Integrating Information in Biological Ontologies and Molecular Networks to Infer Novel Terms

Le Li[1] **and Kevin Y. Yip**[1,2,3,4*]

[1]Department of Computer Science and Engineering,
[2]Hong Kong Bioinformatics Centre,
[3]CUHK-BGI Innovation Institute of Trans-omics, and
[4]Hong Kong Institute of Diabetes and Obesity, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong
[*]kevinyip@cse.cuhk.edu.hk

ABSTRACT

Currently most terms and term-term relationships in Gene Ontology (GO) are defined manually, which creates cost, consistency and completeness issues. Recent studies have demonstrated the feasibility of inferring GO automatically from biological networks, which represents an important complementary approach to GO construction. These methods (NeXO and CliXO) are unsupervised, which means 1) they cannot use the information contained in existing GO, 2) the way they integrate biological networks may not optimize the accuracy, and 3) they are not customized to infer the three different sub-ontologies of GO. Here we present a semi-supervised method called Unicorn that extends these previous methods to tackle the three problems. Unicorn uses a sub-tree of an existing GO sub-ontology as training part to learn parameters in integrating multiple networks. Cross-validation results show that Unicorn reliably inferred the left-out parts of each specific GO sub-ontology. In addition, by training Unicorn with an old version of GO together with biological networks, it successfully re-discovered some terms and term-term relationships present only in a new version of GO. Unicorn also successfully inferred some novel terms that were not contained in GO but have biological meanings well-supported by the literature.
Availability: Source code of Unicorn in Matlab is available on our supplementary Web site at http://yiplab.cse.cuhk.edu.hk/unicorn/.

Introduction

Gene Ontology (GO)[1] is the most widely-used biological ontology. It systematically summarizes current knowledge of gene products and their relationships across a wide range of species. GO contains standardized terms in three sub-categories, namely biological processes (BP), cellular components (CC), and molecular functions (MF). These terms are organized hierarchically in directed acyclic graphs (DAGs), which are tree-like structures that allow a node to have multiple parents, corresponding to the specialization of a term from multiple general terms. A gene can be annotated by multiple GO terms. If a gene is annotated by a GO term, it is also annotated by all its ancestral terms automatically. GO has been extensively used in various applications, such as assessing functional similarity of genes[2–4], predicting gene functions[5–7], and interpreting biological data[8–10].

Most of the term-term relationships in GO are defined manually, assisted by text-mining of the literature. There are several limitations to this manual curation process. First, with the rapid expansion of biological knowledge, both the number and complexity of biological publications have become difficult to handle even with the help of text-mining. Second, the same biological concept can be described in different ways in different publications, which creates a challenge for different curators to represent the concept in a consistent manner. Finally, there is considerably more research on a subset of well-studied genes and their relationships, leading to unbalanced levels of detail in different parts of GO.

One complementary approach to GO construction is to infer terms and term-term relationships automatically from biological networks. This approach is attractive given the large amount and variety of network data already available, and the relative low cost of creating new networks and expanding existing ones using high-throughput experimental methods. The feasibility of inferring GO automatically from biological networks has been recently demonstrated[11]. In this study, a method called Network-eXtracted Ontology (NeXO) was proposed to cluster genes hierarchically based on their connections in the networks and subsequently transform the resulting clustering tree into a DAG. By using four types of molecular networks as input, NeXO was able to recover around 40% of the terms in GO based on an alignment of the terms in the NeXO and GO DAGs. Later, another method called Clique eXtracted Ontology (CliXO) was proposed to further improve the accuracy of the automatically constructed ontology[12]. This method identifies cliques of different sizes in an integrated biological network by progressively loosening the stringency for an edge to be drawn between two genes in the networks. Each identified clique forms a term that

annotates the composing genes, and a new term becomes a parent of an existing term if the clique corresponding to the new term is a superset of the existing term. A major novelty of CliXO was its ability to use quantitative measures in the biological networks, such as the confidence score of the existence of an edge, in the ontology inference process. The best DAG constructed by CliXO achieved about 40% in both precision and recall when compared to the actual GO DAG.

