BIOINFORMATICS ORIGINAL PAPER

Vol. 28 no. 1 2012, pages 69–75
doi:10.1093/bioinformatics/btr610

Systems biology

Advance Access publication November 4, 2011

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

Stefan R. Maetschke[1], Martin Simonsen[2,3], Melissa J. Davis[1,4] and Mark A. Ragan[1,2,*]

[1]Institute for Molecular Bioscience, [2]Australian Research Council Centre of Excellence in Bioinformatics, The University of Queensland, Brisbane QLD 4072, Australia, [3]Bioinformatics Research Centre, Aarhus University, Aarhus 8000, Denmark and [4]Queensland Facility for Advanced Bioinformatics, The University of Queensland, Brisbane QLD 4072, Australia

Associate Editor: Burkhard Rost

## ABSTRACT

**Motivation:** Protein–protein interactions (PPIs) are pivotal for many biological processes and similarity in Gene Ontology (GO) annotation has been found to be one of the strongest indicators for PPI. Most GO-driven algorithms for PPI inference combine machine learning and semantic similarity techniques. We introduce the concept of *inducers* as a method to integrate both approaches more effectively, leading to superior prediction accuracies.

**Results:** An inducer (ULCA) in combination with a Random Forest classifier compares favorably to several sequence-based methods, semantic similarity measures and multi-kernel approaches. On a newly created set of high-quality interaction data, the proposed method achieves high cross-species prediction accuracies (Area under the ROC curve ≤ 0.88), rendering it a valuable companion to sequence-based methods.

**Availability:** Software and datasets are available at http://bioinformatics.org.au/go2ppi/

**Contact:** m.ragan@uq.edu.au

Received on June 16, 2011; revised on October 12, 2011; accepted on October 28, 2011

## 1 INTRODUCTION

A major challenge in systems biology is the accurate mapping of the interactome—the set of all protein-protein interactions (PPIs) within a cell. Understanding the interactome is essential in deciphering protein function and cell behavior (Eisenberg *et al.*, 2000).

Large-scale interaction maps have been experimentally determined but are incomplete and show high error rates (von Mering *et al.*, 2002). Consequently, *in silico* methods have been developed to infer PPIs from various sources of information including sequence features such as n-gram composition, phylogenetic relationships, e.g. interologs, and Gene Ontology (GO) annotation. Lu *et al.* (2005) provide an excellent overview of features indicative for PPIs.

Several studies have recognized similarity in GO annotation as one of the strongest predictors for protein interaction (Lin *et al.*, 2004; Miller *et al.*, 2005; Patil and Nakamura, 2005). GO annotation-driven interaction inference is based on the observation that proteins localized to the same cellular compartment are more likely to interact than are proteins that reside in spatially distant compartments (Shin *et al.*, 2009). Similarly, proteins that share a common biological process or molecular function have been found to be predictive for PPI (Qi *et al.*, 2006).

GO is organized as a graph with terms as nodes and edges describing relationships (Ashburner *et al.*, 2000). The literature distinguishes two fundamentally different approaches to exploit GO annotation for the prediction of protein interactions: (i) the *semantic similarity measure* (SSM) approach and (ii) the *machine learning* (ML) approach. SSMs are unsupervised methods that operate on a hierarchical graph (DAG) of relationships to measure similarities between entities. For instance, similarity between GO terms can be expressed by the shortest path between terms within the GO DAG. ML approaches for PPI inference, on the other hand, are supervised and the GO annotation of a protein pair is encoded by a feature vector, indicating assigned or shared GO terms, which is subsequently evaluated by an ML classifier.

The ML approach has the advantage that advanced classification algorithms such as Support Vector Machines or Random Forests (Breiman, 2001) can be applied, which are usually more accurate predictors than unsupervised methods. However, the traditional encoding of shared GO terms within the feature vector ignores the relationships between GO terms and therefore hides important information from the classifier. The SSM approach, on the other hand, explicitly exploits topological relationships between GO terms but is typically limited to comparatively simple predictors, e.g. term probabilities. These competing properties of the SSM and the ML approach have been recognized by many authors, and hybrid methods have been developed, in which the output of a SSM method serves as input (among others) to an ML classifier. However, the hybrid approach still does not allow the classifier to exploit term relationships directly. We hypothesized that a better integration of SSMs with ML algorithms may lead to improved prediction accuracies and we propose the concept of *inducers* to achieve such integration. In the following, we first discuss the various hybrid ML approaches and SSMs, before explaining inducers in Section 2.2.

Patil and Nakamura (2005) trained a Naive Bayes classifier using input features such as shared GO terms, PFAM domains and sequence similarity to infer PPI. Shared GO terms were also exploited by He *et al.* (2009) to identify novel interactions in human. Rhodes *et al.* (2005) used interolog data, expression profiles and the likelihood of the least-frequent shared GO terms for PPI inference. Jansen *et al.* (2003) employed a Bayesian network and

[*]To whom correspondence should be addressed.

