35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

A comparison of multi-label techniques based on problem transformation for protein functional prediction

A. F. Giraldo-Forero[1], J. A. Jaramillo-Garzón[1,2], and C. G. Castellanos-Domínguez[1]

***Abstract*— A comparative analysis of four multi-label classification methods is performed in order to determine the best topology for the problem of protein function prediction, using support vector machines as base classifiers. Comparisons are done in terms of performance and computational cost of parallelized versions of the algorithms, for determining its applicability in high-throughput scenarios. Results show that the performance of the binary relevance strategy, together with a technique of class balance, remains above several recently proposed techniques for the problem at hand, while employing the smallest computational cost when parallelized. However, stacked classfiers and chain clasifications can be conveniently used in pipelines, due to the low number of false positives reported.**

***Index Terms*— Bioinformatics, Multi-label learning, Protein annotation, Support Vector Machines.**

I. INTRODUCTION

The exponential growth of information derived from sequenced genomes, and so the number of protein sequences with missing annotations increases rapidly. Cosequently,functional annotation of proteins has become one of the central problems in molecular biology. Manually curating of annotations turns out to be impossible because of the large amount of data. Thus, the need for computational tools allowing to automate functional annotations has continued to rise in recent years.

Automatic functional annotation of proteins has followed three main approaches: homology-based methods, subsequence-based methods, and feature-based methods. In homology-based methods, query proteins are searched against public databases using local alignment search tools such as BLAST or PSI-BLAST and annotations with the highest scoring hits are transferred onto the new sequence [1]. Despite some known drawbacks such as low sensitivity, and propagation of database errors, this approach is the most widely used among biologists, because as it is historically the first successful method. Subsequence-based methods, search for highly conserved sub-sequences that could be related to protein functionality. To this end, it is common to use stochastic models describing protein families. Nowadays large collections of protein families and domains can be found in databases like [2], where the families are represented by *Hidden Markov Models* (HMM). These approaches, however, tend to have low specificity [3].

[1] Signal Processing and Recognition Group, Universidad Nacional de Colombia, s. Manizales, Campus La Nubia, km 7 vía al Magdalena, Colombia. {afgiraldofo, jajaramillog, cgcastellanosd}@unal.edu.co

[2] The research center of the Instituto Tecnológico Metropolitano, Calle 73 No 76A-354, Medellín, Colombia. {jorgejaramillo }@itm.edu.co

Feature-based methods compute a set of numerical features from protein sequences and search for a mathematical function, known as classifier, that correctly assigns new proteins to their true classes from the computed feature space. Since proteins can be associated to multiple functional categories at the same time, current machine learning methods commonly use binary relevance strategies, that is, one classifier is trained in recognizing each class in an independent way [3], [4]. However, this strategy does not consider correlations among classes and, consequently, can miss potentially important information [5]

Multilabel learning is a branch of machine learning where multiple target labels must be assigned to each instance. Multilabel learning methods can be grouped into two categories: problem transformation and algorithm adaptation [6]. Methods of the first group transform the learning task into one or more single-label classification problems by employing several topologies [4], [7], [8], [9]. The second group of methods extends specific learning algorithms in order to handle multilabel data directly [10], [11]. In this context, problem transformation methods provide major flexibility since they can be easily implemented from traditional learning algorithms and thus users are able to employ standard software packages. Furthermore, high-throughput methods can be easily integrated, which is essential for the scientific community working in Biomedical and Bioinformatics applications, mainly in genomics and proteomics.

This paper presents a comparative analysis aimed to determine the best topology for multi-label classification based on problem transformation strategies, for the problem of protein function prediction. Comparisons are done in terms of performance and computational cost, over four different topologies: Binary Relevance, Pairwise Comparisson, Chain Classifications, and Stacked Classifiers. In all cases, support vector machines are used as baseline classifiers.

II. MATERIALS AND METHODS

The notations that will be used throughout this paper are defined as follows. Consider a classification problem where each instance ($\boldsymbol{x} \in \mathcal{X}$ can be associated with one or more of Q possible classes. Then, let $T = \{\boldsymbol{X}, \boldsymbol{Y}\}$ be the training set, where $\boldsymbol{X}$ is the *feature matrix*, containing the training instances $\boldsymbol{x}_n$, $n = 1, 2, ..., N$ in its rows, while $\boldsymbol{Y}$ is the *label matrix*, with each row being a binary vector $\boldsymbol{y}_n = \{y_n^1, y_n^2, ..., y_n^q, ..., y_n^Q\}$ with $y_n^q \in \{1, -1\}$ indicating wether or not the n-th instance must be associated to de q-th class. The goal of the multilabel classification is to use the information in T for obtaining a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$, that correctly assigns a subset of labels to new instances.