These two studies clearly show that existing biological networks, though incomplete and noisy, contain useful information that can be used to automatically infer GO with a reasonable accuracy. On the other hand, one limitation of both NeXO and CliXO is that they infer DAGs purely based on the input network (either a single biological network or a network integrated from multiple biological networks), which implies that 1) they are unsupervised methods that cannot make use of the information contained in the existing GO, 2) the way of integrating the biological networks is not guaranteed to optimize the accuracy of ontology construction, and 3) given a fixed set of input networks, both methods cannot infer different DAGs specifically for the three different sub-ontologies of GO.

Here we extend these previous works by describing a semi-supervised method called Unicorn (Unification of Discordant Networks), which integrates multiple biological networks in a way tailored for inferring a particular sub-ontology of GO. The key idea is that each existing GO sub-ontology contains parts that are highly accurate and complete, which can be used as a training set to find out the best way to integrate biological networks for inferring the whole sub-ontology. The resulting DAG inferred by Unicorn is then expected to supplement parts of the sub-ontology not as well constructed. By using training data from a particular sub-ontology, the way to integrate the biological networks is specific to this sub-ontology. Unicorn is semi-supervised because it considers both the training part of GO and the natural distribution of edge weights in the biological networks during data processing and integration.

One major challenge of integrating different biological networks is their different distributions of edge weights and semantics, such as expression correlations in a co-expression network and similarity scores in a functional network. Unicorn uses a novel discretization procedure to turn edge weights into nominal values such that they are highly correlated with the gene-gene similarity values based on the training set of the GO sub-ontology. The resulting discretized values in the different networks can then be integrated easily.

We tested Unicorn by 1) evaluating its accuracy on left-out parts of the GO sub-ontologies not involved in training, 2) constructing a DAG by using an old version of GO for training, and comparing the newly discovered terms with a new version of GO, and 3) surveying the literature for supports of novel terms discovered by Unicorn that are not in existing GO. These tests showed that Unicorn can construct specific GO sub-ontologies accurately and identify biologically meaningful new terms.

One recent study has also engaged multiple biological networks to infer gene ontology in a supervised manner[13]. Our work is fundamentally different from it in that the method in this study does not attempt to find the optimal way to integrate networks, that it assumes edge weights in different networks can be combined in a straightforward manner, that it does not discover novel terms, and that it cannot be evaluated using a training-testing procedure. Another recent study has attempted to extend existing GO by using biological networks[14], but the method cannot infer GO automatically. Finally, there is a method that groups related terms based on genes that they annotate[15], which can also discover term-term relationships as we do, but does not aim at inferring novel terms or constructing the ontology.

In the followings we describe the details of Unicorn and the empirical tests we have performed using data from *S. cerevisiae*.

Methods

The overall pipeline of Unicorn for integrating multiple biological networks and inferring a GO sub-ontology is illustrated in Fig 1. There are seven main steps, the details of which will be given in the corresponding sections below. Step 1: A sub-tree of a GO sub-ontology is selected as the training part. Step 2: For every pair of genes both annotated by a term in the training part (a "training gene pair"), their similarity in the sub-ontology is computed based on a simplified version[12] of the Resnik semantic similarity measure[28]. Step 3: For each biological network, the edges are filtered based on the ontological similarity values of the training gene pairs, with a goal of removing edges irrelevant to the GO sub-ontology. Step 4: The weights of the retained edges are discretized in a concerted manner such that the different networks can be easily integrated. Step 5: The discretized networks are integrated to maximize the correlation between the discretized edge weights in the integrated network and the ontological similarity values of the training gene pairs. Integrating networks in this way is expected to make the resulting edge weights of the gene pairs not in the training set (the "left-out gene pairs") useful for inferring their ontological relationships. Step 6: The CliXO method[12] is run on the integrated network to infer a DAG based on all the genes. Step 7: The terms in the inferred DAG and the actual DAG of the GO sub-ontology are aligned[11] to evaluate the similarity of the two DAGs based on the left-out gene pairs, and to discover novel terms in the inferred DAG.

The first 5 steps are novel to Unicorn, at the end of which a single integrated biological network is created and supplied as the standard input to CliXO. The semi-supervised nature of Unicorn allows it to make good use of the information in existing GO as compared to CliXO (Table **??**).