GO annotation, Munich Information Center for Protein Sequences (MIPS), essentiality and mRNA co-expression as features to predict interactions in yeast. Lin *et al.* (2004) and Lu *et al.* (2005) assessed the importance of different features and found MIPS and GO functional similarity to be the most indicative features of PPI. De Bodt *et al.* (2009) exploited sequence orthology, expression data and the maximum depth of the common ancestors of the GO annotations for PPI inference. Miller *et al.* (2005) and Ben-Hur and Noble (2005) both employed a SVM and, among various feature kernels, GO similarity measured by the maximum of the log likelihoods of common ancestors terms. Recent work by Qiu and Noble (2008) also uses a multi-kernel SVM but applies it to the prediction of co-complexed protein pairs.

In contrast to supervised machine learning methods, SSMs are unsupervised techniques to measure term similarities over taxonomies. The most common, *node-based* techniques by Resnik (1995), Jiang and Conrath (1997), Lin (1998) and many others utilize the information content of the most informative common ancestor to measure term similarity. *Edge-based* methods, on the other hand, exploit the lengths of the paths between terms (Pesquita *et al.*, 2009). Only a few authors have utilized SSMs to infer PPIs. Wu *et al.* (2006) defined a similarity measure that is composed of three components, such as overlap of induced term subgraphs, term generality and term distances to lowest common ancestors. Guo *et al.* (2006) compared five SSMs to predict human PPIs and found Resnik's measure (Resnik, 1995) to perform best. Finally, Jain and Bader (2010) introduced a novel SSM that attempts to compensate for the unequal depths of different branches of the GO DAG. In an evaluation on a yeast PPI dataset, the prediction performance (Area under the ROC curve) was on par with Resnik's similarity metric. Applications of SSMs to other prediction problems are referenced in Section 11 of the Supplementary Material.

## 2 METHODS

In this section, we first describe GO, then present *inducers*, and close with a description of the datasets employed for their evaluation.

### 2.1 GO

GO is a hierarchically organized, controlled vocabulary to characterize gene products (Ashburner *et al.*, 2000). It is composed of the three subontologies: biological process (BP), cellular component (CC), molecular function (MF). Each subontology is represented by a rooted DAG with nodes referring to GO terms and edges defining relationships between terms. The inducers, described in the following, use only *is_a* and *part_of* edges and ignore all other relationships.

### 2.2 Term inducers

Term inducers define sets of GO terms that are induced within the DAG by the GO annotation of protein pairs. Inducers are motivated by the assumption that an induced term set is richer in information, and can be a more accurate predictor of protein interaction, than the original annotation. For instance, the terms along the shortest path between two GO terms within the DAG may be better indicators of protein interaction than the two GO terms alone. In the following, we first establish a general framework before describing individual inducers in detail.

Let a GO DAG be given as a graph $G(T,R)$ with term set $T$, where terms represent nodes within the graph, and a relationship or edge set $R \subset T\times T$. Furthermore, let two proteins $p_1$ and $p_2$ be annotated by term sets $S_1 \subset T$ and $S_2 \subset T$, respectively. From the two original term sets $S_1$ and $S_2$ a new, induced term set $S$ is constructed by applying an *inducer* $I$ as follows:

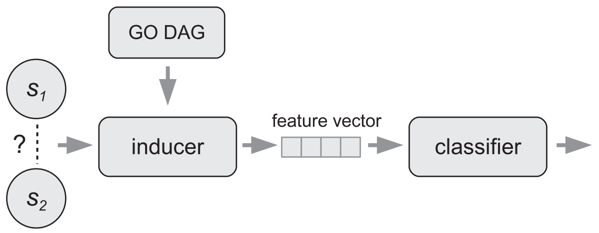

**Fig. 1.** Architecture. An inducer operates over the GO DAG and maps two term sets $S_1$ and $S_2$, assigned to two proteins, onto a feature vector that serves as input to an ML classifier. Section 10 within Supplementary Material provides a pseudocode description of the inducer approach.

$$\begin{aligned} I: T\times T &\rightarrow T \\ (S_1,S_2) &\mapsto S \end{aligned} \tag{1}$$

The induced term set is subsequently projected onto a feature vector $v$ by associating each GO term $t \in T$ with an arbitrary but unique index $idx(t) \in \{1,\ldots,|T|\}$ and setting $v_{idx(t)} = 1$ if $t \in S$, or 0 otherwise. The resulting binary feature vector is typically high-dimensional ($|T|$ is large) but sparse ($S << T$) and serves as input to a standard ML classifier, either to predict whether two proteins are interacting or to train the classifier. Figure 1 depicts the architecture of the system.

While inducers can generate term sets in various ways, we are particularly interested in mappings that express the relationships between terms as given by the ontology graph—similar to the way SSMs exploit topological information. We distinguish three classes of inducers. *Basic inducers* ignore term relationships, serve as controls and represent the traditional ML approach. *Ancestral inducers* are based on ancestor terms derived from a set of protein annotations and resemble node-based SSMs. *Shortest path-inducers*, which include terms along the shortest path or paths between two term sets, are similar to edge-based SSMs.

*2.2.1 Basic inducers* The *AL* (All Labels) inducer computes the union of the term sets $S_1$ and $S_2$, which annotate protein $p_1$ and $p_2$, respectively:

$$\mathrm{AL}(S_1,S_2) = S_1 \cup S_2 \tag{2}$$

Equally simple is the *AC* (All Common terms) inducer, which is defined as the intersection of the two term sets $S_1$ and $S_2$:

$$\mathrm{AC}(S_1,S_2) = S_1 \cap S_2 \tag{3}$$

Note that basic inducers do not induce an enriched term set. They represent the common approach, in which the annotation of a protein pair is described by an indicator vector that encodes either all assigned GO terms (*AL*) or the GO terms shared (*AC*) by the two proteins.

