Committee-based Active Learning to Select Negative Examples for Predicting Protein Functions

Marco Frasca[1 [0000−0002−4170−0922]], Maryam Sepehri[1 [0000−0001−9710−3273]], Alessandro Petrini[1 [0000−0002−0587−1484]], Giuliano Grossi[1 [0000−0001−9274−4047]] and Giorgio Valentini[1 [0000−0002−5694−3919]]

Dipartimento di Informatica, Università degli Studi di Milano, Via Celoria 18, 20135 Milano, Italy
{frasca,sepehri,grossi,valentini}@di.unimi.it,
alessandro.petrini@unimi.it

Abstract. The Automated Functional Prediction (AFP) of proteins became a challenging problem in bioinformatics and biomedicine aiming at handling and interpreting the extremely large-sized proteomes of several eukaryotic organisms. A central issue in AFP is the absence in public repositories for protein functions, e.g. the Gene Ontology (GO), of well defined sets of negative examples to learn accurate classifiers for AFP. In this paper we investigate the Query by Committee paradigm of active learning to select the negatives most informative for the classifier and the protein function to be inferred. We validated our approach in predicting the Gene Ontology function for the *S.cerevisiae* proteins.

Keywords: Query by Committee, active learning, protein function prediction.

1 Scientific Background

The Automated Function Prediction (AFP) of proteins involves sophisticated computational techniques to accurately predict the annotations of proteins and proteomes. AFP is characterized by several issues, including the selection of negative examples to train high quality predictors. The Gene Ontology (GO) [1], the reference repository of protein functions, usually stores positive associations (also referred to as annotations) between GO terms and proteins, whereas unannotated proteins are rarely marked as negative for a given term — a protein not currently annotated with a GO term, might be a positive example which has not been detected yet due to insufficient studies. Surprisingly, a few works investigated this problem, mainly leveraging the GO structure (a directed acyclic graph) to choose negative examples. Since in the GO the true path rule (TPR) holds, which transfers annotations for a node to all its ancestors, initial approaches considered as negative examples for a term all proteins directly annotated (i.e., before applying the TPR) in neither descendant nor ancestral

terms [2]. Indeed, proteins annotated with descendants belong also to the current term (descendants specialize the node concept), and annotations (direct) with ancestors might be transferred to some descendants in light of future studies. Under the assumption that proteins are rarely annotated with more than one child of the same node, proteins annotated with sibling terms (i.e. terms sharing at least one parent) have also been adopted as negative examples [3]. More recent works selected negatives based on the empirical conditional probability of annotating a protein with the GO function of interest given all the annotations that protein had with all the other functions [4], or just with the most specific ones [5], in all three branches. Lastly, in [6, 7] authors assessed the relevance of several protein features, extracted from protein networks, in detecting false negatives using a GO temporal holdout setting, thus providing insights about how to effectively select negative proteins.

We present here a preliminary work which addresses the negative selection problem by leveraging Query by Committee (QBC) active learning (AL) [9] to appropriately select the negative proteins. Unlike usual approaches to active learning, which typically aim to obtain the true labels of some selected data points, our approach undertakes the selection of negative examples (whose labels are obviously known). The rationale behind this approach is that the capability of active learning to focus on the most informative examples can be leveraged to filter out from the training set unhelpful non-positive proteins — or even harmful. Pool-based QBC considers most informative the examples from a pool of unlabeled examples on which the committee members (classifiers) most disagree. Hence, in our setting QBC is used to select as negative examples a subset of proteins from the pool represented by non-positive proteins. We experimentally validated our approach using two well-known classifiers to predict, in a genome-wide fashion, the GO functions of S. cerevisiae (yeast) proteins.

