Supplementary material

Gene Ontology-driven inference of protein-protein interactions using inducers

S. R. Maetschke, M. Simonsen, M. J. Davis, M. A. Ragan

October 12, 2011

1 Integration of ontologies

GO is divided into the three sub-ontologies: biological process (BP), molecular function (MF), cellular component (CC), represented as disjunct graphs[1]. We are interested in the prediction accuracies of inducers for individual sub-ontologies and the accuracy when combining the information of the three sub-ontologies. Natively, inducers operate over individual sub-ontologies. To integrate information we implemented a simple approach and concatenated the feature vectors generated by the inducers for each of the sub-ontologies and using the combined vector as input to a classifier. Figure 1 depicts the architecture of the system.

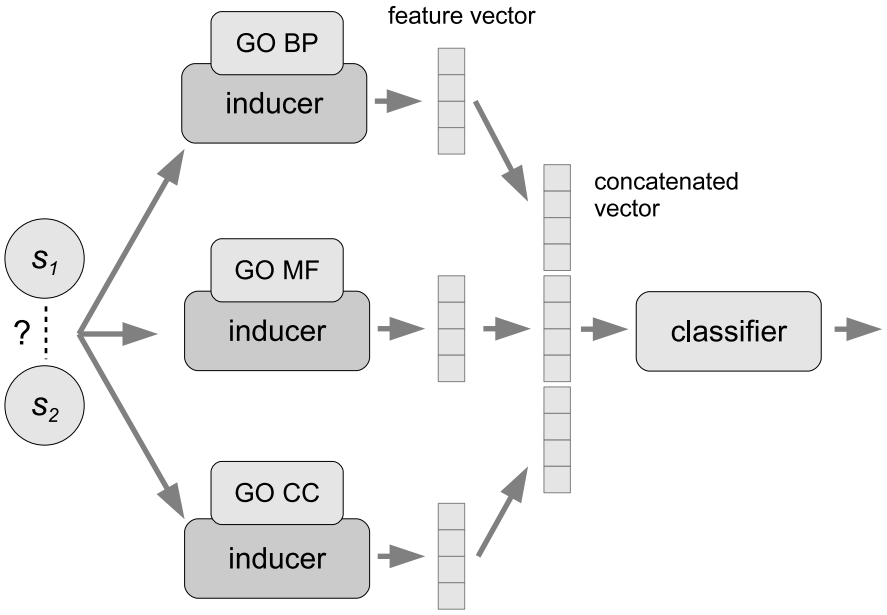

Figure 1: Architecture of the system to integrate the three sub-ontologies (BP, MF, CC). For each sub-ontology an inducer creates a feature vector. The feature vectors are concatenated and serve as input to a classifier. S_1 and S_2 are the GO term sets, which annotate the two proteins a prediction is performed for.

[1]Recent versions of GO contain certain cross-links between ontologies that we ignore here.

We evaluated the prediction accuracy of the *ULCA* inducer with a RF classifier on Park's dataset (YP) individually for each sub-ontology, when combining two ontologies, and when combining all three ontologies (BP+CC+MF), using the prediction system depicted in Figure 1. Table 1 shows the results of this comparison. 4-fold cross-validation was performed (in accordance to Park (2009)) and GO annotations with *IGI* or *IPI* evidence codes were excluded to avoid any bias (see Rogers and Ben-Hur (2009)).

AUC	std	F_1	Precision	Recall	Ontologies
0.74	0.01	0.68	0.67	0.71	MF
0.84	0.02	0.76	0.78	0.74	CC
0.86	0.01	0.78	0.80	0.76	BP
0.86	0.02	0.79	0.82	0.76	CC+MF
0.87	0.01	0.79	0.81	0.77	BP+MF
0.89	0.01	0.81	0.85	0.78	BP+CC
0.90	0.01	0.82	0.86	0.78	BP+CC+MF

Table 1: Prediction performance of the *ULCA* inducer with RF classifier on Park's dataset (YP) for different (combinations of) sub-ontologies.

The accuracies for the individual ontologies are lower than the combined approaches, with the MF ontology showing the lowest AUC. Integrating the BP and the CC ontologies leads to a substantial improvement over the individual ontologies, while integration with the MF ontology improves the prediction accuracies only marginally.

2 Comparison with other predictors

In this section we compare the prediction accuracy of the inducer approach with other prediction methods, such as sequenced based PPI predictors evaluated by Park (2009) and a SVM with multiple kernels by Ben-Hur and Noble (2005). For all comparisons we used the *ULCA* inducer with a RF classifier and the combined ontologies (see Section 1).

2.1 Comparison with Park

Park (2009) performed a comparative evaluation of five sequence-based methods ($M1$-$M4$, C) on a yeast and a human protein-protein interaction dataset. Method $M1$ by Martin *et al.* (2005) utilizes trigram signatures as features. $M2$ by Pitre *et al.* (2006) exploits the co-occurrence of sub-sequences. $M3$ by Shen *et al.* (2007) evaluates trigrams over a reduced amino-acid alphabet and $M4$ by Guo *et al.* (2008) employs auto-correlation over seven physicochemical properties to predict interactions. C by Park (2009) is a consensus over the methods $M1$-$M4$. All methods utilize SVMs as machine learning classifier.

Method	Yeast	Human
M1	0.83	0.86
M2	0.79	0.81
M3	0.60	0.67
M4	0.75	0.83
C	0.85	0.91

Table 2: Prediction performance (AUC) of five sequence-based methods on yeast and human PPI datasets. Data taken from Park (2009).

Table 2 shows the prediction performance (AUC) of those five methods (values taken from Park (2009)). Results are 4-fold cross-validated. Training was performed with a balanced dataset but tests were performed with 10 times more negatives (non-interactions) than positives (interactions). The PPI datasets created by Park use protein identifiers different from the Uniprot identifiers our GO-driven predictor requires and a mapping of identifiers is necessary. In case of Park's human interaction dataset we could map only 84% of the identifiers and therefore did not use this dataset for any comparison. However, almost all of the DIP identifiers in Park's yeast dataset with 2159 proteins and 3867 interactions could be mapped, resulting in a very similar dataset (*YP*) with 2155 proteins and 3844 interactions. We therefore performed all our comparisons with sequence-based methods on the *YP* dataset. Figure 2 compares the prediction accuracy of the five sequenced-based methods with that of the *ULCA* inducer in combination with a RF classifer and integrated ontologies.

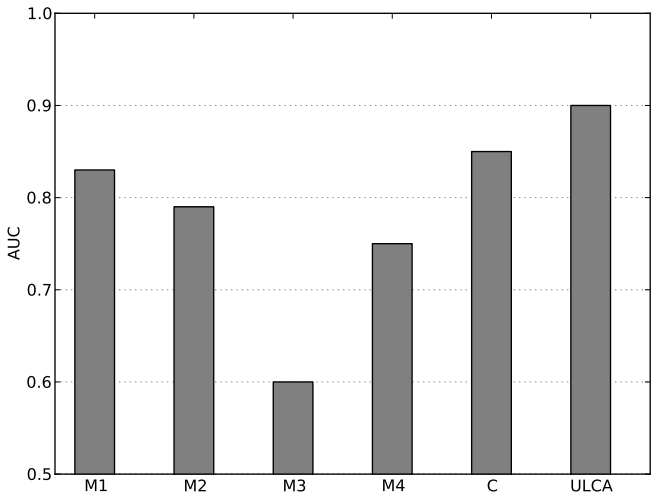

Figure 2: Comparison of prediction accuracy (AUC) of the *ULCA* inducer using a RF classifier on Park's *YP* dataset with five sequence-based methods (M1-M4, C). Park did not report standard deviations and therefore no error bars are shown.

