
THE LAST COLUMN PRESENTS THE NUMBER OF CLASSES AT EACH LEVEL OF THE HIERARCHY (1ST/2ND/3RD/4TH LEVELS)

Protein Type	Signature Type	# of Attributes	# of Examples	# Classe/Level
Enzyme	Interpro	1216	14027	6/41/96/187
	Pfam	708	13987	6/41/96/190
	Prints	382	14025	6/45/92/208
	Prosite	585	14041	6/42/89/187
GPCR	Interpro	450	7444	12/54/82/50
	Pfam	75	7053	12/52/79/49
	Prints	283	5404	8/46/76/49
	Prosite	129	6246	9/50/79/49

The results of the HLCS algorithm were compared with the Global-Model Naive Bayes (GMNB) approach [22], where the authors proposed a Naive Bayes model to deal with a hierarchical classification problem.

In order to evaluate the algorithms we used the metrics of hierarchical precision (hP), hierarchical recall (hR) and hierarchical F-measure proposed by [23]. These measures are, in fact, extended versions of the known measures like precision, recall and F-measure, tailored to the scenario of hierarchical classification. These measures are calculated as follows: A label set C_i assigned to an instance d_i is called consistent with a given hierarchy if C_i forms a connected "proper" subgraph of the hierarchy graph rooted in the top node, i.e. if $c_k \in C_i and c_j \in Ancestors(c_k)$, then $c_j \in C_i$. Then for any instance (d_i, C_i) classified into subset $C_i^{'}$ we extend sets C_i and $C_i^{'}$ with the corresponding ancestor labels:

$$A_i = \bigcup_{c_k \in C_i} Ancestors(c_k),$$
$$A_i^{'} = \bigcup_{c_k \in C_i^{'}} Ancestors(c_k)$$

Hence, the measures are calculated as shown in Equations 7, 8 e 9.

$$hR = \frac{|A_i \bigcap A_i^{'}|}{|A_i|} \tag{7}$$

$$hP = \frac{|A_i \bigcap A_i^{'}|}{|A_i^{'}|} \tag{8}$$

$$hF - Measure = \frac{2 * hP * hR}{hP + hR} \tag{9}$$

The HLCS experiments were carried out using the 10-fold cross-validation method and the results are described by the average computed over each dataset. The comparison results between the proposed HLCS method and the GMNB approach are shown in Table II.

TABLE II
HIERARCHICAL PRECISON (HP), RECALL (HR) AND HF-MEASURE (HF) ON THE HIERARCHICAL PROTEIN FUNCTION DATASETS.

Protein Type	Signature Type	HLCS			GMNB		
		hP	hR	hF	hP	hR	hF
Enzyme	Interpro	87.80	85.36	86.56	94.96	89.58	90.53
	Pfam	86.34	81.47	83.83	95.15	86.94	88.72
	Prints	89.69	82.33	85.85	92.21	87.26	87.98
	Prosite	90.35	86.27	88.26	95.14	89.53	90.70
GPCR	Interpro	90.26	74.30	81.51	87.60	71.33	77.01
	Pfam	82.53	60.30	69.69	77.23	57.52	64.40
	Prints	86.50	68.18	76.25	87.06	69.42	75.38
	Prosite	79.42	60.45	68.65	75.64	53.73	61.14

The results show that the HLCS had a better performance in the databases of the GPCR proteins group type, probably due to the fact that these bases have a much better distribution of classes among the different levels of hierarchy. In other results, when compared with algorithm GMNB, there is a certain balance between the measurements.

The main advantage of the algorithm HLCS against GMNB is the form in which the model presents the results generated. While GMNB approach applies a model based on probability, HLCS generates a set of rules making the knowledge acquired readily understood by medical and scientific communities. The

following example shows a piece of one of the rules generated by it.

IF PS00786 = 0 and PS01184 = 0 and PS00399 = 1 and PS00137 = 0 and PS00774 = 0 and PS00687 = 0 and PS00506 = 0 and ... and MW = (37182.5-39108] and SL = (337.5-356.5] **THEN** EC 3.6.3.30.

VI. Conclusion

The paper introduced a new global model approach for hierarchical classification problems using LCS and applied it to the classification of biological dataset. The proposed HLCS unveils a global classification model in the form of an ordered list of IF-THEN classification rules which can predict terms at all levels of the hierarchy, satisfying the parent-child relationships between terms. The advantage of HLCS in contrast to other approaches is their adaptability. Based on the LCS model, the HLCS makes constant iterations of environmental samples to create their classification rules, making it a more flexible classification model.

The results comparing HLCS with the GMNB algorithm show that the HLCS had similar results on some measures and this proves that the use of LCS models can be an alternative to the hierarchical prediction problems. During the experiments we observed the need to better define the parameters used in the algorithm HLCS, in order to optimize the performance and robustness of the model system and achieve the most significant conclusions.

As future research, we intend to evaluate this method on a larger number of datasets and compare it against other global hierarchical classification approaches. Although in this paper the HLCS was applied in GPCR and enzymes dataset, it is generic enough to be applied to other hierarchical classification datasets.