2 Preliminaries and notations

Vectors and matrices are denoted using standard, lower bold and upper bold symbols as $\boldsymbol{x}$ and $\boldsymbol{X}$. Protein pairwise similarities are represented by an undirected weighted graph $G\langle V, \boldsymbol{W}\rangle$, where $V = \{1, \ldots, n\}$ is the set of nodes/proteins and $\boldsymbol{W}$ is the $n \times n$ matrix of intra-protein functional similarity: $W_{ij} \in [0, 1]$ is the similarity between proteins $i, j \in V$, with $W_{ij} = 0$ when i and j are not connected. Given a protein function, the labels are described by the binary vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$, where $y_i = 1$ if protein i is annotated with that function (positive instance), -1 otherwise. Here the GO terms are adopted as protein functions. Let $V_+ := \{i \in V | y_i = 1\}$ and $V_- := \{i \in V | y_i = -1\}$ be the subsets of positive and non positive proteins, respectively. A relevant issue in AFP is the labeling imbalance: most GO functions posses a highly unbalanced labeling, that is $\frac{|V_+|}{|V_-|} \ll 1$. Furthermore, the labeling is known only for a subset $S \subset V$ of proteins, where it is unknown for its complement set $U := V \setminus S$. We consequently denote by $S_+ := S \cap V_+$ and $S_- := S \cap V_-$ the known sets of positive and non positive proteins for a given GO term, respectively.

The *Automated protein Function Prediction* (AFP) problem consists in inferring the labeling for proteins U using the known labels and the connection matrix $\boldsymbol{W}$.

The complexity of AFP is increased by the fact that the GO rarely stores *negative* annotations between proteins and functions, and only positive annotations are usually available. Thus, non positive proteins (proteins in S_-) typically do not correspond to *negative* annotations, and some of them might be redundant for the current task. Moreover, some non positive proteins might become positive in future, in case further studies would annotate them. This makes central the need to select informative negatives among non positive proteins to be used as negative examples during the learning of automated models for solving AFP — indirectly, it would also cope with the label imbalance, since the disproportion between positives and negatives would be reduced.

2.1 Instance representation.

Following [10], the input proteins are represented through a two-dimensional feature vector, obtained by operating a projection of nodes S onto the space $\mathbb{R}^2$, so that the node $i \in S$ is associated with the point $\boldsymbol{x}_i \equiv (x_{i1}; x_{i2})$, where $x_{i1} = \sum_{j \in S_+} W_{ij}$ and $x_{i2} = \sum_{j \in S_-} W_{ij}$. This embedding casts into the position of point $\boldsymbol{x}_i$ the imbalance in the neighborhood of protein i in the graph, and sensibly reduces the input space dimension, thus speeding up the computation. Moreover, recent studies have confirmed that this two features are informative for inferring GO functions [6, 7]. The obtained training set is $L = \{(\boldsymbol{x}_i, y_i) | i \in S\}$.

2.2 Data

We retrieved the yeast protein network from the STRING database, version 10.5 [11], which merges several sources of information about proteins, including experimental data, such as GRID, HPRD, IntAct, and curated data, like Biocarta and KEGG. The matrix $\boldsymbol{W}$ of Section 2 is obtained from the STRING connections $\widehat{\boldsymbol{W}}$ after the normalization $\boldsymbol{W} = \boldsymbol{D}^{-1/2} \widehat{\boldsymbol{W}} \boldsymbol{D}^{-1/2}$, which preserves the connection symmetry. $\boldsymbol{D}$ is the diagonal matrix with non-null elements $d_{ii} = \sum_j \widehat{W}_{ij}$. As suggested by STRING curators, we set the threshold for connection weights to 700. The final network contains 6391 proteins. Annotations for the three GO branches, namely Biological Process (BP), Molecular Function (MF), and Cellular Component (CC), have been downloaded from the UniProt GOA, release 69 (9 May 2017), by retaining solely experimentally validated annotations. To discard too generic terms and having a minimum of information to learn, we chose functions with 10–100 annotations, obtaining 162, 227 and 660 terms for CC, MF, and BP branches, respectively.

3 Algorithm

We propose a novel approach to address the negative selection in AFP, which leverages a variant of Query by Committee active learning to appropriately select

the most informative negative examples. In particular, our technique focuses on the selection of most informative negatives for the specific classification model, rather than selecting those negative examples "most informative" in general. We empirically validated our proposal on two well-known supervised classifiers.