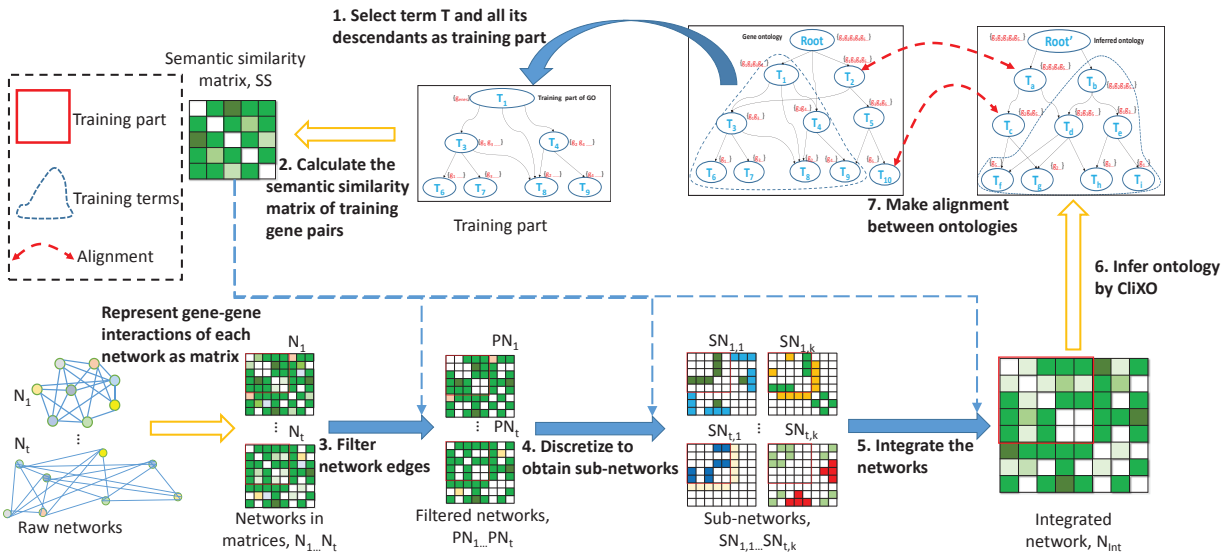


Figure 1. The overall pipeline of Unicorn for integrating multiple heterogeneous networks and inferring a GO sub-ontology. In the matrix representation of each network, a darker color indicates a larger value. Diagonal entries are ignored by CliXO and are always set to 0. The colors of the matrix entries after Step 4 are the new edge weights after discretization (explained in Fig S4).

Selection of training part (Step 1)

There are two key considerations when choosing the training part from a GO sub-ontology, namely 1) the size of it should be big enough to capture sufficient information for guiding the network filtering, discretization and integration steps, and 2) the left-out part should not be too fragmented for otherwise it would be difficult to have terms in the inferred ontology that are not directly due to the training part, thereby making it hard to evaluate the effectiveness of Unicorn objectively. Consequently, for each GO sub-ontology, we select every sub-tree with a root between the 2nd and 5th levels as a training part, and use each of them to infer a DAG in turn.

Filtering edges in biological networks (Step 2 and 3)

Existing biological networks contain a lot of interactions discovered by high-throughput experiments, including some low-confidence interactions that could be false positives. There are also interactions irrelevant to the target GO sub-ontology. To prevent these interactions from misleading the ontology inference process, previous studies have filtered them using arbitrary edge weight thresholds or requiring each network to have the same final number of interactions[11,12]. In our semi-supervised pipeline, we instead use the training part to determine an appropriate threshold for each network individually.

We first compute an ontological similarity value for each training gene pair. As in a previous study[12], we define the similarity between two genes g_a and g_b by a simplified Resnik measure based on the training part of the target sub-ontology:

$$s'(g_a, g_b) = IC(LCA_{ab}) \stackrel{\text{def}}{=} -\log(|G_{LCA_{ab}}|/|G_{tot}|), \tag{1}$$

where LCA_{ab} is the lowest common ancestor term of genes g_a and g_b in the training part of the GO DAG, $IC(LCA_{ab})$ is its information content, $|G_{LCA_{ab}}|$ is the number of genes annotated by this lowest common ancestor term, and $|G_{tot}|$ is the total number of genes annotated by the terms in the training part (the "training genes"). A normalized score between 0 and 1 is then defined by dividing the simplified Resnik score by its maximum possible value: $s(g_a, g_b) = \frac{s'(g_a,g_b)}{-\log(1/|G_{tot}|)}$. Basically, if two genes are commonly annotated by a term that does not also annotate many other genes, they will receive a large similarity value based on this measure.