Methods for multi-label classification based on the transformation of the problem, define different topologies for decomposing h into a set of binary classifiers h_k, $k = 1, 2, ..., K$ in order to better explode the information contained in the training set.

A. Binary Relevant (BR)

In this topology, a number of classifiers equal to the number of classes is trained $(K := Q)$. For training each classifier, the whole feature matrix is used $\boldsymbol{X}_k := (X)$, while only the q-th column of the label matrix is considered. This way, the set of labels for each instance is redefined as $\boldsymbol{Y}_k = \boldsymbol{y}^k$. Therefore, the following holds:

$$h_k : \mathcal{X} \rightarrow \{1, -1\}$$

So, each binary classifier h_k predicts one of the labels associated with the instances.

B. Pairwise Comparison (PC)

In this topology, one binary classifier is trained for each pair of classes. So, let $P = \{(1,2), (2,1), (2,3), ..., (p,q), ..., (Q, Q-1)\}$ be the set of all $2-$permutations of the set of numbers $1, \dots, Q$. The total number of classifiers will then be $K := |P| = Q(Q-1)$. Let suppose now that the k-th classifier h_k is associated to the pair of classes (p, q). Then, for training such a classifier, the feature matrix will include only the rows corresponding to the instances related to those classes, that is, the rows of the feature matrix $\boldsymbol{X}_k$ will be those instances $\{\boldsymbol{x}_n | (y_n^q = 1) \wedge (y_n^p = 1)\}$. In this case, the rows of the label matrix will be assigned as $\boldsymbol{Y}_k = \boldsymbol{y}^p$. Note that the classifiers associated to the pairs (p, q) and (q, p) will have the same feature matrix but will differ in their label matrices.

Finally, as there are $Q(Q-1)$ class assignments for each instance, the Q labels are selected using a voting scheme. Each label is considered to be true if the number of votes for that class is higher than a predefined threshold that maximizes a given performance measure. This approach is known as *OneThreshold* in [7].

C. Chain Clasifications (CC)

In this scheme, classifiers are trained in a predefined order. As in *Binary Relevant* it is necessary to build $K := Q$ classifiers, but this time, the feature matrix for the $k-$th classifier is enriched with the output of the previous one.

That is,

$$\boldsymbol{X}_1 := \boldsymbol{X}, \boldsymbol{X}_2 = [\boldsymbol{X}_1, h_1(\boldsymbol{X}_1)], \dots,$$
$$\boldsymbol{X}_k = [\boldsymbol{X}_{k-1}, h_{k-1}(\boldsymbol{X}_{k-1})], \dots,$$
$$\boldsymbol{X}_K = [\boldsymbol{X}_{K-1}, h_{K-1}(\boldsymbol{X}_{K-1})]$$

This way, each classifier will be designed as:

$$h_k : \mathcal{X} \times \{1, -1\}^{j-1} \rightarrow \{1, -1\}$$

Predictions are done by successfully applying classifiers in the order of the chain [8].

D. Stacked Classifiers (STA)

In this scheme, two levels of classifiers are constructed. For the first level, denoted by $\boldsymbol{h}^1$, *Binary Relevant* method is used. The second level is denoted as $\boldsymbol{h}^2$, and is constructed by including the predictions of the previous level in the feature set, that is,

$$\boldsymbol{X}_1^2 = [\boldsymbol{x}, h_2^1(\boldsymbol{X}), \dots, h_Q^1(\boldsymbol{x})), \dots,$$
$$\boldsymbol{X}_Q^2 = [\boldsymbol{x}, h_1^1(\boldsymbol{X}), \dots, h_{Q-1}^1(\boldsymbol{x}))$$

Therefore the classifier for the k-th class in the second level will be in the form [9]:

$$h_k^2 : \mathcal{X} \times \{1, -1\}^{Q-1} \rightarrow \{1, -1\}$$

III. Experimetal Setup

The workflow of the experimental setup for each baseline classifier has three main components: *Database*, which comprises the construction and pre-processing of the dataset; *parameter tuning*, comprising the steps for searching optimal parameters for the classifier, and *classification*, which describes training and testing of the models.

Figure 1 illustrates the wokflow of the process developed for each baseline classifier. Ovals, squares and diamonds are used to depict datasets, computational processes, and conditional statements, respectively.