3.1 QBC active learning for negative selection

Let $0 < B < |S_-|$ be the cardinality of a subset of negative examples $\widehat{S}_- \subset S_-$, which has to be selected in order to maximize the performance of a classifier trained using the examples $S_+ \cup \widehat{S}_-$. B is also referred to as *budget* of the negative selection algorithm.

This problem is tackled through a pool-based QBC-AL algorithm, which typically examines a pool of unlabeled examples and selects only those that are most informative according to the committee models (classifiers in our setting), and asks for their labels. This avoids to save annotation cost by discarding redundant labeling examples that contribute little new information [9]. Common approaches for pool-based QBC is to ask the label of those points on which committee members most disagree [12].

We adopt a variant of active learning, since we want the QBC algorithm to select instances whose label is known already (equal to -1). Nevertheless, we may exploit AL to pick out the "most informative" negative points for training our model. Our QBC-AL algorithm is defined as follows.

QBC-AL procedure (template).

1. A seed training set $I(0) = S_+ \cup S_-(0)$ is selected, where $S_-(0) \subset S_-$ is randomly drawn and balanced (i.e., $|S_-(0)| = |S_+|$).[1] Due to the rarity of positives, $I(0)$ contains all available positives.
2. At iteration $t \geq 1$, learn m committee models $f_k : I(t-1) \rightarrow \{-1, 1\}$, $k \in \{1, 2, \ldots, m\}$.
3. Build $I(t)$ by adding to $I(t-1)$ the s instances in $S_- \setminus I(t-1)$ with highest degree of disagreement among the committee members.
4. Update the committee classifiers using $I(t)$.
5. Iterate steps 2–4 until time $\bar{t}$, with $|I(\bar{t})| = |S_+| + B$ (budget is exhausted).

The rationale is that examples on which the classifiers most disagree have a higher 'utility' for the committee. Further, it is beneficial in QBC ensuring diversity among committee classifiers [13]: a common approach is to use bagging for learning the m committee classifiers [14], in which m random subsets $I_1, I_2, \ldots, I_m$ of $I(\bar{t})$ are randomly drawn, and the member f_k is trained using the set I_k. In our setting each I_i contains all available positives and a randomly drawn subset of non positive examples in $I(\bar{t})$.

The Vote Entropy has been employed as measure of disagreement, a natural measure for quantifying the uniformity of classes assigned to an example by the different committee members [12]. Given an instance $\boldsymbol{x} \in \mathbb{R}^q$, its

[1] A balanced seed training set counterbalances the predominance of -1 labels, and experimentally performed best

Vote Entropy disagreement is $V(\boldsymbol{x}) = -\nu_x \log \nu_x - (1-\nu_x)\log(1-\nu_x)$, where $\nu_x = \frac{\sum_{k=1}^{m} \mathbb{I}\{f_k(\boldsymbol{x})=1\}}{m}$, and $\mathbb{I}$ is the indicator function. Thus ν_x is the proportion of members that predicted $\boldsymbol{x}$ as positive. Accordingly, the closer ν_x to 0.5, the higher the Vote Entropy disagreement.

We validated the QBC-AL algorithm to select negatives in AFP by adopting two popular feature-based models at Step 2 of the procedure, which are briefly described below.

Support Vector Machines. Given a training set $L = \{(\boldsymbol{x}_i, l_i)\} \in \mathbb{R}^q \times \{-1, 1\}$, the Support Vector Machine (SVM) [15, 16] learns the hyperplane $\hat{\boldsymbol{\omega}} \in \mathbb{R}^q$ unique solution of the following optimization problem:

$$\min_{\boldsymbol{\omega} \in H_{\mathcal{K}}} \frac{1}{2}\|\boldsymbol{\omega}\|^2_{\mathcal{K}} + \ C \sum_{i}^{|L|} e_i(\boldsymbol{\omega}) \tag{1}$$