*2.2.2 Ancestral inducer* Common ancestors of terms within the GO DAG are generalized concepts (subsumers) of their descendants, and form the basis of many SSMs. Ancestral inducers aim to take advantage of this information by generating sets that include ancestral terms of the original GO protein annotation.

Let the set of parent terms of an individual GO term $t$ be denoted as $P(t)$. Also, let the set of parent terms $P(S)$ over a set of terms $S$ be defined as the union of the corresponding ancestor sets: $P(S) = \bigcup_{t\in S} P(t)$. With this definition, the set of ancestor terms can be written as $A(S) = S \cup A(P(S))$, with $A(\emptyset) = \emptyset$. Note that this recursive definition of ancestors includes the original term set $S$ and all its ancestral terms up to the root term $r$.

We furthermore define the depth $D(t)$ of a term within the GO DAG as the longest path from the root $r$ to term $t$, which can easily be calculated as $D(t) = \max\{D(u) | u \in P(t)\} + 1$, with $D(r) = 0$. Equipped with the definitions above we now describe several ancestral inducers.

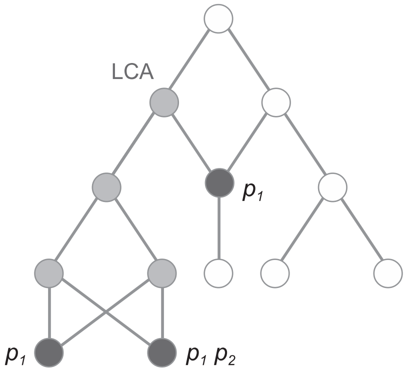

**Fig. 2.** Example of the *ULCA* inducer. Proteins $p_1$ and $p_2$ are annotated by the black nodes (terms) within the GO DAG. Dark gray nodes are induced by the *ULCA* inducer up to the lowest common ancestor (*LCA*).

The *AA* (All Ancestors) inducer computes the union of all ancestors of two term sets $S_1$ and $S_2$:

$$\mathrm{AA}(S_1,S_2)=A(S_1)\cup A(S_2) \tag{4}$$

It is an extension of the *AL* inducer that includes the ancestral terms.

Similarly, the *ACA* (All Common Ancestors) inducer extends the *AC* inducer by computing the intersection of the ancestor sets:

$$\mathrm{ACA}(S_1,S_2)=A(S_1)\cap A(S_2) \tag{5}$$

The lowest common ancestor of two term sets is the shared ancestor with the largest depth $D(\cdot)$. The *OLCA* (Only Lowest Common Ancestors) inducer generates this set. Note that there can be multiple lowest common ancestor terms (LCATs), all with the same depth.

$$\mathrm{OLCA}(S_1,S_2)=\operatorname*{arg\,max}_t \{D(t)|t\in ACA(S_1,S_2)\} \tag{6}$$

The *LCA* (Lowest Common Ancestors) inducer extends the *OLCA* set with the original term sets $S_1$ and $S_2$:

$$\mathrm{LCA}(S_1,S_2)=S_1\cup S_2\cup OLCA(S_1,S_2) \tag{7}$$

However, the *LCA* set still does not contain terms along the paths from the original term sets $S_1$ and $S_2$ to the lowest common ancestors. The *ULCA* (Up to Lowest Common Ancestors, see Fig. 2) extends the *LCA* set by incorporating those terms:

$$\mathrm{ULCA}(S_1,S_2)=\{t|t\in AA(S_1,S_2)\wedge D(t)>=d_{lca}\}, \tag{8}$$

with $d_{lca}$ being the depth of the lowest common ancestors.

To assess the importance of the lowest common ancestor for PPI prediction, we also compute the *WLCA* (Without Lowest Common Ancestors) set, which is the *ULCA* set without the lowest common ancestors or equivalently, the *AA* set without the *ACA* set:

$$\mathrm{WLCA}(S_1,S_2)=AA(S_1,S_2)\setminus ACA(S_1,S_2) \tag{9}$$

*2.2.3 Shortest-path inducers* The length of the shortest path between two terms within the GO DAG is a measure for their relatedness. Shortest-path inducers strive to take advantage of this property by generating sets that include the terms along the shortest paths between two term sets. The shortest path within a graph is, however, defined only between two individual terms but not between sets of terms. A natural generalization for term sets would be the *k-minimum spanning tree* (k-MST) but is known to be NP-hard—although there are fast approximations (Garg, 1996). We employ a very simple heuristic to approximate the k-MST for proteins annotated with more than one GO term.

For the following definitions, we treat the edges between nodes within the GO DAG as undirected and of constant weight. Let $sp(t_1,t_2)$ be the set of terms along a shortest path between $t_1$ and $t_2$. We define the shortest path set $sp(t,S)$ between a single term $t$ and a term set $S$, based on the shortest shortest path set between $t$ and any term in $S$:

$$sp(t,S)=\operatorname*{arg\,min}_{sp(t,u)} \{|sp(t,u)| \mid u\in S\}, \tag{10}$$

with $|sp(t,u)|$ being the number of terms along a shortest path between $t$ and $u$. Note that $sp(t,S)$ can be computed efficiently by terminating the algorithm as soon as a term in $S$ is encountered.

