Predicting the functions of proteins in Protein-Protein Interaction networks from global information

Hossein Rahmani
Leiden Institute of Advanced Computer Science, Universiteit Leiden, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
hrahmani@liacs.nl

Hendrik Blockeel
Leiden Institute of Advanced Computer Science, Universiteit Leiden, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
and
Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium
blockeel@liacs.nl

Andreas Bender
Division of Medicinal Chemistry, Leiden/Amsterdam Center for Drug Research, Universiteit Leiden, 2333 CC Leiden, The Netherlands
andreas.bender@pharma-it.net

Editor: Sašo Džeroski, Pierre Geurts, and Juho Rousu

Abstract

In this work we present a novel approach to predict the function of proteins in protein-protein interaction (PPI) networks. We classify existing approaches into inductive and transductive approaches, and into local and global approaches. As of yet, among the group of inductive approaches, only local ones have been proposed for protein function prediction. We here introduce a protein description formalism that also includes global information, namely information that locates a protein relative to specific important proteins in the network. We analyze the effect on function prediction accuracy of selecting a different number of important proteins. With around 70 important proteins, even in large graphs, our method makes good and stable predictions. Furthermore, we investigate whether our method also classifies proteins accurately on more detailed function levels. We examined up to five different function levels. The method is benchmarked on four datasets where we found classification performance according to F-measure values indeed improves by 9 percent over the benchmark methods employed.

Keywords: Protein Function Prediction, PPI Networks, Important Protein Selection

1. Introduction

In recent years, much effort has been invested in the construction of protein-protein interaction (PPI) networks (Stelzl et al., 2005). Much can be learned from the analysis of such networks with respect to the metabolic and signalling processes present in an organism, and the knowledge gained can also be prospectively employed e.g. to predict which proteins are suitable drug targets, according to an analysis of the resulting network (Ma’ayan et al., 2007). One particular machine learning task that has been considered is predicting the functions of proteins in the network.

A variety of methods have been proposed for predicting the classes of proteins. On a high level we can distinguish two types of approaches, namely inductive and transductive ones. Inductive learning approaches, also called model-based approaches, construct a model (a mathematical function) that maps a description of a protein onto its functions. Transductive approaches, on the other hand, immediately make predictions for the proteins in the network, without going through the intermediate
stage of constructing a model that can be used afterwards for making predictions. The difference between these two will be described more formally in the next section.

Transductive approaches are often “global”: information on the whole network is taken into account when making predictions. The inductive approaches that have been used until now are typically local, in the sense that the description of a protein (from which its labels are to be predicted) contains information about the local neighborhood of the protein, not about the network as a whole. This is not an inherent property of inductive approaches, though; one might just as well try to construct a description that contains global information. Accordingly, in this paper we explore the usefulness of one particular kind of global information for the task of protein function prediction, namely the relative position of a protein with respect to specific other proteins.

This paper is structured as follows. In Section 2 we define the learning problem formally. In Section 3 we briefly review approaches that have been proposed before to solve this problem. In Section 4 we present a new inductive learning approach; we do not present any new learning algorithms but a new description format of proteins, which contains global rather than local information. In Section 5 we empirically evaluate the performance of several learning algorithms when using this format, and, as a control experiment, compare this performance to that of a previously proposed approach. We present our conclusions in Section 6.

2. Problem Statement

Mathematically, PPI networks can be represented as graphs, and the problem we consider is that of predicting the labels of nodes in this graph.

Consider an undirected graph $G$ with node set $V$ and edge set $E$, where each node $v \in V$ is annotated with a description $d(v) \in D$ and, optionally, a label $l(v) \in L$. We assume that there exists a “true” labelling function $\lambda$ from which $l$ is a sample, that is, $l(v) = \lambda(v)$ where $l(v)$ is defined.

In transductive learning, the task is to predict the label of all the nodes. That is, given the graph $G = (V, E, d, l)$, with $l$ a partial function, the task is to construct a completed version $G' = (V, E, d, l')$ with $l'$ a complete function that is consistent with $l$ where $l(v)$ is defined.

In practice, there is an additional constraint that $l'$ should approximate $\lambda$. This is imposed by some optimization criterion $o$, the exact form of which expresses assumptions about $\lambda$. For instance, $o$ could express that nodes that are directly connected to each other tend to have similar labels by stating that the number of $\{v_1, v_2\}$ edges where $l'(v_1) \neq l'(v_2)$ should be minimal. The assumptions made about $\lambda$ are called the bias of the transductive learner.