where $e_i(\boldsymbol{\omega}) = 1 - l_i\langle \boldsymbol{\omega}, \phi_{\mathcal{K}}(\boldsymbol{x}_i)\rangle$, if $l_i\langle \boldsymbol{\omega}, \phi_{\mathcal{K}}(\boldsymbol{x}_i)\rangle < 1$ (margin constraint violation), 0 otherwise, and $\boldsymbol{\mathcal{K}}$ is a kernel implementing the inner product $\mathcal{K}(\boldsymbol{x}, \boldsymbol{z}) = \langle \phi_{\mathcal{K}}(\boldsymbol{x}), \phi_{\mathcal{K}}(\boldsymbol{z})\rangle$ of two vectors $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^d$ according to a feature map $\phi_{\mathcal{K}} : \mathbb{R}^d \rightarrow H_{\mathcal{K}}$. $H_{\mathcal{K}}$ is suitable high dimensional space. The margin of an instance $\boldsymbol{x}_i$ is $\left|\sum_{j=1}^{|L|} \alpha_j l_j \mathcal{K}(\boldsymbol{x}_j, \boldsymbol{x}_i)\right|$, where $\alpha_j \geq 0$ are the Lagrange multipliers (see for instance [16]).

Two popular choices of $\mathcal{K}$ are adopted in this work, namely the *linear* kernel $\mathcal{K}_1(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}^T\boldsymbol{z}$, and the *Gaussian* kernel $\mathcal{K}_2(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2})$.

Decision Trees. Let $X_1, \ldots, X_q$ be q predictor variables (discrete or continuous), $L = \{(\boldsymbol{x}_i, l_i)\} \in \mathbb{R}^q \times \{-1, 1\}$ be a set of labeled observations on a class variable Y (binary in our case) that takes values -1, 1. Briefly, the decision tree (DT) algorithm [17] learns a model $T : \mathbb{R}^q \rightarrow \{-1, 1\}$ for predicting the values of Y from observations $\boldsymbol{x}_i$ by simply partitioning the space $\mathbb{R}^q$ into 2 disjoint sets A_-, A_+, such that the predicted value of Y is -1 if $T(\boldsymbol{x}_i) \in A_-$, 1 if $T(\boldsymbol{x}_i) \in A_+$.

Classification tree methods grows from an initial (root) node by recursively partitioning the data set one predictor variable at a time. Each node is assigned a label (-1 or 1), and accordingly it is associated with a classification error (based on labels l_i), used to measure the node impurity. At each step, the node to be split is determined by exhaustively searching the split, e.g. $X_j > t$, over all nodes and predictors X_j which minimizes the total impurity of its two child nodes. Then, during the inference process, an instance $\boldsymbol{x}_i$ at the split node moves to one of the two children according to the value of its j-th component x_{ij} —in our example, if $x_{ij} \leq t$ move to left child, else move to right child). The process iterates till a stopping criterion is met (e.g. maximum depth reached). To predict an instance $\boldsymbol{z}$, the algorithm follows the path from the root to a leaf node, and classify $\boldsymbol{z}$ with the label of that leaf node. As impurity measure a common choice is the Gini index [18], which has been adopted also in this work.

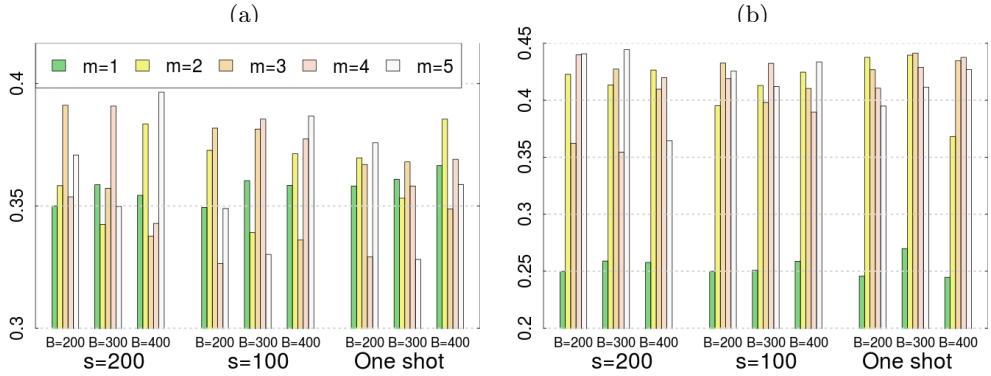

Fig. 1. F values averaged across CC terms for (a) DT and (b) linear SVM algorithms. B is the negative budget, s is the active learning parameter, m the number of committee members.