To compute the shortest path set between two term sets $S_1$ and $S_2$, we first select two arbitrary terms $t_1\in S_1$ and $t_2\in S_2$. In the second step, a remainder set $R$ is constructed, which is the union of the original term sets $S_1$, $S_2$ and the shortest path set $sp(t_1,t_2)$ but without $t_1$ and $t_2$:

$$R=(S_1\cup S_2\cup sp(t_1,t_2))\setminus\{t_1,t_2\} \tag{11}$$

Finally, the union of the shortest path sets between any term $u\in R$ and $R$ without $u$ is constructed:

$$SPS(S_1,S_2)=\bigcup_{u\in R} sp(u,R\setminus u) \tag{12}$$

We call $SPS(S_1,S_2)$ the *Single Shortest Path* inducer because only one shortest path is taken into account by $sp(t,S)$. Within a DAG, however, there can be many shortest paths and we analogously define an *All Shortest Paths* inducer $SPA(S_1,S_2)$:

$$SPA(S_1,S_2)=\bigcup_{u\in R} sp'(u,R\setminus u), \tag{13}$$

with $sp'(t,S)$ being the union of all shortest path sets between a term $t$ and a term set $S$.

## 2.3 Data

We used data from Uniprot, GO, the STRING database and other authors to evaluate the prediction accuracies of the inducers presented above.

*2.3.1 Uniprot* The Uniprot (Bairoch *et al.*, 2005) database (version 18, 2011-06-28) was downloaded and stored within an XML database (Berkeley DB). GO annotations for the proteins within the interaction datasets were extracted from this database. Annotations derived from physical (IPI) or genetic (IGI) interactions were excluded to avoid giving the inducer approach an advantage over other methods evaluated [(Rogers *et al.*, 2009), and Section 7 of Supplementary Material].

*2.3.2 GO* The GO database (Revision 1.1551) without cross-products, inter-ontology links and has_part relationships was downloaded. We also filtered out all regulatory relationships, and maintain only the *is_a* and *part_of* relationships.

*2.3.3 Benchmark datasets* To create a comprehensive benchmark set with high-quality interactions, seven PPI networks (Table 1) were extracted from the STRING database (Jensen *et al.*, 2009). We downloaded the database (version 9.0) and filtered for experimentally validated interactions tagged as *binding* with a confidence score ≥0.9 [see von Mering *et al.* (2005) for a description of the scoring scheme]. Protein identifiers were mapped to Uniprot IDs and redundant interactions, self-links and links without a *binding* tag were removed. The upper half of Table 1 lists the generated networks with their protein and interaction numbers.

Despite filtering for direct *binding*, indirect interactions are not entirely excluded. Many experimental methods, including the two most common techniques, two-hybrid and co-immunoprecipitation, do not distinguish between *direct interactions*, where two proteins are physically in contact, and *indirect interactions* e.g. mediated by other members of the complex. While reported interactions tend to be direct this cannot be guaranteed in all cases.

Negative samples (non-interactions) for cross-validation were generated by randomly and uniformly sampling from the set of all proteins pairs that

**Table 1.** Benchmark datasets

| Label | Species | Source | Proteins | Interactions |
|---|---|---|---|---|
| *SC* | *Saccharomyces cerevisiae* | STRING | 3291 | 15238 |
| *HS* | *Homo sapiens* | STRING | 3296 | 3490 |
| *EC* | *Escherichia coli* | STRING | 589 | 1167 |
| *SP* | *Schizosaccharomyces pombe* | STRING | 904 | 742 |
| *AT* | *Arabidopsis thaliana* | STRING | 756 | 541 |
| *MM* | *Mus musculus* | STRING | 1088 | 500 |
| *DM* | *Drosophila melanogaster* | STRING | 658 | 321 |
| *YP* | *Saccharomyces cerevisia* | Y. Park | 2152 | 3844 |
| *AB* | *Saccharomyces cerevisia* | A. Ben-Hur | 4233 | 10517 |
| *AB-rel* | *Saccharomyces cerevisia* | A. Ben-Hur | 736 | 750 |

Benchmark datasets used for evaluations, with their numbers of proteins and interactions. The first seven datasets were extracted from the STRING database. The remaining three datasets were generated by the authors specified.

are not reported to interact within the STRING database. Jansen *et al.* (2003) filtered for proteins in separate subcellular compartments, but Ben-Hur and Noble (2005, 2006) have shown that this introduces a bias, rendering the classification task significantly easier, and protein localization was therefore not taken into account. A possible alternative for high-quality negative data, the *Negatome* (Smialowski *et al.*, 2010)—a database of non-interacting protein pairs—did not contain enough samples for our purposes.

For further evaluations and comparison with other methods, we downloaded yeast datasets created by Park (2009) and Ben-Hur and Noble (2005) (for details see Section 2 of the Supplementary Material). The *YP* dataset is marginally different from the original set because four protein identifiers could not be mapped to Uniprot accession numbers (required by our method), but remains very similar with 3844 compared with 3867 interactions. Ben-Hur and Noble (2005) extracted two PPI datasets from the BIND database: the larger *AB* dataset and the smaller *AB-rel* dataset with reliable interactions only for which we successfully mapped all identifiers to Uniprot accession numbers.