The *ULCA* inducer approach with an AUC of 0.90 outperforms the five sequenced-based methods, including the consensus method (*C*) for which an AUC of 0.85 was reported.

2.2 Comparison with Ben-Hur and Noble

Ben-Hur and Noble (2005) employed a SVM and a combination of multiple kernels, derived from sequence data, GO annotation, network properties and interologs, to predict protein-protein interactions in yeast. Interaction data were extracted from the BIND database and two datasets were created: a large dataset (*AB*) with 10517 interactions between 4233 proteins, and a smaller set (*AB-rel*), filtered for reliable interactions, with 750 interactions and 736 proteins. Equal numbers of negative samples were generated by randomly sampling non-interacting protein pairs.

We downloaded the datasets provided by Ben-Hur and Noble (2005), mapped the protein identifiers to Uniprot accession numbers, and measured the prediction accuracy (AUC, AUC_{50}) of the *ULCA* inducer with a RF classifier on those two datasets. Table 3 compares the accuracies of the inducer approach (ULCA) with the accuracies reported by Ben-Hur and Noble (2005) for the GO-kernel (GOK) alone and a combination of all kernels (AK1). The GO-kernel measured the similarity between two proteins similar to Resnik's SSM by computing the maximum log likelihood over the common ancestors of the GO term annotations.

AUC	AUC_{50}	Method	Dataset
0.68	-	*GOK*	*AB*
0.85	0.63	*ULCA*	*AB*
0.95	-	*GOK*	*AB-rel*
0.98	0.58	*AK*1	*AB-rel*
0.93	0.58	*ULCA*	*AB-rel*

Table 3: Prediction performance of kernel-based methods (AK,GOK) by Ben-Hur and Noble (2005) and the *ULCA* inducer with a RF classifier on two datasets (*AB*, *AB-rel*). Results are 5-fold cross-validated.

Since the multi-kernel (*AK*1) method exploits additional sources of information beyond GO annotation it is expected to outperform the inducer approach, which utilizes GO annotation only. This expectation is confirmed by the results that show an AUC of 0.98 for the *AK*1 method on the reliable dataset (*AB-rel*), while the inducer approach achieves only an AUC of 0.93. No AUCs for the larger *AB* dataset were reported, and a comparison to the AK method on this dataset is therefore not possible. However, prediction accuracies for the GO-kernel (*GOK*) alone on both dataset are available. On the reliable dataset (*AB-rel*) the inducer approach with an AUC of 0.93 is slightly inferior to the GO-kernel with an AUC 0.95. On the larger dataset (*AB*) the *ULCA* inducer performs substantially better (AUC of 0.85 compared to 0.68).

3 Comparison of inducers

In this section we compare the prediction accuracies of inducers on various datasets. Table 4 lists the involved PPI datasets with their numbers of proteins and interactions. datasets were extracted from Version 9.0 of the STRING database (Jensen *et al.*, 2009) and filtered for experimentally validated interactions with a confidence score ≥ 0.9. Also, redundant interactions, self-interactions and interaction without a *binding action* tag were removed.

Label	Species	#proteins	#interactions
SC	*Saccharomyces cerevisiae*	3291	15238
HS	*Homo sapiens*	3296	3490
EC	*Escherichia coli*	589	1167
SP	*Schizosaccharomyces pombe*	904	742
AT	*Arabidopsis thaliana*	756	541
MM	*Mus musculus*	1088	500
DM	*Drosophila melanogaster*	658	321

Table 4: Protein-protein interaction datasets extracted from the STRING database with their numbers of proteins and interactions.

Table 5 contains the prediction accuracies (AUCs) for all inducers using a NB classifier on all dataset listed in Table 4. Furthermore, mean values and standard deviations over inducers and datasets are reported.

Inducer	*MM*	*DM*	*AT*	*HS*	*SC*	*SP*	*EC*	mean	std
AC	0.55	0.66	0.61	0.70	0.68	0.69	0.79	0.67	0.07
ACA	0.54	0.67	0.66	0.73	0.70	0.80	0.89	0.71	0.10
OLCA	0.54	0.67	0.67	0.75	0.74	0.80	0.90	0.72	0.11
AA	0.63	0.72	0.73	0.70	0.72	0.80	0.89	0.74	0.08
AL	0.61	0.72	0.73	0.73	0.74	0.77	0.87	0.74	0.07
SPA	0.63	0.72	0.75	0.72	0.72	0.82	0.91	0.75	0.08
SPS	0.64	0.72	0.75	0.73	0.73	0.82	0.91	0.76	0.08
WLCA	0.62	0.72	0.76	0.78	0.77	0.87	0.92	0.78	0.09
LCA	0.64	0.76	0.78	0.78	0.78	0.87	0.93	0.79	0.08
ULCA	0.65	0.77	0.78	0.80	0.79	0.88	0.93	0.80	0.08
mean	0.61	0.71	0.72	0.74	0.74	0.81	0.89		
std	0.04	0.03	0.05	0.03	0.03	0.05	0.04		

Table 5: Prediction performance (AUC) of inducers with NB classifier for different datasets. AUCs averaged over the three ontologies and 10-fold cross-validated.

The highest accuracy over all datasets was achieved by the *ULCA* inducer with a mean AUC of 0.80, closely followed by the *LCA* inducer with an AUC of 0.79. The similarity in accuracy of those two inducers is not surprising, since the *LCA* and *ULCA* inducer are algorithmically very similar. The only difference is that the latter includes the terms from the annotating terms up to the lowest common ancestor, while the former excludes them. The lowest prediction accuracy (mean AUC = 0.67) was achieved by the *AC* inducer, showing that the classical approach of inferring interactions based on shared term sets is actually the worst method. Simply taking the union of all annotating terms, as implemented by the AL inducer, leads to considerably higher prediction accuracies (mean AUC = 0.74).

The superior performance of the AL inducer over the AC inducer is likely due to the fact that an intersection of term sets reduces the number of GO terms indicated within the feature vector, and thereby reduces the opportunity for the classifier to detect correlations between annotations. For instance, let two proteins be annotated as existing in different cellular compartments C1 and C2 respectively. The AC inducer constructs the intersection, which is the empty set $\emptyset$, leaving no information for the classifier to exploit. The AL inducer constructs the union {C1,C2} and the classifier can therefore learn whether proteins within those compartments tend to interact or not. Of course, a single protein could be annotated with both compartments, which would complicate the learning task; however, in many cases the learner can distinguish whether the annotation originates from a single protein, or from two proteins. For example, if C1 is a subsumer of C2 then C1 and C2 would never appear together as annotations of a single protein, and must originate from two different proteins.