In inductive learning, the task is to learn a function $f : D \rightarrow L$ that maps a node description $d(v)$ onto its label $l(v)$. That is, given $G = (V, E, d, l)$, we need to construct $f : D \rightarrow L$ such that $f(d(v)) = l(v)$ when $l(v)$ is defined, and $f$ is defined for all elements of $D$. Note that $f$ differs from $l$ in that it maps $D$, not $V$, onto $L$. This implies, for instance, that it can also make predictions for a node $v$ that was not in the original network, as long as $d(v)$ is known.

Besides the bias expressed by the optimization criterion $o$ (which may still be present), there is now also a bias imposed by the choice of $D$: whenever two different nodes have the same description, they are assumed to have the same labels: $d(v_1) = d(v_2) \Rightarrow \lambda(v_1) = \lambda(v_2)$. Additionally, the learning algorithm used to learn $f$ has its own inductive bias (Mitchell, 1997): given exactly the same inputs, two different learning algorithms may learn different functions $f$, according to assumptions they make about the likely shape of $f$.

Thus we have three types of bias. Transductive learners have a transductive bias, which is implied by the choice of the optimization criterion $o$. Inductive learners have a description bias, imposed by the choice of $d$, as well as an inductive bias, imposed by the choice of the learning algorithm that is used to learn $f$ from $(d(v), l(v))$ pairs. In this paper we will explore for one particular description function $d$ whether it represents a suitable description bias.

In the context of protein function prediction in PPI networks, the nodes $v$ are proteins; the descriptions $d(v)$ can be any description of the protein that can be derived from the network structure.
(where no additional information, such as the protein structure, is assumed to be available; we assume we learn from the network structure only); the labels \( \ell(v) \) are sets of protein functions.

Note that many proteins have more than one function (Rassow et al., 1995); this is why a node label can be any set of functions. Most off-the-shelf machine learning techniques can only learn classifiers that predict a single value, not a set of values. The fact that node labels are sets may seem to form a problem in this respect. To remedy this situation, if we have \( n \) possible functions, the task of predicting a subset of these functions can easily be transformed into \( n \) single-function prediction tasks: for each possible function a binary classification task is then constructed where nodes are to be assigned the class true or false depending on whether the protein has that function or not. This is the setting we will focus on in this paper.

3. Related work

Among transductive approaches to the protein function prediction problem, the Majority Rule approach has a prominent role (Schwikowski et al., 2000). This method assigns to a protein those functions that occur most frequently among its neighbors (typically a fixed number of functions is predicted, for instance, the three most frequently occurring functions in the neighborhood). One problem with this approach is that it only considers neighbors of which the function is already known, ignoring all others. To address this problem, global optimization-based function prediction methods have been proposed. Any probable function assignment to the whole set of unclassified proteins is given a score, counting the number of interacting pairs of nodes with no common function; the function assignment with the lowest value will be the best assignment (Vazquez et al., 2003; Sun et al., 2006).

Another improvement over the original implementation was made by observing higher-level interactions (Chua et al., 2006). Level \( k \) interaction between two proteins means that there is a path of length \( k \) between them in the network. Proteins that have both a direct interaction and shared level-2 interaction partners have turned out to be more similar to each other (i.e. having same functions). Taking this further, one can make the assumption that in dense regions (subgraphs with many edges, relative to the number of nodes) most nodes have similar functions. This has led to clustering approaches which first cluster the networks (with clusters corresponding to dense regions), and subsequently predict the function of unclassified proteins based on the cluster they belong to (King et al., 2004; Brun et al., 2004).

Among the inductive approaches, Milenkovic and Przulj’s (2008) graphlet-based approach has been used in the area of protein function predictions. The node description \( d(v) \) that is built here, in their terminology the “signature vector”, describes the local neighborhood of the node in terms of so-called graphlets, small graph structures as a part of which each node occurs. Most other inductive approaches use similar signatures. Typical for them is that they describe only the local structure of the network near the node to be predicted, however remote changes in the network do not influence the signature at all.

4. A global description of proteins

In this work we will now introduce an inductive approach that uses global node descriptions to the area of protein-protein interactions; that is, any change (e.g., addition or removal of an edge) in the network, wherever it occurs, may influence a node’s description. Our hypothesis is that the inclusion of additional information will improve the function prediction of unknown nodes which will be investigated in the following in detail.