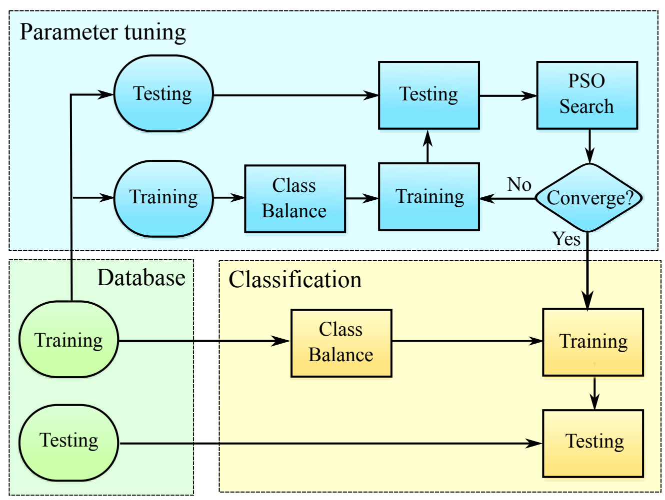

Fig. 1. Scheme of the baseline classifiers

A. Database

The database is comprised of ten different classes corresponding to the ontology *molecular function*, grouping 2326 proteins belonging to the *Embryophyta* taxonomy of the Uniprot database [12] with at least one annotation in the Gene Ontology Annotation project [13]. Proteins with unknown evidence of the existence or resulting from computational predictions were discarded. Aiming to avoid overtraining, the dataset does not contain protein sequences with a sequence identity superior to 40%, which were discarded by employing the CD-HIT software package [14]. After that, categories with less than 100 sequences were discarded in

TABLE I

NUMBER OF PROTEIN SEQUENCES PER CLASS

Functions	Samples	Functions	Samples
DnaBind*	143	ProtBind	1117
TranscFact	102	Kinase	103
Catal*	401	Transf*	217
Transp	133	Hydrol*	237
Bind*	194	TranscReg*	152

order to ensure statistically significant results. The number of sequences per class is shown in I. Proteins were characterized according to the schema used in [4].

B. Parameter tuning

Support vector machines (SVM) are used as baseline classifiers and, consequently their free parameters must be properly tuned. Such tuning is carried out by a *Particle Swarm Optimization* (PSO) meta-heuristic [15] which explores a two-dimensional search space generated by all the possible pairs of values that can be assigned to the trade-off constant of the SVM (C) and the dispersion parameter of the gaussian kernel (σ). To this end, a new partition on the training set is done following a cross-validation of ten folds, in order to avoid over-training of the models. Each resulting training set is balanced by *Synthetic Minority Oversampling Technique* (SMOTE) [16]. The limits of the search space were defined as (10^{-2},10^{4}) for σ and (1,10^{-7}) for C. Additionally, the number of particles for the search was set to 10, while the maximum number of iterations was set to 30.

C. Classification

Due to the nature of the problem and the transformation methods, a high class imbalance in binary classifiers is induced. If untreated, it could seriously deteriorate the sensitivity of the prediction. For this reason, a method of oversampling called SMOTE was used. The main advantage of this method is that prevents excess of adjustment commonly caused by random over-sampling, since synthetic samples are not exact copies of the original ones.

Classification is implemented following the strategies described in the section II with support vector machines (SVM) as base classifiers. All results are derived from a 10-fold cross-validation, using the parameters of the SVM that were tuned in the previous stage.

IV. RESULTS AND DISCUSSION

Table III shows the performance of each strategy over the whole set of classes. Best results for each metric on each class are highlighted in boldface. The sensitivity (S_n), specificity (S_p), geometric mean (G_m) and, Matthews correlation coefficient (M_{cc}) are used as classification performance measures:

$$S_n = \frac{n_{TP}}{n_{TP}+n_{FN}} \qquad S_p = \frac{n_{TN}}{n_{TN}+n_{FP}}$$

$$G_m = \sqrt{\frac{n_{TP}n_{TN}}{(n_{TP}+n_{FN})(n_{TN}+n_{FP})}}$$

$$M_{cc} = \frac{n_{TP}n_{TN} - n_{FP}n_{FN}}{\sqrt{(n_{TP}+n_{FP})(n_{TP}+n_{FN})(n_{TN}+n_{FP})(n_{TN}+n_{FN})}}$$

Being n_{TP}, n_{FP}, n_{TN}, and n_{FN} the true positive, false positive, true negative and false negative, respectively. Additionally, in order to analyze the applicability of each strategy over high-throughput tasks, Table II presents the time in seconds for the training stage on one of the partitions. These times are measured in its parallelized versions: notation *Classifiers* denotes the number of parallel processes compatible with the topology, while notation *Cores* indicates the number of threads that are used in practice, given to the characteristics of the machine; a number of 20 threads was used as limit value for parallel processing. The tests were performed using a dual Intel® Xeon X5660 with 12 cores at 2.8 *GHz,* under a Linux machine and without limitations of ram. The scripts were implemented using the R Project for Statistical Computing.