RF classifiers have achieved high accuracies in PPI prediction tasks and we were interested in the performance of the different inducers when combined with a RF classifier. Table 6 lists the prediction accuracies (AUCs) achieved with the RF classifier.

Inducer	*MM*	*DM*	*AT*	*HS*	*SC*	*SP*	*EC*	mean	std
AC	0.55	0.66	0.61	0.69	0.68	0.69	0.80	0.67	0.07
ACA	0.53	0.67	0.66	0.78	0.77	0.80	0.89	0.73	0.11
OLCA	0.54	0.68	0.67	0.80	0.78	0.80	0.90	0.74	0.11
AA	0.62	0.71	0.73	0.80	0.81	0.77	0.87	0.76	0.07
AL	0.63	0.72	0.72	0.83	0.84	0.80	0.88	0.77	0.08
SPA	0.64	0.72	0.74	0.83	0.85	0.81	0.91	0.79	0.08
SPS	0.64	0.72	0.74	0.83	0.85	0.82	0.91	0.79	0.08
WLCA	0.64	0.73	0.76	0.86	0.87	0.88	0.92	0.81	0.09
LCA	0.63	0.77	0.77	0.87	0.87	0.87	0.93	0.82	0.09
ULCA	0.65	0.78	0.78	0.87	0.88	0.88	0.93	0.82	0.09
mean	0.61	0.72	0.72	0.82	0.82	0.81	0.89		
std	0.04	0.04	0.05	0.05	0.06	0.05	0.04		

Table 6: Prediction performance (AUC) of inducers with RF classifier for different datasets. AUCs averaged over the three ontologies and 10-fold cross-validated.

Apart from the *AC* inducer, employing the RF classifier generally leads to slightly higher prediction accuracies for inducers when compared to the NB classifier. The low accuracy of the *AC* inducer even in combination with an RF classifier again indicates that a shared-term approach is unsuitable to measure similarities between GO annotations. The improved performance of the RF classifier is largely due to higher accuracies on the human (*HS*) and the yeast (*SC*) datasets (AUCs increase from 0.74 to 0.82). For the other datasets essentially no improvement can be observed, which is likely due to the small size of these datasets.

In the following we furthermore compare the prediction accuracies of inducers on Park's yeast dataset (YC), which is independent from our datasets. Figure 3 shows the accuracies (AUCs) of the inducers with a Naive Bayes classifier (NB), and Figure 4 shows the results for a combination with a Random Forest (RF) classifier.

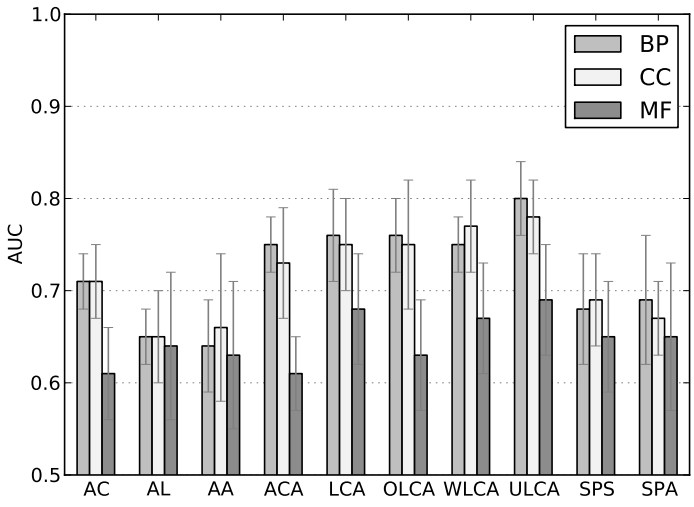

Figure 3: Prediction accuracies (AUC) of different inducers with a NB classifier for the three ontologies (BP, CC, MF) on Park's dataset (*YC*). Results are 10-fold cross-validated. Error bars show standard deviation.

The results are in good agreement with results based on the SC dataset. A notable exception is

the performance of the AC inducer, which achieves higher accuracies then the AL and AA inducers when the inducers are combined with a NB classifier. In combination with an RF classifier (see Figure 4), however, the AL and AA inducers are superior, especially on the MF ontology. It is also noteworthy that the accuracies for the RF classifier are consistently higher than those of the NB classifier, with the exception of the AC inducer, for which no improvement can be observed.

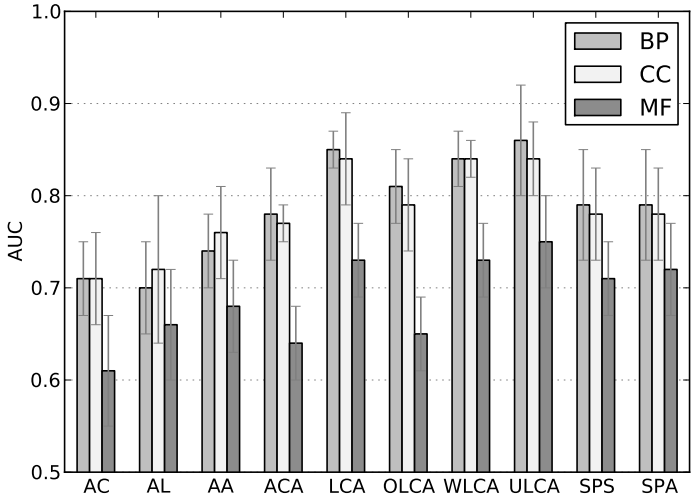

Figure 4: Prediction accuracies (AUC) of different inducers with a RF classifier for the three ontologies (BP, CC, MF) on Park's dataset (YC). Results are 10-fold cross-validated. Error bars show standard deviation.

4 Comparison of semantic similarity measures

We introduced the concept of inducers to combine semantic similarity measures (SSMs) and machine learning (ML) methods with the aim to improve prediction accuracies. To enable a fair comparison between SSMs and the inducer approach, we firstly performed an evaluation to identify the most-accurate SSM for PPI prediction. While a large number of SSMs has been proposed, for many applications Resnik's measure (Resnik, 1995) with different aggregation strategies still performs best or is among the top-performers. Pesquita *et al.* (2009) reviewed SSMs applied to biomedical ontologies and in 5 of 11 studies Resnik's method with BMA or MAX as aggregation strategy was identified as the best performer. For PPI prediction Resnik's with MAX aggregation was listed as the best performer. We therefore focused our comparison largely on Resnik's measure with different aggregation strategies and variations of it. In the following we first describe the different SSMs evaluated before presenting the actual evaluation results.