## 3 RESULTS

If not stated otherwise, for all evaluations 10-fold cross-validation was performed by splitting the dataset into 10 parts, training on 9 parts, testing on the remaining part and repeating this procedure for each part. Sample sets were balanced with an equal number of positives and negatives. Prediction accuracy was measured by the Area under the ROC curve (AUC) or the AUC up to the first 50 false positives ($\text{AUC}_{50}$). We calculate $AUC=\frac{1}{2}\sum_{k=1}^{n}(X_k-X_{k-1})(Y_k+Y_{k-1})$, where $X_k$ is the false positive rate and $Y_k$ is the true positive rate for the $k$-th output in the ranked list of predicted confidence or similarity scores generated by the classifier or SSM. To reduce computation time, ontology terms that are obsolete or unused were removed from the DAG, provided they were not part of the sub-DAG induced by used terms.

First, we evaluated the prediction accuracy of the inducers introduced in Section 2.2. As described there, induced term sets were projected onto a feature vector and processed by a classifier. Naive Bayes was chosen as the underlying classifier, since no optimization of parameter settings is required and training is fast. Performance was measured on the *SC* (yeast) dataset, due its large size and high-quality interactions (see Section 3 of Supplementary Material for results on other datasets). Figure 3 shows the prediction accuracies of all inducers for the three ontologies.

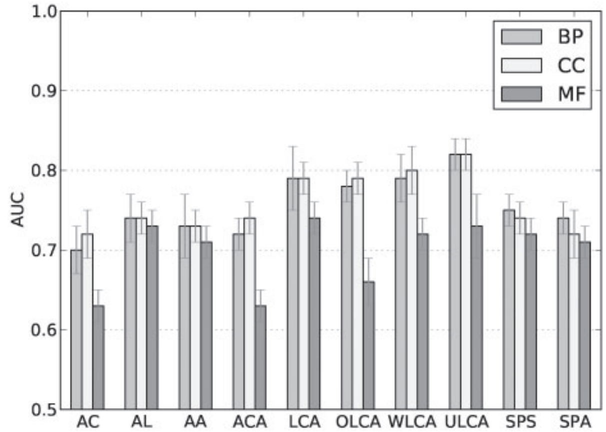

**Fig. 3.** Prediction accuracies (AUC) of different inducers with a Naive Bayes classifier for the three ontologies (BP, CC, MF) on *SC* dataset. Error bars show SD.

Strikingly, all inducers that compute common ancestors (·*CA*) show higher prediction accuracies than shortest path (*SPS*, *SPA*) and basic inducers (*AC*, *AL*), with the *ULCA* method being the top performer. But the results also reveal that it is not the lowest common ancestor terms (LCATs) alone that are responsible for the superior performance. First, the accuracy of the *OLCA* inducer (which generates LCATs only) is lower than that of the *ULCA* method. Second, terms along the paths from the original term sets up to, but without, the LCATs are not sufficient for top performance either, as the lower accuracy for the *WLCA* inducer (which excludes LCATs) indicates. Finally, the combination of the original term sets and LCATs as generated by the *LCA* inducer is still inferior to the *ULCA* method.

The prediction accuracy of the shortest-path methods (*SPS*, *SPA*) is slightly better than that of the basic inducer *AC*, comparable to *AL* but inferior to all common ancestor methods (·*CA*). Note that within a tree topology and for proteins annotated with single GO terms, the shortest path inducer and the *ULCA* inducer would be essentially equivalent. In agreement with Jain and Bader (2010), we find that the predictive power of the BP and CC ontologies is similar, with a slight advantage for the BP ontology, while the predictive power of the MF ontology is considerably lower.

As stated previously, shortest path and common ancestor inducers resemble many SSMs. The main difference between inducers and SSMs is that the former map induced terms on a (high-dimensional) feature vector to be evaluated by a supervised classifier, while the latter compute a single similarity value using an unsupervised approach. Figure 4 compares the prediction accuracies of the *ULCA* inducer on the *SC* dataset with SSMs by Jiang and Conrath (1997) (*JIA*), Resnik (1995) (*RES*), Lin (1998) (*LIN*), Pesquita *et al.* (2008) (*PES*), Gentleman (http://www.bioconductor.org/packages/release/bioc/html/GOstats.html) (*GEN*) and Schlicker *et al.* (2006) (*SCH*). *SP* is the length of the shortest path as generated by the *SPS* inducer. For all SSMs, the accuracies achieved with the *best-match-average* (BMA) aggregation strategy are shown. In agreement with Pesquita *et al.* (2008), *maximum* (MAX) and *average* (AVG) strategies were found to be inferior (see Section 4 of Supplementary Material). Qi *et al.* (2006) found Random Forests (RF) superior to NB classifiers, and to achieve peak accuracies we combined the *ULCA* inducer