We describe a node as follows. Assume that there are \( n \) nodes in the network, identified through numbers 1 to \( n \). Each node is then described by an \( n \)-dimensional vector. The \( i \)th component in the vector of a node \( v \) gives the length of the shortest path in the graph between \( v \) and node \( i \).
It has been hypothesized before that shortest-path distances are relevant in PPI network analysis; for instance, Rives and Galitski (2003) cluster nodes based on shortest-path distance profiles. As of yet, however, such shortest-path distances have not been considered in the context of inductive learning of protein function predictors which is the rationale behind the current work.

A potential disadvantage of this method is that in large graphs, one gets very high-dimensional descriptions, and not all learners handle learning from high-dimensional spaces well. It is possible, however, to reduce the dimensionality of the vector by only retaining the shortest-path distance to a few “important” nodes. This essentially represents a feature selection problem. A node $i$ is important if the shortest-path distance of some node $v$ to $i$ is likely to be relevant for $v$’s classification. If the feature $f_i$ denotes the shortest path distance to node $i$, one possible measure of the relevance of $f_i$ for the label of a node (which is a set of functions) is the following.

For each function $j$, let $G_j$ be the set of all proteins that have that function $j$. Let $\text{Mean}_{k \in G_j}(f_{ik})$ be the average $f_i$ value take over all proteins $k$ in $G_j$, and $\text{Var}_{k \in G_j}(f_{ik})$ the variance of the $f_i$ value take over all proteins $k$ in $G_j$. The following formula, inspired by ANOVA (analysis of variance), gives an indication of how relevant $f_i$ is for the function set as a whole:

$$\forall p_i \in P; A_i = \frac{\text{Var}_j[\text{Mean}_{k \in G_j}(f_{ik})]}{\text{Mean}_j[\text{Var}_{k \in G_j}(f_{ik})]}$$  \hspace{1cm} (1)

where $P$ is the set of all proteins in the network and $F$ contains all possible functions. $\text{Var}_j$ and $\text{Mean}_j$ denote the Variance and Mean operators taken over all values of $j$. A high $A_i$ denotes a high relevance of feature $f_i$. Figure 1 shows the intuitive representation of formula 1. Imagine there are three functions $F_1$, $F_2$ and $F_3$ in the network. First, we put the proteins having the same function in the same group and forming the $G_1$, $G_2$ and $G_3$ groups. Second, in order to calculate the ANOVA value of each protein $P_i$ in the network, we find the shortest path distance of the protein $P_i$ to all the members of each group (i.e., $f_{ij}$). Finally, we calculate the average and variance of different $f_{ij}$ in each function group $G_j$.

![Figure 1: Intuitive representation of formula 1.](image)

To illustrate this measure, figure 2 shows two different scenarios. In the first scenario, all three averages and all three variances are equal. If the X axis shows the value of shortest path distance to protein $P_i$, then we can not predict one specific function based on the shortest path distance to the protein $P_i$. So, protein $P_i$ does not discriminate different functions in this scenario and is not an “important” protein. In the second scenario, the values of variances are equal but average values are different. In this scenario, if the shortest path distance of one protein to protein $P_i$ is smaller than $\mu_1$ or bigger than $\mu_3$ then we predict function $F_1$ or $F_3$ for that protein respectively. If the shortest
path distance is between $\mu_1$ and $\mu_3$ then we predict the function $F_2$ for that protein. In the second scenario, protein $P_i$ discriminates different functions so we could use it as an important protein.

Figure 2: Discriminative protein versus non-discriminative protein.

In the following, we will empirically determine whether the shortest-path distances to all, or a few particular, nodes are indeed informative with respect to a protein’s functions by evaluating the performance of the method on a benchmark dataset.

5. Experiments
We performed four consecutive experiments.

1. We evaluated the potential of the proposed protein description for protein function prediction by assessing multiple learning systems and finding the learning system whose inductive bias best fits our dataset. This step was made to alleviate the risk of concluding that the description is unsuitable, when the cause for bad results is in fact a poor choice of learner.