Many SSMs utilize the *Information Content* (IC) of a (GO) term t, which is defined as

$$IC(t) = -\log \frac{n(t)}{N}, \tag{1}$$

where $n(t)$ is the number of times term t appears in the dataset either directly or as ancestor of another term and N is the number of terms within the dataset. Resnik's similarity between two terms t_1 and t_2 is then calculated as the maximum IC over the set of common ancestor terms $C(t_1, t_2)$ of those two terms:

$$RES(t_1, t_2) = \max_{t \in C(t_1, t_2)} IC(t). \tag{2}$$

Jiang and Conrath (1997), Lin (1998), and Schlicker *et al.* (2006) have proposed improvements of Resnik's measure, where Jiang's measure is defined as

$$JIA(t_1, t_2) = 1 - IC(t_1) + IC(t_2) - 2 \cdot RES(t_1, t_2), \tag{3}$$

Lin's measure is defined as

$$LIN(t_1, t_2) = \frac{2 \cdot RES(t_1, t_2)}{IC(t_1) + IC(t_2)}, \tag{4}$$

and Schlicker's measure is defined as

$$SCH(t_1, t_2) = (1 - p(a)) \cdot LIN(t_1, t_2), \tag{5}$$

with $p(a)$ being the probability of the common ancestor of terms t_1 and t_2. Gentleman (2006) implemented a method that defines the similarity between two sets of terms S_1 and S_2 as the size of the intersection set of the ancestor terms divided by the size of the union set, with $A(S)$ being the set of ancestors of term set S:

$$GEN(S_1, S_2) = \frac{|A(S_1) \cap A(S_2)|}{|A(S_1) \cup A(S_2)|}. \tag{6}$$

Pesquita *et al.* (2008) proposed an improved method and replaced the sizes of the term sets by the sums over their ICs:

$$PES(S_1, S_2) = \frac{\sum_{t \in A(S_1) \cap A(S_2)} IC(t)}{\sum_{t \in A(S_1) \cup A(S_2)} IC(t)}. \tag{7}$$

To complete the suite of node-based SSMs described above we also implemented the edge-based measure SP, which computes term similarity as the length of the shortest path between two terms:

$$SP(t_1, t_2) = |shortestpath(t_1, t_2)|. \tag{8}$$

SSMs were originally developed to measure word similarities and are therefore confined to compare only two terms at a time. Proteins however, are annotated by multiple GO terms and some SSMs need to be modified to measure the similarity between two sets of terms instead of individual terms. The three most-common strategies compute similarities between all possible pairings of GO terms within the two sets and then aggregate the similarity values by calculating the *maximum*, the *average* or the *best-match-average*.

Table 7 contains the prediction accuracies for the different SSMs in combination with each aggregation strategy. Results are computed on the yeast dataset SC for each of the three ontologies. The methods by Gentleman (2006) (*GEN*) and Pesquita *et al.* (2008) (*PES*) do not require an aggregation strategy but we were interested in the impact on the performance and report those results as well.

Method	Aggregator	BP	std	CC	std	MF	std	mean
RES	max	0.61	0.03	0.59	0.02	0.53	0.01	0.58
RES	avg	0.60	0.02	0.59	0.04	0.53	0.02	0.57
RES	bma	0.73	0.03	0.76	0.03	0.62	0.04	0.70
JIA	max	0.55	0.04	0.55	0.03	0.51	0.02	0.54
JIA	avg	0.57	0.03	0.57	0.02	0.51	0.02	0.55
JIA	bma	0.58	0.02	0.65	0.02	0.52	0.03	0.58
LIN	max	0.61	0.02	0.59	0.03	0.53	0.02	0.58
LIN	avg	0.61	0.05	0.58	0.03	0.53	0.02	0.57
LIN	bma	0.73	0.02	0.75	0.03	0.62	0.03	0.70
SCH	max	0.61	0.02	0.59	0.03	0.53	0.02	0.58
SCH	avg	0.61	0.02	0.59	0.03	0.53	0.02	0.58
SCH	bma	0.73	0.03	0.75	0.04	0.62	0.03	0.70
PES	max	0.61	0.04	0.59	0.04	0.53	0.02	0.58
PES	avg	0.61	0.04	0.59	0.02	0.53	0.02	0.58
PES	bma	0.74	0.03	0.74	0.03	0.62	0.03	0.70
PES	n/a	0.73	0.03	0.74	0.02	0.62	0.02	0.70
GEN	max	0.60	0.02	0.58	0.02	0.52	0.02	0.57
GEN	avg	0.59	0.03	0.58	0.03	0.52	0.02	0.56
GEN	bma	0.76	0.03	0.76	0.03	0.67	0.02	**0.73**
GEN	n/a	0.76	0.01	0.75	0.03	0.65	0.03	0.72
SP	max	0.59	0.03	0.57	0.03	0.52	0.02	0.56
SP	avg	0.58	0.03	0.57	0.03	0.52	0.02	0.56
SP	bma	0.75	0.02	0.77	0.03	0.66	0.03	**0.73**

Table 7: Prediction performance (AUC) of SSMs on the yeast dataset (*SC*) for all three ontologies (BP, CC, MF) using different aggregators such as *maximum* (max) , *average* (avg) or *best-match-average* (bma). *PES* and *GEN* can also be used without an aggregator (n/a). Results are 10-fold cross-validated.

The results above show firstly that *best-match-average* is the best aggregation strategy in all cases, and secondly that all SSMs – apart from Jiang's measure – achieve very similar prediction accuracies. With *best-match-average* the AUC is typically 0.70 and only Gentleman's measure and the Shortest-path method achieve a slightly higher AUC of 0.73.

All SSMs evaluated show accuracies considerably lower than those of the *ULCA* inducer with a NB or RF classifier. Since inducers are very similar to SSMs, the improvement in performance is likely due to the combination with a machine learning classifier. The latter has two advantages over the SSMs evaluated. First, machine learning classifiers are supervised and take negative examples into account and second, they can model higher order statistics, while the SSMs evaluated are unsupervised and limited to term probabilities.

To validate the importance of the machine learning component, we reduced the *ULCA* method to a SSM by removing the classifier component and replacing it by the information content *IC* as it is used in Resnik's and related methods. More precisely we defined a SSM named *ULCA'*, which is computed as the maximum *IC* over the term set induced by the *ULCA* inducer:

$$ULCA'(t_1, t_2) = \max_{t \in ULCA(\{t_1\},\{t_2\})} IC(t). \tag{9}$$

The following Table 8 contains the prediction accuracies (AUCs) of Resnik's method (*RES*) with different aggregation strategies, the *ULCA* based measure *ULCA'* with different aggregators and the original ULCA inducer in combination with a Naive Bayes (NB) and a Random Forest (RF) classifier.

Method	Agg./ML	BP	CC	MF	mean
RES	max	0.61	0.59	0.53	0.58
RES	avg	0.60	0.59	0.53	0.57
RES	bma	0.73	0.76	0.62	0.70
ULCA′	max	0.58	0.58	0.50	0.55
ULCA′	avg	0.57	0.57	0.50	0.55
ULCA′	bma	0.69	0.73	0.50	0.64
ULCA	NB	0.82	0.82	0.73	0.79
ULCA	RF	0.91	0.91	0.82	0.88

Table 8: Prediction performances (AUC) on the yeast dataset (*SC*) over the three ontologies for Resnik's method and the inducer with (*ULCA*) and without (*ULCA′*) classifier component. Results are 10-fold cross-validated.