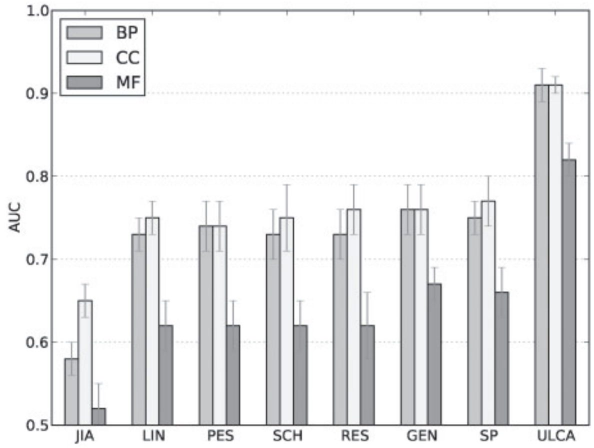

**Fig. 4.** Comparison of the *ULCA* inducer (last three bars), using an RF classifier, with semantic similarity measures (*JIA*, … , *SP*) on the *SC* dataset for all three ontologies (BP, CC, MF). Results are 10-fold cross-validated. Error bars show SD.

with an RF. To avoid possible bias, we optimized the RF parameters (#trees=200, depth=max, #features=200) on an independent dataset of *Chlamydia trachomatis* interactions, extracted from the BioGrid database and used nowhere else in this study.

The results show substantially higher AUCs of the inducer approach for all three ontologies compared with all SSMs. Within the suite of SSMs, the *GEN* and *SP* methods performed best, closely followed by other SSMs, with the exception of the *JIA* methods. Section 4 of Supplementary Material provides more detailed data.

In addition to SSMs, we furthermore compared the prediction performance of the *ULCA* inducer to state-of-the-art machine learning approaches. Table 2 lists the AUC and AUC$_{50}$ scores of five sequence-based methods, a hybrid approach and a GO-kernel in comparison to the *ULCA* inducer with an RF classifier. The inducer used an integration of the three ontologies to achieve the best performances (see Section 1 of Supplementary Material for details).

Note that methods $M1$-$M4$ and $C$ exploit sequence information only and use SVMs as classifiers. Method $M1$ by Martin *et al.* (2005) utilizes trigram signatures as features, $M2$ by Pitre *et al.* (2006) relies on the co-occurrence of subsequences, $M3$ by Shin *et al.* (2007) evaluates trigrams over a reduced amino acid alphabet and $M4$ by Guo *et al.* (2008) computes auto-correlation over seven physicochemical properties to predict interactions. Method $C$ by Park (2009) is a consensus over the methods $M1$-$M4$. The prediction accuracies of all five methods were evaluated by Park (2009) on the *YP* dataset.

Ben-Hur and Noble (2005) employed a SVM with a combination of various kernels derived from sequence data, GO annotation, network properties and interolog information to predict PPIs. Method *AK* indicates the case where all kernels were combined, while method *GOK* means that only the GO kernel was used. The latter measured the similarity between two proteins by computing the maximum log likelihood over the common ancestors of the GO term annotations.

The inducer approach achieves substantially higher AUCs than all sequence-based methods ($M1$-$M4$), including the consensus method ($C$). In comparison to Ben-Hur's multi-kernel method (*AK*), the inducer approach is inferior on the set of reliable interactions *AB-rel* and slightly inferior to the GO kernel alone (*GOK*) on the same dataset. On the full dataset (*AB*), the inducer performs substantially better than the GO kernel (*GOK*). Section 2 of the Supplementary Material describes the comparisons above in more detail.

**Table 2.** Comparison with other prediction methods

| Author | Method | Dataset | Other<br>AUC / AUC$_{50}$ | ULCA<br>AUC / AUC$_{50}$ |
|---|---|---|---|---|
| Martin | $M1$ | *YP* | 0.83 / – | **0.90** / 0.57 |
| Pitre | $M2$ | *YP* | 0.79 / – | **0.90** / 0.57 |
| Shen | $M3$ | *YP* | 0.60 / – | **0.90** / 0.57 |
| Guo | $M4$ | *YP* | 0.75 / – | **0.90** / 0.57 |
| Park | $C$ | *YP* | 0.85 / – | **0.90** / 0.57 |
| Ben-Hur | *AK* | *AB-rel* | **0.98** / 0.58 | 0.93 / 0.58 |
| Ben-Hur | *GOK* | *AB-rel* | **0.95** / – | 0.93 / 0.58 |
| Ben-Hur | *GOK* | *AB* | 0.68 / – | **0.85** / 0.63 |

Comparison of the *ULCA* inducer using an RF classifier with other prediction methods. The column labeled 'ULCA' shows the AUC and AUC$_{50}$ accuracies of the inducer on the specified dataset, while the column labeled 'Other' shows the prediction accuracies of the method under comparison. Accuracies in bold indicate the best result.

Park (2009) also evaluated the cross-species accuracy by training a classifier on one species and predicting interactions for a different species. He found the cross-species accuracy of the sequence-based predictors to be low. For instance, a 4-fold cross-validation AUC of 0.85 was achieved on the yeast dataset, and an AUC of 0.90 on the human dataset. However, when trained on yeast and applied to human, the prediction accuracy dropped from 0.90 to 0.68.

Figure 5 displays the cross-species and self-test accuracies (AUC) of the *ULCA* inducer with an NB classifier on the BP ontology over seven species (results for other ontologies can be found in the Section 5 of the Supplementary Material). Dark colors indicate low prediction accuracies (AUCs) and the values along the diagonal are the AUCs of the self-test (training and test on the same dataset). Self-test accuracy is not very useful in general, but in case of an NB classifier (which has a low VC dimension) provides an optimistic but reasonable approximation of its maximum cross-validation accuracy.