2. We compared the performance of this system with that of Majority Rule (Schwikowski et al., 2000), a transductive learner.\(^1\)

3. We investigated the effect of the ANOVA-based node selection criterion on predictive performance: Does a reduction of the number of important nodes increase or decrease the predictive performance, and is there a clear optimum with respect to the number of important nodes that should be selected?

4. While these experiments focused on prediction of functions on the highest level of the functional hierarchy, we check whether our method also yields good predictive accuracy at lower levels.

We evaluate predictive performance using the following measures: area under the ROC curve (AUC) (Provost and Fawcett, 1997), precision, recall, and F-measure. We do not include predictive accuracy (percentage of predictions that are correct) because for several function prediction tasks, the class distribution is highly skewed (e.g., 1% of the protein has that function, 99% does not), and in such cases predictive accuracy (the percentage of predictions that is correct) does not carry much information. AUC and precision/recall are much more robust to skewed class distributions.

5.1 Datasets
We apply our method to four *S. cerevisiae* PPI networks: DIP-Core (Deane et al., 2002), VonMering (von Mering et al., 2002), Krogan (Krogan et al., 2006) and MIPS (Mewes et al., 2000), which contain 4400, 22000, 14246 and 44514 interactions among 2388, 1401, 2708 and 7928 proteins respectively.

\(^1\) Majority Rule was selected for its ease of implementation, and because it is still a regularly used reference method.
We consider 17 high level functions for evaluating our function predictors. Figure 3 shows high level MIPS function categories with their corresponding function number.

<table>
<thead>
<tr>
<th>Functional Category</th>
<th>Function Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>METABOLISM</td>
<td>01</td>
</tr>
<tr>
<td>ENERGY</td>
<td>02</td>
</tr>
<tr>
<td>CELL CYCLE AND DNA PROCESSING</td>
<td>10</td>
</tr>
<tr>
<td>TRANSCRIPTION</td>
<td>11</td>
</tr>
<tr>
<td>PROTEIN SYNTHESIS</td>
<td>12</td>
</tr>
<tr>
<td>PROTEIN FATE (folding, modification, destination)</td>
<td>14</td>
</tr>
<tr>
<td>PROTEIN WITH BINDING FUNCTION OR COFACTOR REQUIREMENT (structural or catalytic)</td>
<td>16</td>
</tr>
<tr>
<td>REGULATION OF METABOLISM AND PROTEIN FUNCTION</td>
<td>18</td>
</tr>
<tr>
<td>CELLULAR TRANSPORT, TRANSPORT FACILITIES AND TRANSPORT ROUTES</td>
<td>20</td>
</tr>
<tr>
<td>CELLULAR COMMUNICATION/SIGNAL TRANSDUCTION MECHANISM</td>
<td>30</td>
</tr>
<tr>
<td>CELL RESCUE, DEFENSE AND VIRULENCE</td>
<td>32</td>
</tr>
<tr>
<td>INTERACTION WITH THE ENVIRONMENT</td>
<td>34</td>
</tr>
<tr>
<td>TRANSPOSABLE ELEMENTS, VIRAL AND PLASMID PROTEINS</td>
<td>38</td>
</tr>
<tr>
<td>CELL FATE</td>
<td>40</td>
</tr>
<tr>
<td>DEVELOPMENT (Systemic)</td>
<td>41</td>
</tr>
<tr>
<td>BIOGENESIS OF CELLULAR COMPONENTS</td>
<td>42</td>
</tr>
<tr>
<td>CELL TYPE DIFFERENTIATION</td>
<td>43</td>
</tr>
</tbody>
</table>

Figure 3: MIPS high level function categories.

5.2 Comparison of Learners

Given the input data and a particular function to predict, any standard machine learning tool can be used to build a model that predicts from a node’s description, whether the node has a particular function or not. We have experimented with several methods, as available in the Weka data mining toolbox (Witten and Frank, 1999), namely decision trees (J48), random forests, an instance based learner (IBk), Naive Bayes, radial basis function networks, Support Vector Machine (libSVM), Classification via Regression (CVR) and Voting Feature Intervals (VFI). We examined three kernel functions namely polynomial, radial basis and sigmoid kernels in the libSVM method, and select the kernel which gives the highest AUC value among the three types of kernel function. These methods were chosen to be representative for a broad range of machine learning methods. This comparative evaluation was made on the DIP-Core data set. The results are shown in Figure 4. Looking at average AUC over the functions to be predicted, we see that Random Forests score best (AUC = 0.7), with IBk a close second (0.67). These averages may seem close, but when we look at individual labels, we see that there is only one win and one draw for IBk, and 15 losses, compared to RF. This shows that the difference, while small, is significant.