The results above show that the prediction accuracy of the *ULCA* method drops even below the level of Resnik's similarity measure when the machine learning component is replaced by the information content (IC). Both the NB classifier and IC are based on conditionally independent estimates of term probabilities, but the NB classifier achieves a substantially higher AUC, which we attribute to the fact that it is supervised and in contrast to term probabilities takes negative examples into account. Random Forests can furthermore model dependencies between input variables, which might explain their superior performance in comparison to the NB classifier.

Inducers combined with ML classifiers show relatively higher prediction accuracies on the MF ontology than the unsupervised methods (*RES*, *ULCA′*). This improved performance is likely due to the fact that unsupervised methods always correlate similar GO annotation with high confidence of interaction. In case of BP and CC annotation this is an appropriate model, but not for MF annotation. For instance, two proteins might be annotated as being enzymes (similar annotation) but an interaction is not likely and a low confidence score should be generated. On the other hand, a transporter protein and its cargo may have no similar molecular function but yet interact. Supervised methods can learn to make such distinctions based on training data, whereas unsupervised methods have to rely on similarity in GO annotation alone. In short, the data show that similar annotation of MF is not a good indicator for PPI, while similar BP or CC annotation is. Consequently, unsupervised methods perform well on BP and CC annotation but fail on MF data, whereas supervised methods can exploit dependencies within GO annotation beyond mere similarity of terms, which leads to improved prediction accuracies, especially for MF annotation.

5 Cross-species predictions

Park (2009) evaluated the cross-species prediction accuracy of sequenced based methods by training a predictor on one species but applying it to a different species and found sequence-based methods to achieve low cross-species prediction accuracies.

GO is designed to be a species independent annotation system and we therefore expected to achieve good cross-species prediction accuracies. The three matrices below show the cross-species prediction accuracies (AUCs) for the *ULCA* inducer with a NB classifier for the three GO sub-ontologies. Rows represent the datasets the predictor is trained on and columns are labeled by the datasets the predictor has been applied to (tested on). The darker the cell color the lower is the prediction accuracy, and values along the diagonal represent the self-test AUCs (training and test on the same dataset).

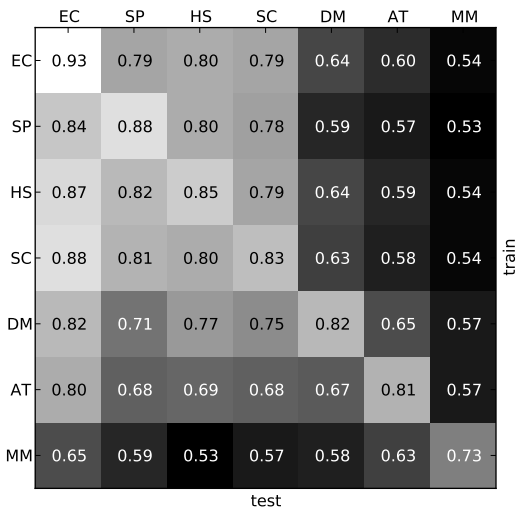

Figure 5: Cross-species prediction accuracies (AUCs) of the *ULCA* inducer with a NB classifier on the BP ontology. Rows are labeled with training data and columns are labeled with test data.

Figure 5 depicts the cross-species accuracies of the *ULCA* inducer on the BP ontology. The best prediction accuracies were achieved for *E. coli* (*EC*) — see first column of Figure 5 — and the lowest accuracies were achieved on the mouse PPI dataset (*MM*) – last column. There is a clear tendency for AUCs below the diagonal to be higher than the corresponding AUC above the diagonal (see *EC* versus *SP* for instance), indicating that the prediction accuracy on the target species largely depends on the dataset the predictor is tested on (target species). Consequently, the prediction accuracy on the target species is high, when the self-test accuracy for that species is high, and low otherwise. For the three smallest datasets (*DM*, *AT*, *MM*), we find substantially lower prediction accuracies, and we suspect the inferior performance is partially due to the small network size.

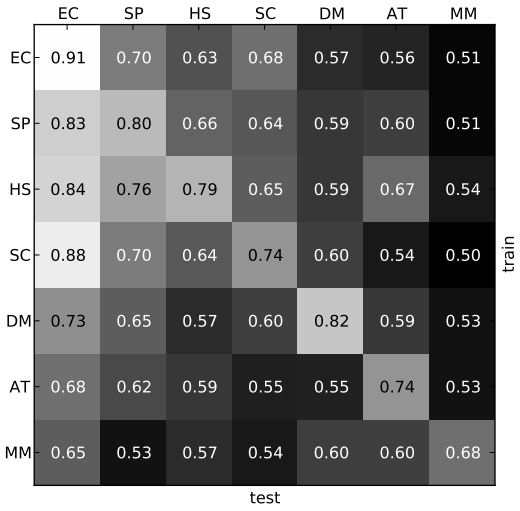

Figure 6: Cross-species prediction accuracies (AUCs) of the *ULCA* inducer with a NB classifier on the CC ontology. Rows are labeled with training data and columns are labeled with test data.

Figure 6 shows the cross-species accuracies of the *ULCA* inducer on the CC ontology. The trend is similar to that observed on the BP ontology. Prediction accuracies are slightly lower overall but the accuracy on a target species (test) again appears to be largely related to the self-test accuracy of that species. As before, accuracies on the three smallest datasets (*DM*, *AT*, *MM*) tend to be considerably lower than for the other four datasets.

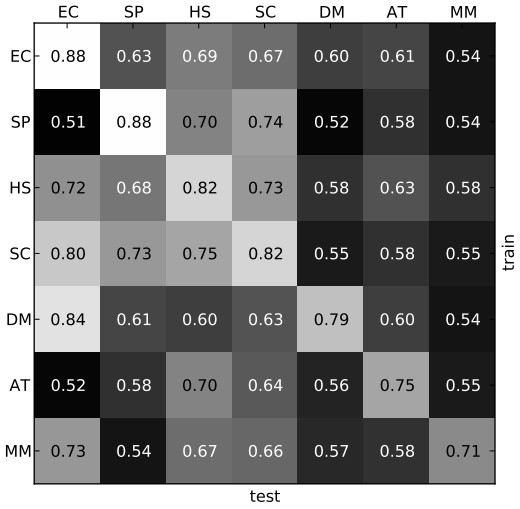

Figure 7: Cross-species prediction accuracies (AUCs) of the *ULCA* inducer with a NB classifier on the MF ontology. Rows are labeled with training data and columns are labeled with test data.

Figure 7 shows the cross-species accuracies of the *ULCA* inducer on the MF ontology. The picture becomes more complex in comparison to Figure 5 and Figure 6. Generally, prediction accuracies are lower than those for the BP and CC ontology, but there are exceptions, most

notably the rather high AUC of 0.84 when training on the DM dataset and testing on EC. Also surprising is the low accuracy (AUC = 0.51), when training on the SP dataset and testing on EC, since high AUCs have been achieved on the other ontologies in this case.