Apart from the three smallest PPI datasets (*DM*, *AT*, *MM*) the cross-species accuracies of the GO-driven approach are good, with AUCs between 0.78 up to 0.88. The highest cross-species AUC of 0.88 was achieved for a predictor trained on the yeast (*SC*) and tested on *Escherichia coli* (*EC*). There is a clear trend for AUCs below the diagonal of the matrix to be higher than the corresponding AUCs above the diagonal, indicating that the prediction accuracy largely depends on the test dataset (target species) and only to a lesser degree on the training dataset (source species). Consequently, datasets with high self-test accuracies lead to high test accuracies for all training datasets, while datasets linked to low self-test accuracies (e.g. *MM* = 0.73) result in low cross-species accuracies. For example, a classifier trained on human (*HS*) and tested on *E. coli* (*EC*) achieves an AUC of 0.87, while a predictor trained on *E. coli* and applied to human reaches an AUC of only 0.80.

Error rates on the target species can be contributed to false interactions or false GO annotations. Jones *et al.* (2007) estimated error rates as high as 49% for *ISS* evidence codes, and 18% for GO

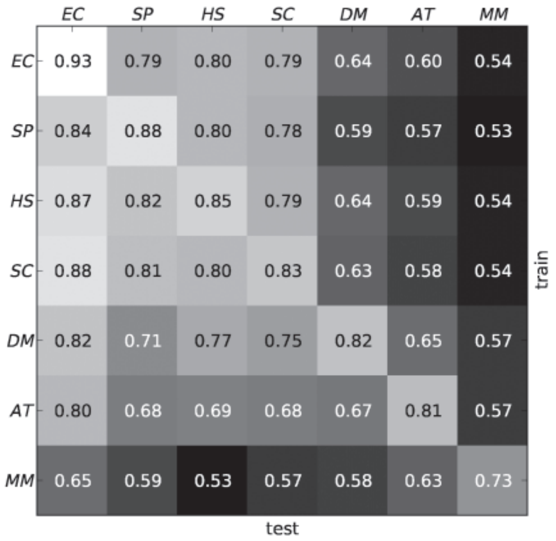

**Fig. 5.** Cross-species prediction accuracies (AUCs) of the *ULCA* inducer with a NB classifier using the BP ontology.

annotation from experimental evidence. High self-test accuracies of an NB classifier indicate highly consistent datasets, on which a classifier can perform well. Since GO annotation is essentially designed to be species-independent (Ashburner *et al.*, 2000), a model trained on one species can be transferred and successfully be applied to a different species.

## 4 DISCUSSION

Inducers encompass the traditional ML approach, in which only the assigned GO terms are mapped onto a feature vector (ignoring the GO topology), and also SSMs. For instance, Resnik's SSM is effectively an *OLCA* inducer with a simple classifier, utilizing information content as term weights.

Numerous variants of Resnik's method with different aggregation strategies have been developed, but for many applications do not achieve higher accuracies. 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. The *ULCA* inducer presented here significantly outperforms Resnik's method with BMA/MAX aggregation because the latter (and most other SSMs) are unsupervised methods, limited to simple estimators (e.g. term probabilities), while machine learning methods are able to model higher-order statistics. To confirm this, we replaced the classifier component of the *ULCA* inducer by the average or the maximum information content over the induced term set. The resulting prediction accuracies drop to the level of SSMs (Section 4 of Supplementary Material).

The inducer approach outperformed the sequence-based method evaluated (Table 2), but it is noteworthy that Patil and Nakamura (2005) found sequence similarity and GO annotation to have low correlation ($r = -0.13$), implying that a combination of GO and sequence-based methods could lead to further improvements in prediction accuracy. While the cross-species accuracy of sequence-based methods is low, they have the advantage of being applicable in cases where sequence data but no further annotation is available, e.g. newly sequenced genomes. On the other hand, annotation software such as *Blast2Go* (Götz *et al.*, 2008) can be employed to transfer GO terms to unannotated gene products. Since annotation transfer is largely based on sequence similarity as well, it remains an open question whether the superior accuracy of the inducer approach can be maintained in this case.

The computational burden of the inducer-based approach can be alleviated by using a reduced term set, a so-called *GO slim*. While slims lead to much shorter feature vectors and therefore shorter prediction times, they also reduce the prediction accuracy considerably (Section 6 of Supplementary Material). Furthermore, the mapping of GO annotations to slim terms via `map2slim.pl` (http://search.cpan.org/~cmungall/go-perl/scripts/map2slim) is algorithmically more complex than the *ULCA* inducer and also requires the selection of a suitable slim.

Here, we utilized a binary encoding to indicate the presence or absence of shared GO terms in an induced term set. However, the binary encoding cannot distinguish between the case in which a GO term is assigned to only one protein, and that in which it is assigned to none of the two proteins. This loss of information can be avoided by using a ternary encoding, e.g. 1 for GO terms assigned to one of the two proteins, 2 for GO terms assigned to both proteins and 0 otherwise. We found the ternary encoding not beneficial for the *ULCA* inducer, but it considerably improved the accuracy of the *AL* inducer (Section 8 of Supplementary Material).