It is interesting to see that RF performs best among all learners in 13 out of 17 cases, and the 4 cases where it does not are all characterized by a high class skew. (Figure 5 visualises this.) This is, in hindsight, not surprising: Random Forests are ensembles of decision trees, and these are known to perform less well on highly skewed class distributions. In our case, however, while most datasets have a strong class skew, for the large majority of them this is not problematic.

We have concluded from the above results that Random Forests are our best candidate for learning from the given type of data, and we have used this method in the remaining experiments.
5.3 Comparison with a transductive method

We next compared Random Forests and Majority Rule in predicting the proteins’ functions in four datasets DIP-Core, VonMering, Krogan and MIPS. Firstly, we selected 700 nodes based on the ANOVA Measure. Then, we found the shortest path of each protein to those selected proteins. We used this information as the input for Weka and calculated the average Precision, Recall, F-measure and AUC for each function class in a 10-fold cross validation. Figure 6 compares the average precision, over all classes, of Majority Rule (MR) and Random Forests (RF). Figure 7 similarly compares the recall of MR and RF, and Figure 8 the F-measures. We see that, over the four datasets, RF has higher precision (11% higher in average) but smaller Recall (10% smaller in average). RF and MR perform almost similarly with respect to F-measure. The AUCs are compared in Figure 9: again, RF tends to have higher scores (+6%).

5.4 Different Number of Important Proteins

Furthermore, we investigated the effect of selecting a different number of important proteins on the classification metrics. We selected the 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200 and 300 most important proteins according to Equation 1. For each number \( n \), we created a dataset where each protein in the network is described using its distance to the \( n \) most important proteins. We trained a random forest from that dataset and recorded its precision, recall, F-measure and AUC; these numbers are finally plotted against \( n \). For each combination of function and dataset this gives a separate curve.
The shape of the curves is qualitatively very similar in all cases. Figures 10–13 show a few representative cases. In general, we observe the following behavior:

- When the number of important proteins is limited to less than 10, this typically yields bad predictive performance. Even though in a few cases the curves already start increasing in this area, they do not reach their maximum.

- In the area of 10-50 important proteins, there is usually a major improvement in all four metrics.

- For most of the functions, selecting 50-70 important proteins is enough to obtain good classification results. Beyond this area, there is usually no major improvement in performance. Nevertheless, in a small number of cases, as visible in Figures 10 and 13, the performance keeps increasing significantly when the number of important proteins is raised to 300.

We did not systematically increase the number of important proteins beyond 300 because of computational complexity reasons. (Weka’s random forest learning method is relatively slow for datasets of this size; the fact that it needs to be run for each separate function and dataset, and that each time a ten fold cross-validation is performed, makes it necessary to limit the number of datasets on which it is run.) Nevertheless, our results show that in the large majority of cases a reduction of the number of important proteins to a relatively small number (50-70) is possible without predictive performance suffering too much from this. To really maximize predictive performance, however, experimenting with a larger number of important proteins may be useful.
Looking at the Figures 10–13, we further notice that the effect of the number of important proteins is much more pronounced for the F-measure than for the AUC. In fact, in several cases there is no clearly perceivable trend in the AUC metric: even though in general AUC tends to go up with increasing F-measure, in some cases it remains relatively constant, and random variations are relatively large compared to the systematic variation. Figure 14 illustrates this clearly. Thus, from the point of view of comparing the quality of different models, the F-measure seems a more dependable metric than AUC.

5.4.1 Comparison with Random Selection

The above experiments show that it is possible to reduce the number of important nodes significantly without predictive performance suffering too much, but they do not answer the question whether this is because our ANOVA-based selection method performs well, or because any small number of “important” nodes would simply give us enough information, no matter how we define “important”. To answer this question, we compared our ANOVA-based selection method to random selection (i.e. simply choosing \( n \) proteins at random and describing other proteins by their distances to these proteins).

We compared the F-measures obtained when using the ANOVA-based and random selection criteria for eleven different numbers of important proteins: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,
Figure 10: Function F-10 in VonMering dataset. Selecting less than 10 important proteins is not enough for discriminating proteins with different function.