Since BR topology has been considered as a *naive approach to multi-label learning* because correlations between classes are ignored [5], new proposals have emerged in order to take account of these correlations. In CC and STA the correlations are considered by incorporating information from the labels of the other classes as input to subsequent stages of classification. The results in table III show that incorporating the label information, rises specificity, but seriously degrades sensitivity. This is evident in the Table III where the sensitivity of STA is lower than the one reached by BR for all classes. On the other hand, for CC the increase of specificity and consequent decrease of sensitivity occurs gradually according to the order of defined chain. Due to this order, the classes *Catal*, *Tranf*, and *Hydrol* have the lowest sensitivity, since they are the last ones in the chain. The chain is defined taking the results in descending order acquired by BR in G_m, thus leaving: *ProtBind*, *Transp*, *TranscFact*, *TranscReg*, *DnaBind*, *Kinase*, *Transf*, *Catal*, *Bind*, and *Hydrol*.

This loss of sensitivity is consistent with the fact that new topologies are designed to minimize the hamming loss [7], [8], [9], which causes the system to have high accuracies without regarding the class membership of correctly classified instances. This is, however, a misleading measure when classes are not equal in size, since instances of the target class represent a minor percentage of the total size of the dataset. As a result, they emphasize on the specificity, while causing a loss of sensitivity. In the process of functional annotation of proteins, however, it is important to obtain high specificities and sensitivities together, i. e. to maximize a balanced measure such as geometric mean or Matthews correlation coefficient. This is clearly accomplished with the BR strategy, also with the advantage that it is much faster than the other topologies studied.

V. CONCLUSION

A comparison of four of the most relevant multilabel classification methods, based on *problem transformation* was

TABLE III

S_n, S_p, G_m, AND M_{cc} VALUES OVER 10 FUNCIONAL CLASSES

Function	**Sensitivity**				**Specificity**				**Geomatric Mean**				**Matthews Correlation Coefficient**			
	BR	PC	CC	STA	BR	PC	CC	STA	BR	PC	CC	STA	BR	PC	CC	STA
DnaBind*	**0.818**	0.790	0.392	0.504	0.804	0.764	**0.924**	0.908	**0.811**	0.777	0.602	0.676	**0.353**	0.3	0.258	0.308
TranscFact	**0.922**	0.775	0.667	0.099	0.818	0.831	0.705	**0.979**	**0.868**	0.802	0.685	0.31	**0.369**	0.312	0.164	0.103
Catal*	0.736	**0.923**	0.274	0.177	0.696	0.353	0.872	**0.975**	**0.716**	0.571	0.489	0.416	**0.336**	0.226	0.154	0.261
Transp	**0.82**	0.774	0.752	0.692	0.938	0.921	0.944	**0.973**	**0.877**	0.844	0.842	0.820	0.574	0.498	0.549	**0.627**
Bind*	**0.717**	0.701	0.536	0.541	0.709	0.646	0.788	**0.942**	0.713	0.673	0.65	**0.714**	0.251	0.197	0.210	**0.45**
ProtBind	0.978	**0.983**	0.978	0.167	**0.969**	0.056	**0.969**	0.943	**0.976**	0.235	**0.976**	0.396	**0.948**	0.103	**0.948**	0.175
Kinase	**0.864**	0.447	0.534	0.427	0.748	0.812	0.774	**0.876**	**0.804**	0.602	0.643	0.612	**0.280**	0.133	0.148	0.182
Transf*	**0.816**	0.793	0.198	0.129	0.722	0.610	0.914	**0.986**	**0.767**	0.696	0.426	0.357	**0.333**	0.237	0.111	0.217
Hydrol*	0.713	**0.738**	0.072	0.190	0.656	0.551	0.913	**0.978**	**0.684**	0.638	0.256	0.431	0.23	0.175	-0.016	**0.26**
TranscReg*	**0.888**	0.796	0.493	0.566	0.771	0.769	**0.890**	0.874	**0.827**	0.782	0.663	0.703	**0.366**	0.315	0.277	0.301
	0.827	0.772	0.459	0.349	0.783	0.631	0.839	**0.944**	**0.804**	0.662	0.592	0.544	**0.404**	0.25	0.219	0.288

TABLE II

DETAILS IN THE PARALLELIZATION

Details	**Topology**			
	BR	PC	CC	STA
Times	2270.6	2574.2	32286.5	3634.27
Cores	10	20	1	20
Classifiers	10	90	1	20

carried out, in order to identify the most suitable topology classification to the problem of protein function prediction. The methods were compared by specificity and sensitivity as diagnostic measures and the geometric mean and the Matthews correlation coefficient as average overall performances. Additionally, the training time of each strategy in their parallelized versions was measured as an indicator of their feasibility to be used for high-throughput tasks.