Acknowledgments:

We want to thank Dr. Carlos N. Silla Jr. and Dr. Alex A. Freitas for kindly providing us with the datasets used in these experiments.

References

[1] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 85, no. 8, pp. 2444–8, Apr. 1988.

[2] S. F. Altschul, T. L. Madden, A. A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped blast and psi-blast: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[3] R. D. King, P. H. Wise, and A. Clare, "Confirmation of data mining based predictions of protein function," *Bioinformatics*, vol. 20, no. 7, pp. 1110–1118, May 2004.

[4] A. A. Freitas, D. C. Wieser, and R. Apweiler, "On the importance of comprehensible classification models for protein function prediction," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 7, no. 1, pp. 172–182, Jan. 2010.

[5] R. T. Alves, M. R. Delgado, and A. A. Freitas, "Multi-label hierarchical classification of protein functions with artificial immune systems," pp. 1–12, 2008.

[6] C. N. Silla, Jr. and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Min. Knowl. Discov.*, vol. 22, no. 1-2, pp. 31–72, Jan. 2011.

[7] F. Wu, J. Zhang, and V. Honavar, "Learning Classifiers Using Hierarchically Structured Class Taxonomies." *Lecture notes in computer science*, vol. 3607, pp. 313–320, Jan. 2005.

[8] A. A. Freitas and A. C. P. F. L. de Carvalho, *A Tutorial on Hierarchical Classification with Applications in Bioinformatics*. Idea Group, January 2007, ch. VII, pp. 175–208.

[9] B. Vroling, M. P. A. Sanders, C. Baakman, A. Borrmann, S. Verhoeven, J. P. G. Klomp, L. Oliveira, J. de Vlieg, and G. Vriend, "Gpcrdb: information system for g protein-coupled receptors," *Nucleic Acids Research*, vol. 39, no. Database-Issue, pp. 309–319, 2011.

[10] K. F. Tipton and S. Boyce, "History of the enzyme nomenclature system," *Bioinformatics*, vol. 16, no. 1, pp. 34–40, 2000.

[11] D. Filmore, "It's a gpcr world," *odern Drug Discovery*, pp. 24–27, 2004.

[12] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[13] J. Hurst and L. Bull, "A self-adaptive neural learning classifier system with constructivism for mobile robot control," in *Parallel Problem Solving from Nature - PPSN VIII*, ser. Lecture Notes in Computer Science, X. Yao, E. Burke, J. Lozano, J. Smith, J. Merelo-Guervs, J. Bullinaria, J. Rowe, P. Tino, A. Kabn, and H.-P. Schwefel, Eds. Springer Berlin / Heidelberg, 2004, vol. 3242, pp. 942–951.

[14] S. W. Wilson, "Classifier fitness based on accuracy," *Evol. Comput.*, vol. 3, no. 2, pp. 149–175, Jun. 1995.

[15] P. L. Lanzi and D. Loiacono, "Classifier systems that compute action mappings," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, ser. GECCO '07. New York, NY, USA: ACM, 2007, pp. 1822–1829.

[16] A. Hamzeh and A. Rahmani, "A new architecture of xcs to approximate real-valued functions based on high order polynomials using variable-length ga," in *Proceedings of the Third International Conference on Natural Computation - Volume 03*, ser. ICNC '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 515–519.

[17] A. Orriols-Puig, J. Casillas, and E. Bernadó-Mansilla, "Fuzzy-ucs: preliminary results," in *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, ser. GECCO '07. New York, NY, USA: ACM, 2007, pp. 2871–2874.

[18] M. V. Butz, T. Kovacs, P. L. Lanzi, and S. W. Wilson, "Toward a theory of generalization and learning in xcs," *IEEE Trans. Evolutionary Computation*, pp. 28–46, 2004.

[19] M. V. Butz, D. E. Goldberg, and W. Stolzmann, "Introducing a genetic generalization pressure to the anticipatory classifier system – part i: Theoretical approach," in *Proceedings of the 2000 Genetic and Evolutionary Computation Conference (GECCO)*. Morgan Kaufmann, 2000, pp. 34–41.

[20] E. Bernadó-Mansilla and J. M. Garrell-Guiu, "Accuracy-based learning classifier systems: models, analysis and applications to classification tasks," *Evol. Comput.*, vol. 11, no. 3, pp. 209–238, Sep. 2003.

[21] L. M. Romão and J. C. Nievola, "Prediction of Protein Function Using Learning Classifier Systems," *IADIS International Conference on Applied Computing*, pp. 395–401, 2011.

[22] C. N. Silla and A. A. Freitas, "A Global-Model Naive Bayes Approach to the Hierarchical Prediction of Protein Functions," *2009 Ninth IEEE International Conference on Data Mining*, pp. 992–997, Dec. 2009.

[23] S. Kiritchenko, S. Matwin, and A. F. Famili, "Functional annotation of genes using hierarchical text categorization," in *in Proc. of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology (held at ISMB-05*, 2005.