4 Results

We name SVM QBC (resp. DT QBC) the method using SVMs (resp. DTs) both at step 2 of the QBC-AL procedure and as final model over the set $I(\bar{t})$. Firstly, to evaluate the usefulness of QBC, we implemented an active learning negative selection using only one member (baseline AL, $m = 1$), where the most informative instances are those with smaller margin for SVMs, and those belonging to the leaves with higher impurity for DTs. Generalization capabilities have been evaluated using a 3-fold cross validation (CV) procedure, and measured in terms of F_1 measure (F in short), which is a measure suitable for unbalanced labelings. The model parameters, C for linear SVM, C and σ for Gaussian SVM, and the tree maximum depth for DT, have been learned through inner 3-fold CV.

We first investigated the impact of parameters s and B on the model performance, by tuning them on the CC GO terms. Furthermore, the variant of AL, named *One shot*, has been implemented, in which all the negatives ($B - |S(0)|$) are selected at the first iteration of step 3 of the QBC-AL procedure. There is at least one QBC configuration which outperforms the baseline AL in all the settings, and the improvements are remarkable when using SVM (see Fig. 1). Interestingly, with just 200 negatives, SVM QBC already achieves its top performance, and adding further examples ($B = 300$ or $B = 400$) does not significantly helps. A similar behaviour is shown also for DT QBC. The *one shot* approach seems penalizing more DT-based methods, whereas SVM even in this setting achieves competitive performance with regard to other choices of s. In particular, in this setting, increasing the number of committee members does not help, differently from other choices of s, where in most cases setting $m = 4, 5$ (i.e. the highest number of members tested) corresponds to the best results. This is to some extent expected, since the one shot strategy involves the committee just once, thus reducing the relevance of QBC.

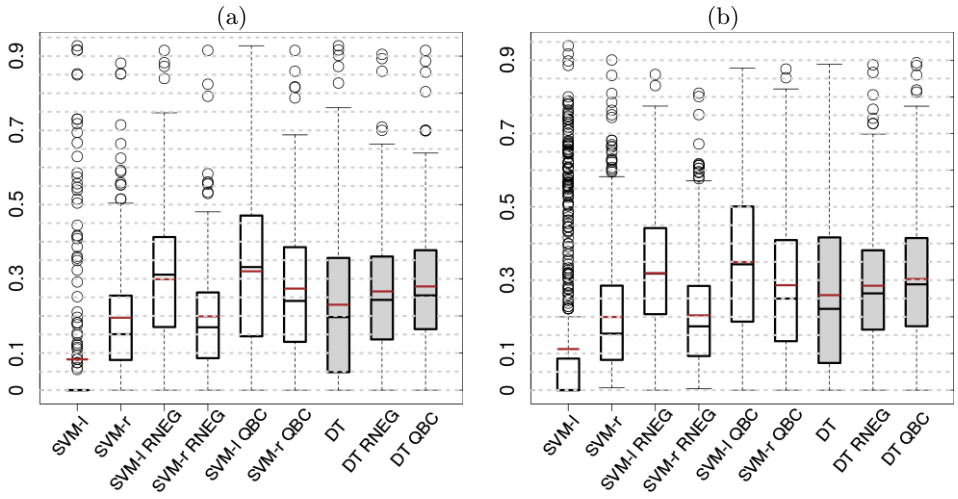

Fig. 2. F values averaged across (a) MF and (b) BP terms. White boxes correspond to SVM methods, gray boxes to DT methods. SVM-l and SVM-r denotes respectively the SVM using linear and Gaussian kernels. Red horizontal segments correspond to mean values.