It is known that PPI networks contain many more non-interactions (negatives) than interactions (positives). Qi *et al.* (2006) estimate a ratio of 600:1 but as classifiers generally cannot be trained on sample sets of this size, balanced datasets are used instead (Ben-Hur and Noble, 2005; Park, 2009; Qiu and Noble, 2008). Performance is frequently measured by the AUC to compare methods and by the $\mathrm{AUC}_{50}$ as a practically relevant measure for experimental validation of top-ranking predictions. We found the AUC stable for different ratios but the $\mathrm{AUC}_{50}$ highly volatile and less reliable (Section 9 of Supplementary Material).

## 5 CONCLUSION

In this work, we introduce the concept of *inducers* to better integrate ML methods and SSMs for the inference of PPIs. The inducer approach outperformed a suite of sequenced-based methods and SSMs, and achieved accuracies close to a multi-kernel method that exploits information beyond mere GO annotation.

On a high-quality yeast PPI dataset, a peak prediction accuracy of 0.91 (AUC) was achieved by the *ULCA* inducer in combination with a RF classifier. We also showed that the cross-species prediction accuracy of the inducer approach is high (AUC $\leq$ 0.88).

When comparing the three individual ontologies of the GO we found, in agreement with other studies (Qi *et al.*, 2006), similarities in biological process and cellular component annotation to be stronger indicators for protein interaction than similar molecular function annotation.

It is noteworthy that the proposed method is not specifically designed for the prediction of PPIs and can be applied to other domains described by ontological data. Considering the growing number of ontologies (Smith *et al.*, 2007) and their improving coverage, formality and integration (Pesquita *et al.*, 2009), we expect methods that can exploit ontological data to become increasingly important.

*Funding*: Australian Research Council Centre of Excellence in Bioinformatics, DP110103384 and CE034822 grants.

*Conflict of Interest*: none declared.

## REFERENCES

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **21**, 25–29.

Bairoch,A. *et al.* (2005) The Universal Protein Resource (UniProt). *Nucleic Acids Res.*, **33**, D154–D159.

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

Ben-Hur,A. and Noble,W.S. (2006) Choosing negative examples for the prediction of protein-protein interactions. *BMC Bioinformatics*, **7** (Suppl. 1), S2.

Breiman,L. (2001) Random Forests. *Mach. Learn.*, **45**, 5–32.

De Bodt,S. *et al.* (2009) Predicting protein-protein interactions in *Arabidopsis thaliana* through integration of orthology, gene ontology and co-expression. *BMC Genomics*, **10**, 288.

Eisenberg,D. and Marcotte,E.M. (2000) Protein function in the post-genomic era. *Nature*, **405**, 823–826.

Garg,N. (1996) A 3-approximation for the minimum tree spanning $k$ vertices. In *Proceedings of the IEEE Foundations of Computer Science*, IEEE Comput. Soc. Press, pp. 302—309.

Götz,S. *et al.* (2008) High-throughput functional annotation and data mining with the Blast2GO suite. *Nucleic Acids Res.*, **36**, 3420—3435.

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.

Guo,X. *et al.* (2006) Assessing semantic similarity measures for the characterization of human regulatory pathways. *Bioinformatics*, **22**, 967–973.

He,M. *et al.* (2009) PPI Finder: a mining tool for human protein-protein interactions. *PLoS ONE*, **2**, e4554.

Jain,S. and Bader,G.D. (2010) An improved method for scoring protein-protein interactions using semantic similarity within the gene ontology. *BMC Bioinformatics*, **11**, 562.

Jansen,R. *et al.* (2003) A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, **302**, 449–453.

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. In *Proceedings of 10th International Conference on Research in Computational Linguistics*, Taipei, pp. 19–33.

Jones,C.E. *et al.* (2007) Estimating the annotation error rate of curated GO database sequence annotations. *BMC Bioinformatics*, **8**, 170.

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

Lin,N. *et al.* (2004) Information assessment on predicting protein-protein interactions. *BMC Bioinformatics*, **5**, 154.

Lu,L.J. *et al.* (2005) Assessing the limits of genomic data integration for predicting protein networks. *Genome Res.*, **15**, 945–953.

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

Miller,J.P. *et al.* (2005) Large-scale identification of yeast integral membrane protein interactions. *Proc. Natl Acad. Sci. USA*, **102**, 12123–12128.

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

Patil,A. and Nakamura,H. (2005) Filtering high-throughput protein-protein interaction data using a combination of genomic features. *BMC Bioinformatics*, **6**, 100.

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.

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. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 448—453.

Rhodes,D.R. *et al.* (2005) Probabilistic model of the human protein-protein interaction network. *Nat. Biotechnol.*, **23**, 951–959.

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.

Shin,C.J. *et al.* (2009) Protein-protein interaction as a predictor of subcellular location. *BMC Syst. Biol.*, **3**, 28.

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

Smialowski,P. *et al.* (2010) The Negatome database: a reference set of non-interacting protein pairs. *Nucleic Acids Res.*, **38**, D540–D544.

Smith,B. *et al.* (2007) The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.*, **25**, 1251–1255.

von Mering,C. *et al.* (2002) Comparative assessment of large-scale datasets of protein-protein interactions. *Nature*, **417**, 399–403.

von Mering,C. *et al.* (2005) STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Res.*, **33**, D433–D437.

Wu,X. *et al.* (2006) Prediction of yeast protein-protein interaction network: insights from the Gene Ontology and annotations. *Nucleic Acids Res.*, **34**, 2137–2150.