The results show that the best topology in terms of global classification performance is BR, which also shows the smallest computational cost when parallelized. However, STA and CC can be conveniently used in pipelines, due to the low number of false positives reported.

As future work, generate a classification scheme to capture the correlations while maintaining linear complexity front to the classes, the same way that BR, and additionally compagnie with the fact of having unbalanced databases.

VI. ACKNOWLEDGMENTS

This work is within the framework of the Dirección de Investigaciones de Manizales (DIMA) of the Universidad Nacional de Colombia and the Centro de Investigación of the Instituto Tecnológico Metropolitano. This research has been partially founded by Colciencias grant 111952128388 and by Jóvenes Investigadores e Innovadores 2011, with the convenio especial de cooperación No. 0043.

REFERENCES

[1] A. Conesa and S. Götz, “Blast2GO: A Comprehensive Suite for Functional Analysis in Plant Genomics.” *International journal of plant genomics*, vol. 2008, p. 619832, 2008.

[2] M. Punta, P. C. Coggill, R. Y. Eberhardt, J. Mistry, J. Tate, C. Boursnell, N. Pang, K. Forslund, G. Ceric, J. Clements, A. Heger, L. Holm, E. L. L. Sonnhammer, S. R. Eddy, A. Bateman, and R. D. Finn, “The Pfam protein families database.” *Nucleic acids research*, vol. 40, no. Database issue, pp. D290–301, Jan. 2012.

[3] H. Oul and E. U. Mumcuoğlu, “SVM-based detection of distant protein structural relationships using pairwise probabilistic suffix trees.” *Computational biology and chemistry*, vol. 30, no. 4, pp. 292–9, Aug. 2006.

[4] J. A. Jaramillo-Garzón, A. Perera-Lluna, and C. G. Castellanos-Domínguez, “Predictability of protein subcellular locations by pattern recognition techniques.” in *Annual International Conference of the IEEE EMBS*, vol. 2010. IEEE, Jan. 2010, pp. 5512–5.

[5] M. Petrovskiy, “Paired Comparisons Method for Solving Multi-Label Learning Problem,” *2006 Sixth International Conference on Hybrid Intelligent Systems (HIS’06)*, pp. 42–42, Dec. 2006.

[6] G. Tsoumakas and I. Katakis, “Multi-label classification: An overview,” *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[7] M. Ioannou, G. Sakkas, G. Tsoumakas, and I. Vlahavas, “Obtaining Bipartitions from Score Vectors for Multi-Label Classification,” *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, pp. 409–416, Oct. 2010.

[8] J. Read, B. Pfahringer, G. Holmes, and E. Frank, “Classifier chains for multi-label classification,” *Machine Learning and Knowledge Discovery in Databases*, pp. 254–269, 2009.

[9] E. Montanés, J. Quevedo, and J. del Coz, “Una mejora de los modelos apilados para la clasificación multi-etiqueta,” *aepia.aic.uniovi.es*.

[10] Y. Liu, R. Jin, and L. Yang, “Semi-supervised multi-label learning by constrained non-negative matrix factorization,” in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 421.

[11] X. Kong, M. Ng, and Z. Zhou, “Transductive Multi-Label Learning via Label Set Propagation,” *IEEE Transactions on Knowledge and*, pp. 1–14, 2011.

[12] E. Jain, A. Bairoch, S. Duvaud, I. Phan, N. Redaschi, B. E. Suzek, M. J. Martin, P. McGarvey, and E. Gasteiger, “Infrastructure for the life sciences: design and implementation of the UniProt website.” *BMC bioinformatics*, vol. 10, p. 136, Jan. 2009.

[13] T. Gene and O. Consortium, “Gene Ontology: tool for the unification of biology,” *Gene Expression*, vol. 25, no. may, pp. 25–29, 2000.

[14] W. Li and A. Godzik, “Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences.” *Bioinformatics (Oxford, England)*, vol. 22, no. 13, pp. 1658–9, Jul. 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16731699

[15] C. Bendtsen., *pso: Particle Swarm Optimization*, 2011, r package version 1.0.1. [Online]. Available: http://CRAN.R-project.org/package=pso

[16] N. Chawla, K. Bowyer, and L. Hall, “SMOTE: synthetic minority over-sampling technique,” *Journal of Artificial*, vol. 16, 2002.