To assess the effectiveness in AFP for models using QBC, we also tested the vanilla SVM and DT, learned on all available training data (no negative selection applied), and the SVM and DT where the B negative examples are uniformly extracted (named SVM RNEG and DT RNEG). We set $s = 100, B = 400$ for DTs, and $s = 200, B = 300$ for SVMs, and $m = 5$, as consequence of the results obtained in the first experiment (see Fig. 1). QBC negative selection always allows to outperform both the vanilla variant and the variant using random negative selection on MF anc BP classes (see Fig. 2). CC results, not reported, shown an analogous trend. Random negative selection performs better than vanilla methods, confirming the need of negative selection in this context. Noticeable is the case of SVM-l, which is the worst method in the vanilla fashion, but with QBC is the top performing method. The improvements of QBC over RNEG negative selection are always statistically significant according to the Wilcoxon signed rank test ($p\text{-}value < 0.05$) [19].

Finally, to analyze to overall performance of our strategy, we compared the top performing algorithm (SVM-l-QBC, $m = 5$, $s = 200$, $B = 300$) with the state-of-the-art networks-based methods for AFP, including also well-known general-purpose methods, namely the *Random Walk* (RW) [20] and the *Label Propagation* (*LP*) [21] algorithms. The top-performing methods proposed specifically for AFP considered here are the followings: the *guilt-by-association* (GBA) method [22], classifying nodes according to their neighboring functions; the *Cost-Sensitive Neural Network* (COSNet) [10, 23], designed for classifying with unbalanced data, and competitive on the MOUSEFUNC benchmark [24]; the *Multi-Source k-Nearest Neighbors* (MS-kNN) [25], among the top-ranked

Table 1. Averaged AUPR values.

	RW	GBA	LP	MS-kNN	RANKS	COSNet	SVM-l-QBC
BP	0.244	0.145	0.224	0.116	0.271	0.241	0.327
MF	0.199	0.125	0.201	0.090	0.236	0.214	0.293
CC	0.367	0.207	0.308	0.218	0.398	0.361	0.427

methods in the recent CAFA2 international challenge for AFP [26]; the *RAnking of Nodes with Kernelized Score Functions* (RANKS) [27], proposed as effective ranking algorithm for AFP. Free parameters, when present, have been learned through inner 3-fold CV. Since the above-mentioned methods do not provide binary predictions, but just a node ranking, in order to have a fair comparison we adopted as performance measure the Area Under the Precision-Recall curve (AUPR), as recently done in the CAFA2 challenge. For our methodology we naturally obtained a node ranking by considering as final prediction the margin of instances in the final model. As we can see from Table 1, our method achieves the top average results in all the GO branches, and the results are statistically significant ($p - value < 0.05$).

5 Conclusion

Preliminary results have shown that Query by Committee active learning might be employed as effective tool to address the negative selection problem in AFP. Despite the promising results, further studies must be carried out to investigate the impact that several features of the method have on the classification abilities, like the adoption of different measures of committee disagreement among the numerous measures proposed in the literature, and of different stopping criterion than fixing a budget of negatives, along with experimentations on other organisms/datasets.

Acknowledgments

This work was supported by the grant title *Machine learning algorithms to handle label imbalance in biomedical taxonomies*, code PSR2017_DIP_010_MFRAS, Università degli Studi di Milano.