To summarize, the cross-species prediction accuracies of the GO-driven ULCA inducer for the larger networks are good and the performance on the target species depends on its self-test accuracy. In general, prediction accuracies on the BP ontology are higher than those of the CC ontology, which are higher than those on the MF ontology. With respect to species we find the best performance for *E. coli* (*EC*) and the worst for *M. musculus* (MM). Good cross-species accuracies have been achieved for *E. coli* (*EC*), *S. cerevisiae* (*SC*) and *H. sapiens* (*HS*). For instance, training the predictor on yeast (*SC*) results in an AUC of 0.80 on the human (*HS*) dataset, when using the *ULCA* inducer with an NB classifier on the BP ontology.

6 Comparison with GO slims

GO slims are subsets of the complete GO term set that provide a means to simplify annotation, to reduce computational expense and to improve the predictive power of shared term approaches. Existing annotations that are drawn from the complete GO term set are reduced to a GO slim term set by mapping tools such as `map2slim.pl`, which has similarities to ancestral inducers but is algorithmically more complex (see `http://search.cpan.org/~cmungall/go-perl/scripts/map2slim` for details).

An interesting question to ask is whether a reduced term set can improve the prediction accuracy of inducers. For instance, in case of the *AC* inducer, which is based on the set of shared terms, a smaller term set increases sensitivity, since it becomes more likely that two proteins share GO annotations. On the other hand, if the term set becomes too small the inducer will start to loose specificity. To study the impact of GO slims on the prediction accuracy of inducers we downloaded four GO slims, listed in Table 9, from `http://www.geneontology.org/GO.slims.shtml`.

Label	Name	Revision	#terms	description
GOA	`goslim_goa`	1.854	62	UniProtKB-GOA
YEA	`goslim_yeast`	1.822	92	Yeast slim
GEN	`goslim_generic`	1.862	127	Generic GO slim
PIR	`goslim_pir`	1.99	467	Protein Inf. Res.

Table 9: GO slims downloaded from Uniprot, with their numbers of GO terms.

We then compared the prediction accuracies of the *AC* and *ULCA* inducers, using a NB classifier, for the four GO slims (see Table 9) with the complete term set (*ALL*) on the *SC* dataset. Figure 8 depicts the results of this comparison.

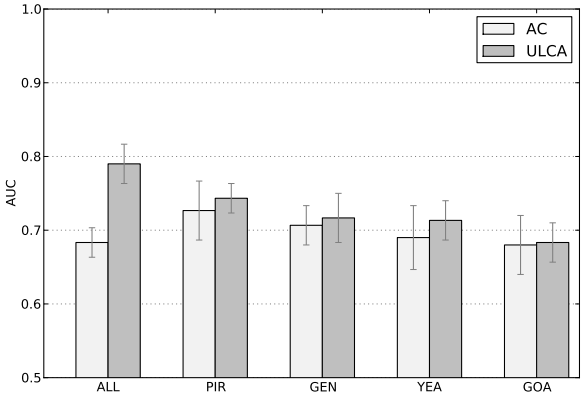

Figure 8: Comparison of AC and $ULCA$ inducers with NB classifier for slim (*GEN, GOA, PIR, YEA*) and complete (*ALL*) term sets on the SC dataset. AUC is averaged over the three ontologies. Results are 10-fold cross-validated. Error bars show standard deviation.

Since the *ULCA* inducer requires a fully connected DAG, all GO slims were augmented with the missing terms to reconstruct DAGs up to their root terms. This augmentation has no impact on the AC method, since those terms are not used for annotation, but it enables a comparison with the *ULCA* method on a slim term set.

As Figure 8 reveals, the prediction accuracy of the AC inducer improves on most GO slims (*GEN, PIR, YEA*) in comparison to the complete term set (*ALL*) – with the notable exception of the *GOA* slim. In contrast, slims have a negative impact on the performance of the *ULCA* inducer in all cases. The highest overall AUC of 0.79 is achieved by the *ULCA* inducer on the complete term set, outperforming all slims significantly.

7 Impact of evidence codes

Rogers and Ben-Hur (2009) highlight the importance of taking into account GO evidence codes when comparing classifiers that rely on different information sources. In our case, including GO annotations derived from physical (IPI) or genetic (IGI) interactions could give the GO-driven inducer methods an unfair advantage over the sequence-based methods we compare against. To study the effect, we measured the prediction accuracy of the *ULCA* inducer on Park's dataset (YP) when including (+IFI) or excluding (-IFI) GO annotation inferred from interactions (IFI).

+IFI	-IFI	Ontology
0.77	0.74	MF
0.86	0.84	CC
0.88	0.86	BP
0.88	0.86	CC+MF
0.89	0.86	BP+MF
0.90	0.89	BP+CC
0.91	0.90	BP+CC+MF

Table 10: Prediction performance (AUC) of the *ULCA* inducer with RF classifier on Park's dataset (YP) for different (combinations of) sub-ontologies with GO annotations inferred from interactions (IFI) included (+IFI) or excluded (-IFI). Results are 4-fold cross-validated.

The results in Table 10 reveal a small drop in prediction accuracy for individual ontologies when

IFI annotations are excluded (-IFI) but the effect is negligible when all ontologies are combined (BP+CC+MF).

8 Feature encoding

Most inducers studied here are based on shared GO terms, e.g. shared ancestors, and we encoded the presence or absence of shared GO terms in an induced term set by a binary feature vector. However, some inducers construct the union of the GO term annotations of two proteins but the binary representation does not distinguish between the case in which a GO term is assigned to only one protein, and that in which a GO term is assigned to none of the two proteins. This loss of information can be avoided by encoding GO terms within the feature vector differently – for instance by using 1 for GO terms assigned to one of the two proteins, 2 for GO terms assigned to both proteins, and and 0 otherwise (ternary encoding).

Table 11 shows the prediction accuracies (measured by AUC, MCC, F_1 and ACC) of the two basic inducers (*AC*, *AL*) and the *ULCA* inducer with binary or ternary encoding. The *AC* inducer serves as baseline and control. It computes the set of shared GO terms, and the two different encodings should not affect its prediction accuracy significantly[2]. The *AL* inducer, on the other hand, constructs the union of term sets, and the results show a substantial improvement in prediction accuracy for the ternary encoding (AUC improves from 0.78 to 0.88) but remain lower than that those of the *ULCA* inducer. The term sets generated by the *ULCA* inducer contain intersections of term sets (lowest common ancestor terms) and unions of term sets (terms up to the lowest common ancestors), but the results show no effect of the encoding on its prediction performance.

Encoding	AUC	std	MCC	F_1	ACC	Inducer
binary	0.78	0.01	0.55	0.68	0.75	AC
binary	0.78	0.02	0.37	0.71	0.68	AL
binary	0.90	0.01	0.65	0.82	0.83	ULCA
ternary	0.78	0.02	0.54	0.69	0.75	AC
ternary	0.88	0.01	0.62	0.80	0.81	AL
ternary	0.90	0.02	0.65	0.82	0.83	ULCA

Table 11: Prediction performance of inducers with RF classifier on Park's dataset (*YP*) for different encodings (binary, ternary) averaged over the three ontologies. Results are 4-fold cross-validated.