Figure 11: Function F-10 in DIP-Core dataset. Major improvement happens when the number of important proteins is between 10 to 50 proteins.

Since random selection yields different results depending on the random choices made, and this influenced the F-measure quite a bit, for each number of important proteins we ran the random selection based method 20 times and reported average F-measure.

Figure 15 compares the ANOVA based selection method with random selection, evaluated according to the F-measure metric, in the DIP-Core dataset. If the number of important proteins
Figure 12: Function F-14 in DIP-Core dataset. Uniform metrics’ values after selecting 50 important proteins.

Figure 13: Function F-20 in DIP-Core dataset. This is an example of a case where classification accuracy may keep improving when increasing the number of important proteins beyond 300.

is less than 20 or over 100, then there is no big difference between these two methods, however between 20 and 100 selected proteins the ANOVA-based selection method clearly improves upon random selection. This result suggests that when the protein description is very detailed (distances to many other proteins are given), it does not matter what these other proteins are, and when it is very coarse (distances to very few other proteins are given), there is not enough information in these distances, regardless of whether the important proteins are selected at random or according
Figure 14: Function F-16 in the Krogan dataset. The AUC is more or less constant when varying the number of important proteins, whereas the F-measure shows a clear increase.

to the ANOVA criterion. However, in between these extremes, our ANOVA criterion clearly selects proteins with a higher information content than randomly selected proteins.

Figure 15: Average F-measure obtained with Random Forests using the ANOVA-based feature selection, versus using random selection, in the DIP-Core dataset.

5.5 Function Levels

Proteins’ functions have hierarchical structures. As we discussed in the section 5.1, we only consider high level MIPS functions. For example, two functions 11.02.01 (rRNA synthesis) and 11.02.03 (mRNA synthesis) are considered similar up to the second function level (i.e. 11.02 =RNA synthesis), but not on deeper levels. In this section, we investigate whether our method also classifies proteins accurately on more detailed levels. We examine up to five different function levels and for each level compare our method with Majority Rule.
Figure 16 compares the F-measure obtained by Majority Rule and our method on the DIP-Core and VonMering datasets, for five different function levels 1–5, where 1 is the level we used in our earlier experiments. Figure 17 visualizes the difference in F-measures between both approaches. The highest improvement is observed for function level 2, when our method has more than 8% higher F-measure value, on average, for the DIP-Core and VonMering datasets. The difference is smallest for very general (level 1) or very specific (level 5) function prediction.

6. Conclusions

To summarize, we have firstly classified existing methods for the prediction of node properties in a network into transductive and inductive methods. This distinction provides insight into potential strengths and weaknesses of the methods, particularly in terms of learning bias. Inductive learning methods make different assumptions about the true labeling function than transductive methods, which guided our choice of algorithm employed in this work. Secondly, we observed that existing inductive learning methods for predicting protein functions in PPI networks use local information, while the use of global information for such methods has as of yet remained unexplored. Accordingly, we have, thirdly, introduced a node description formalism that has not been used previously for protein function prediction and which takes global information into account. Together with this node description formalism we have introduced and evaluated a method for reducing the number of features needed for the description. We analyzed the effect of selecting a different number of important proteins on the classification metrics. We found that, for most of the functions, selecting
50–70 important proteins is enough to obtain good classification results. Beyond this area, there is usually no major improvement in performance. Furthermore, we investigate whether our method also classifies proteins accurately on more detailed levels. We examine up to five different function levels. On four benchmark datasets, DIP-Core, VonMering, Krogan and MIPS, we have shown that a standard learner using this formalism outperforms the benchmark Majority Rule approach according to Precision, F-measure and AUC and, hence, that our description formalism is informative with respect to the prediction of a protein’s functions from its location in the PPI network.

In the future, a more extensive comparison with other learners would be warranted. It would also be interesting to determine to what extent the information in our global protein description is complementary to that used in other (local inductive, or transductive) approaches. The reason is that when several predictors exploit different information when making their predictions, they can typically be combined into a single composite predictor that performs better than each individual one. Finally, while we have focused here on models that predict a single class at a time, there exist a few methods that predict multiple classes simultaneously (Blockeel et al., 2006). Hence, it would be useful to investigate to what extent these classifiers yield better predictions than the single-label prediction approach presented here.

Acknowledgments
This research is funded by the Dutch Science Foundation (NWO) through a VIDI grant.

References