References

1. Ashburner, M., et al.: Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. Nature genetics **25**(1) (May 2000) 25–29
2. Eisner, R., Poulin, B., Szafron, D., Lu, P.: Improving protein prediction using the hierarchical structure of the Gene Oontology. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology. (2005)

3. Mostafavi, S., Morris, Q.: Using the gene ontology hierarchy when predicting gene function. In: Proceedings of the Twenty-Fifth Annual Conference on Uncertainty in Artificial Intelligence (UAI-09), Corvallis, Oregon, AUAI Press (2009) 419–427
4. Youngs, N., Penfold-Brown, D., Bonneau, R., Shasha, D.: Negative example selection for protein function prediction: The NoGO database. PLOS Computational Biology **10**(6) (06 2014) 1–12
5. Youngs, N., Penfold-Brown, D., Drew, K., Shasha, D., Bonneau, R.: Parametric bayesian priors and better choice of negative examples improve protein function prediction. Bioinformatics **29**(9) (2013) 1190–1198
6. Frasca, M., Lipreri, F., Malchiodi, D.: Analysis of informative features for negative selection in protein function prediction. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **10209** (2017) 267–276
7. Boldi, P., Frasca, M., Malchiodi, D.: Evaluating the impact of topological protein features on the negative examples selection. BMC Bioinformatics **19**(14) (Nov 2018) 417
8. Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. Machine Learning **28**(2) (Aug 1997) 133–168
9. Bertoni, A., Frasca, M., Valentini, G.: COSNet: a cost sensitive neural network for semi-supervised learning in graphs. In: ECML PKDD 2011. Volume 6911 of Lecture Notes on Artificial Intelligence., Springer (2011) 219–234 doi:10.1007/978-3-642-23780-5_24.
10. Szklarczyk, D., et al.: String v10: proteinprotein interaction networks, integrated over the tree of life. Nucleic Acids Research **43**(D1) (2015) D447–D452
11. Dagan, I., Engelson, S.P.: Committee-based sampling for training probabilistic classifiers. In: In Proceedings of the Twelfth International Conference on Machine Learning, Morgan Kaufmann (1995) 150–157
12. Melville, P., Mooney, R.J.: Diverse ensembles for active learning. In: Proceedings of the Twenty-first International Conference on Machine Learning. ICML '04, New York, NY, USA, ACM (2004) 74–
13. Abe, N., Mamitsuka, H.: Query learning strategies using boosting and bagging. In: Proceedings of the Fifteenth International Conference on Machine Learning. ICML '98, San Francisco, CA, USA (1998) 1–9
14. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc., New York, NY, USA (1995)
15. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods. Cambridge University Press, New York, NY, USA (2000)
16. Breiman, L., Friedman, G., Olshen, R., Stone, C.: Classification And Regression Trees. Wadsworth, Belmont, CA (1984)
17. Gini, C.: Variabilità e Mutuabilità. Contributo allo Studio delle Distribuzioni e delle Relazioni Statistiche, C. Cuppini, Bologna (1912)
18. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics **1** (1945) 80–83
19. Lovász, L.: Random walks on graphs: A survey. In Miklós, D., Sós, V.T., Szőnyi, T., eds.: Combinatorics, Paul Erdős is Eighty. Volume 2. János Bolyai Mathematical Society, Budapest (1996) 353–398
20. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the Twentieth International Conference on International Conference on Machine Learning. ICML'03, AAAI Press (2003) 912–919

21. Schwikowski, B., Uetz, P., Fields, S.: A network of protein-protein interactions in yeast. Nature biotechnology **18**(12) (December 2000) 1257–1261
22. Frasca, M., Pavesi, G.: A neural network based algorithm for gene expression prediction from chromatin structure. In: IJCNN, IEEE (2013) 1–8 doi:10.1109/IJCNN.2013.6706954.
23. Frasca, M., Bertoni, A., Valentini, G.: UNIPred: Unbalance-aware Network Integration and Prediction of Protein Functions. Journal of Computational Biology **22**(12) (2015) 1057–1074
24. Lan, L., Djuric, N., Guo, Y., S., V.: MS-kNN: protein function prediction by integrating multiple data sources. BMC Bioinformatics **14**(Suppl 3:S8) (2013) S3–S8
25. Jiang, Y., Oron, T.R., et al.: An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome Biology **17**(1) (2016) 184
26. Re, M., Mesiti, M., Valentini, G.: A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks. IEEE ACM Transactions on Computational Biology and Bioinformatics **9**(6) (2012) 1812–1818