9 Impact of size of negative dataset

Protein-protein interaction networks are known to have many more interactions (positives) than non-interactions (negatives). Qi *et al.* (2006) estimate a ratio of 600:1 but sample sets of this size are generally very large and it becomes too time-consuming to train a machine-learning classifier. For instance, our *SC* yeast dataset contains 15238 interactions (positives) and a complete sample set with 601 times that number would contain over 9 million samples. Typically, classifiers for PPI inference are therefore trained on balanced datasets (equal number of positive and negative samples) (Ben-Hur and Noble, 2005; Qiu and Noble, 2008).

In this section we investigate how larger negative sets affect the prediction performance and different performance metrices. For negative-to-positive ratios of 1:1, 5:1, 10:1, 50:1 and 100:1, we trained a Naive Bayes classifier in combination with two basic inducers (*AC*, *AL*) and the *ULCA* inducer on Park's *YC* dataset, using 4-fold cross-validation. Table 12 shows the prediction performances measured by the Area Under the ROC (AUC), the AUC up to the first 50 false positives (AUC_{50}), Matthews correlation coefficient (MCC), the F1-Score, sensitivity (SN) or

[2]The observed slight variations in prediction accuracy are due to cross-validation

recall (RC), specificity (SP) and precision (PR). Note that sensitivity and recall are two different terms for the same measure and are therefore reported in one column (SN/RE).

ratio	AUC ± std	AUC_{50} ± std	MCC	F_1	SN/RE	SP	PR	Inducer
1:1	0.78 ± 0.02	0.58 ± 0.13	0.55	0.67	0.53	0.97	0.94	AC
1:1	0.68 ± 0.01	0.60 ± 0.07	0.27	0.61	0.58	0.68	0.65	AL
1:1	0.82 ± 0.01	0.59 ± 0.10	0.48	0.75	0.77	0.71	0.72	ULCA
5:1	0.79 ± 0.01	0.50 ± 0.00	0.58	0.57	0.41	0.99	0.93	AC
5:1	0.68 ± 0.01	0.61 ± 0.18	0.23	0.34	0.32	0.89	0.37	AL
5:1	0.83 ± 0.01	0.87 ± 0.28	0.40	0.50	0.75	0.75	0.38	ULCA
10:1	0.79 ± 0.01	0.63 ± 0.43	0.58	0.55	0.40	1.00	0.91	AC
10:1	0.68 ± 0.01	0.61 ± 0.11	0.17	0.24	0.24	0.92	0.24	AL
10:1	0.83 ± 0.01	0.93 ± 0.26	0.32	0.36	0.74	0.76	0.24	ULCA
50:1	0.79 ± 0.01	0.63 ± 0.43	0.52	0.49	0.37	1.00	0.74	AC
50:1	0.68 ± 0.01	0.78 ± 0.29	0.07	0.09	0.15	0.96	0.07	AL
50:1	0.83 ± 0.01	1.00 ± 0.00	0.17	0.12	0.73	0.78	0.06	ULCA
100:1	0.79 ± 0.03	1.00 ± 0.00	0.48	0.46	0.36	1.00	0.64	AC
100:1	0.68 ± 0.01	0.92 ± 0.14	0.05	0.06	0.13	0.97	0.04	AL
100:1	0.83 ± 0.00	1.00 ± 0.00	0.12	0.06	0.72	0.79	0.03	ULCA

Table 12: Comparison of inducers and metrices for different negative-to-positive ratios

The AUC is a measure that is insensitive to the class distribution and therefore remains stable for different negatives-to-positives ratios. Apparently, the larger amount of (negative) training data does not improve the AUC of the Naive Bayes classifier. In contrast, the AUC_{50} score clearly improves for all inducers with increasing numbers of negatives, simply because with more negative sample, false positives become less likely. The AUC_{50} is an interesting metric for practical applications, in which a few, highly confident predictions are wanted, e.g. for experimental confirmation of predicted interactions. Note, however, the large standard deviations and fluctuations that render the AUC_{50} a unreliable measure of performance.

Sensitivity, recall, specificity and precision are metrics that depend on a threshold. For a classifier that reports confidence values in the interval [0,1], such as a Naive Bayes classifier, commonly a threshold of 0.5 is used. MCC and F_1 are derived scores that therefore also depend on the chosen threshold. The results in Table 12 seem to indicate that for more unbalanced sample sets the AC inducer outperforms the other two inducers. However, without optimizing the threshold, results between different inducers are not comparable and the results are misleading. For the same inducer there is a clear trend of lower MCCs and F_1 scores for larger numbers of negatives.

10 Pseudocode

This section provides pseudocode describing the inducer system depicted in Figure 1 of the paper.

```
TRAINING
1)   load GO_graph
2)   load PPI_network
3)   annotate proteins in PPI_network with GO terms
4)   for each interaction in PPI_network
5)     compute induced term_set in GO_graph for interaction
6)     create feature_vector from induced term_set
7)     store feature_vector and class label in sample_set
8)  train ML_classifier on sample_set
9)  store trained ML_classifier
```

Interactions within the PPI network are labeled as *interacting* or *non-interacting* and are described by a protein pair and the corresponding label. For the ML classifier we recommend a Random Forest classifier, e.g. the Random Forest classifier available within the WEKA library (Witten *et al.*, 2005). The inducer in Step 5 is implemented according to one of the equations that define the different inducers within the paper. We achieved the best results with the *ULCA* inducer (Equation 8) and its pseudocode is given below.

```
ULCA INDUCER
1) let S_1 and S_2 be sets of GO terms annotating a pair of proteins (P_1,P_2)
2) compute ancestral term sets A_1 = A(S_1) and A_2 = A(S_2) from GO graph
3) compute set C of common terms in A_1 and A_2
4) find term in C with maximal depth D in GO graph
5) construct union U of A_1 and A_2
6) filter U for terms with a depth >= D
7) return the induced term_set
```

During the query phase the ML classifier trained in the previous phase is loaded, induced term sets for all possible parings of proteins are generated and fed into the classifier. Output of the query phase is a list of protein pairs with a confidence values, typically in the interval [0,1], that indicates interaction or non-interaction. The pseudocode of the query phase is shown below.

```
QUERYING
1)   load trained ML_classifier
2)   load GO_graph
3)   load list of proteins
4)   annotate proteins in list with GO terms
5)   for each possible protein_pair derived from protein_list
6)     compute induced term_set in GO_graph for protein_pair
7)     create feature_vector from induced term_set
8)     query ML_classifier with feature_vector
9)     store prediction result
```

11 Related work on SSMs

SSMs were originally developed to measure word similarities over lexical taxonomies (WordNet). Since then SSMs have been applied to Gene Ontology to measure protein similarity (Duan *et al.*, 2006; Lord *et al.*, 2003; Pesquita *et al.*, 2008), infer genetic interactions (Lee and Lee, 2005), predict subnuclear protein localization (Lei and Dai, 2006), assess the quality of interactomes (Brown and Jurisica, 2005; Chen and Xu, 2003), or correlate with gene expression (Sevilla *et al.*, 2005; Xu *et al.*,

2008), protein family association (Couto *et al.*, 2007), pathways (Sheehan *et al.*, 2008; Wang *et al.*, 2007) and functional similarity (Pozo *et al.*, 2008; Xu *et al.*, 2008; Mistry and Pavlidis, 2008; Tao *et al.*, 2007; Yu *et al.*, 2007; Schlicker *et al.*, 2006). Ferreira *et al.* (2010) employed SSMs to classify chemical compounds and optimized the threshold over the SSM output, rendering the method a supervised SSM.

12 Table of acronyms

Acronym	Description
AC	All Common (Terms)
ACA	All Common Ancestor (Terms)
AL	All Labels
AUC	Area Under the ROC
BP	Biological Process (Ontology)
CC	Cellular Compartment (Ontology)
DAG	Directed Acyclic Graph
GEN	Gentleman's semantic similarity measure
GO	Gene Ontology
JIA	Jain's semantic similarity measure
KNN	K Nearest Neighbor (Classifier)
LCAT	Lowest Common Ancestor Term
LIN	Lin's semantic similarity measure
LCA	Lowest Common Ancestor
MF	Molecular Function (Ontology)
MIPS	Munich Information Center for Protein Sequences
ML	Machine Learning
OLCA	Only Lowest Common Ancestor
PES	Pesquita's semantic similarity measure
PPI	Protein-protein interaction
RES	Resnik's semantic similarity measure
RF	Random Forest (Classifier)
ROC	Receiver Operator Characteristic
SCH	Schlicker's semantic similarity measure
SP	Shortest Path
SPA	Shortest Path All
SPS	Shortest Path Single
SSM	Semantic Similarity Measure
SVM	Support Vector Machine
ULCA	Up to Lowest Common Ancestor
WLCA	Without Lowest Common Ancestor

Table 13: Table of acronyms.

References

Ben-Hur,A. and Noble,W.S. (2005) Kernel methods for predicting protein-protein interactions. *Bioinformatics*, **21**, i38-i46.

Brown,K.R. and Jurisica,I. (2005) Online predicted human interaction database. *Bioinformatics*, **21**, 2076–2082.

Chen,Y. and Xu,D. (2003) Computational analyses of high-throughput protein-protein interaction data. *Curr. Protein Pept. Sci.*, **4**, 159-181.

Couto,F. *et al.* (2007) Measuring semantic similarity between gene ontology terms. *Data & Knowledge Engineering*, **61**, 137-152.

Duan,Z-H. *et al.* (2006) The relationship between protein sequences and their gene ontology functions. *BMC Bioinformatics*, **7** (Suppl 4), S11.

Ferreira,J.D. *et al.* (2010) Semantic similarity for automatic classification of chemical compounds. *PLoS Comput Biol.*, **6**, e1000937.

Gentleman,R. (2006) GOstats library in Bioconductor.

Guo,S. *et al.* (2008) Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. *Nucleic Acids Res.*, **36**, 3025–3030.

Jensen,L.J. *et al.* (2009) STRING 8 - a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Res.*, **37**, D412-D416.

Jiang,J.J. and Conrath,D.W. (1997) Semantic similarity based on corpus statistics and lexical taxonomy. *Proc of 10th International Conference on Research in Computational Linguistics*, Taiwan, 19–33.

Lee,P.H. and Lee,D. (2005) Modularized learning of genetic interaction networks from biological annotations and mRNA expression data. *Bioinformatics*, **21**, 2739–2747.

Lei,Z. and Dai,Y. (2006) Assessing protein similarity with gene ontology and its use in subnuclear localization prediction. *BMC Bioinformatics*, **7**, 491.

Lin,D. (1998) An information-theoretic definition of similarity. In *Proc. of the 15th International Conference on Machine Learning. Morgan Kaufmann, San Francisco, California*, 296-304.

Lord,P.W. *et al.* (2003) Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, **19**, 1275-1283.

Martin,S. *et al.* (2005) Predicting protein-protein interactions using signature products. *Bioinformatics*, **21**, 218–226.

Mistry,M. and Pavlidis,P. (2008) Gene ontology term overlap as a measure of gene functional similarity. *BMC Bioinformatics*, **9**, 327.

Park Y. (2009) Critical assessment of sequence-based protein-protein interaction prediction methods that do not require homologous protein sequences. *BMC Bioinformatics*, **10**, 419.

Pesquita,C. *et al.* (2008) Metrics for GO based protein semantic similarity: a systematic evaluation. *BMC Bioinformatics*, **9** (Suppl 5), S4.

Pesquita,C. *et al.* (2009) Semantic similarity in biomedical ontologies. *PLOS Comput Biol.*, **5**, e1000443.

Pitre,S. *et al.* (2006) PIPE: a protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs. *BMC Bioinformatics*, **7**, 365.

Pozo,A.D. *et al.* (2008) Defining functional distances over gene ontology. *BMC Bioinformatics*, **9**,50.

Qi,Y. *et al.* (2006) Evaluation of different biological data and computational methods for use in protein interaction prediction. *Proteins*, **63**, 490–500.

Qiu,J. and Noble,W.S. (2008) Predicting co-complexed protein pairs from heterogeneous data. *PLoS Comput Biol.*, **4**, e1000054

Resnik,P. (1995) Using information content to evaluate semantic similarity in a taxonomy. *Proc. of the 14th International Joint Conference on Artificial Intelligence*, 448-453.

Rogers,M.F. and Ben-Hur,A. (2009) The use of gene ontology evidence codes in preventing classifier assessment bias. *BMC Bioinformatics*, **25**, 1173–1177.

Schlicker,A. *et al.* (2006) A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics*, **7**, 302.

Sevilla,J.L. *et al.* (2005) Correlation between gene Expression and GO semantic similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **2**, 330-338.

Sheehan,B. *et al.* (2008) A relation based measure of semantic similarity for gene ontology annotations. *BMC Bioinformatics*, **9**, 468.

Shen,J. *et al.* (2007) Predicting protein-protein interactions based only on sequences information. *Proc. Natl Acad. Sci. U.S.A.*, **104**, 4337–4341.

Tao,Y. *et al.* (2007) Information theory applied to the sparse gene ontology annotation network to predict novel gene function. *Bioinformatics*, **23**, 529-538.

Wang,J.Z. *et al.* (2007) A new method to measure the semantic similarity of GO terms. *Bioinformatics*, **10**, 1274–1281.

Witten,I.H. and Frank,E. (2005) Data Mining: Practical machine learning tools and techniques. *Morgan Kaufmann, San Francisco, California*, 2nd Edition.

Xu,T. *et al.* (2008) Evaluation of GO-based functional similarity measures using *S. cerevisiae* protein interaction and expression profile data. *BMC Bioinformatics*, **9**, 472

Yu,H. *et al.* (2007) Total ancestry measure: quantifying the similarity in tree-like classification, with genomic applications. *Bioinformatics*, **23**, 2163–2173.