Gene pairs receiving a normalized score no less than a threshold t_s are considered semantically similar. Throughout the whole study, we set t_s to 0.3, which roughly corresponds to defining two genes as similar if they are commonly annotated by a term that annotates no more than 500 genes.

We then use these pairs of similar genes to filter the edges in each biological network (including both the training and left-out gene pairs), such that a large fraction of the retained edges are between semantically similar genes. Specifically, for

each network, we retain only edges with an edge weight no smaller than a threshold t_w, defined as the smallest value that leads to at least 50% of the retained training edges being semantically similar:

$$t_w = \arg\min_w \left[\frac{\sum_{(g_a,g_b)\in T: s(g_a,g_b)\geq t_s} \mathbb{1}(w_{ab} \geq w)}{\sum_{(g_a,g_b)\in T} \mathbb{1}(w_{ab} \geq w)} > 50\% \right], \quad (2)$$

where w_{ab} is the weight of the edge between gene g_a and gene g_b in the network, T is the set of training gene pairs, and $\mathbb{1}$ is the indicator function, i.e., $\mathbb{1}(true) = 1$ and $\mathbb{1}(false) = 0$. Assuming that the general relationships between network edge weights and ontological similarity values are the same for the training and left-out gene pairs, this filtering can effectively retain only the more relevant network edges in the left-out part for inferring the DAG of the sub-ontology. The reason to search for the smallest w that satisfies the requirement in Eq (2) is to retain as many edges in the network relevant to the GO sub-ontology as possible. To identify this t_w, we set w to the largest edge weight in the whole network at the beginning, and progressively reduce it to the next largest edge weight until the requirement in Eq (2) is satisfied. To handle the issue that the requirement in Eq (2) sometimes cannot be satisfied, or can only be satisfied with an extremely large value of t_w, if the percentage of semantically similar training gene pairs does not increase for 5 consecutive reductions of w, the value of w before these 5 reductions would be used as t_w.

Unification of heterogeneous networks by discretizing edge weights (Step 4)

Before the filtered networks can be integrated, one issue that we need to first handle is the very different distributions of edge weight values in these different networks. We have tried various standard ways to process these values, such as linearly scaling all edge weights to the range of 0 to 1. However, the resulting distributions of the different networks were still very different, and direct integration of these networks would place more emphasis on the networks with more edge weights closer to 1. On the other hand, methods such as quantile normalization destroy the original distribution of edge weights in each network and led to serious information loss.

We found a good strategy to unify these heterogeneous networks is to discretize the edge weights in each network into comparable numbers of discrete levels, such that 1) the order of edges based on their original weights is respected, and 2) for the training gene pairs, the consistency between their discretized weights and ontological similarity values is maximized.

To achieve these two goals, we designed a novel discretization algorithm. Given a set of training gene pairs and a biological network, the algorithm searches for a discretization M of the edge weights (i.e., a mapping of the original edge weights to the discrete levels) such that the following objective function is minimized:

$$O(M) = \sum_{(g_{a_1},g_{b_1}),(g_{a_2},g_{b_2})\in T} \mathbb{1}\Big(\; d\left(M(w_{a_1b_1}), M(w_{a_2b_2})\right) \neq d\left(s(g_{a_1},g_{b_1}), s(g_{a_2},g_{b_2})\right)\Big), \quad (3)$$

subject to the constraint that the order of the edges needs to be maintained, i.e., for any two training gene pairs (g_{a_1},g_{b_1}) and (g_{a_2},g_{b_2}), $M(w_{a_1b_1}) > M(w_{a_2b_2})$ only if $w_{a_1b_1} \geq w_{a_2b_2}$ and $M(w_{a_1b_1}) < M(w_{a_2b_2})$ only if $w_{a_1b_1} \leq w_{a_2b_2}$. In the objective function, d is the direction function defined as $d(x,y) = 1$ if $x > y$, $d(x,y) = 0$ if $x = y$ and $d(x,y) = -1$ if $x < y$. This objective function aims at minimizing the number of gene pairs that have different orders according to the discretized edge weight levels and according to their ontological similarity in the GO sub-ontology. Since in Step 6 of our pipeline CliXO is used to infer an ontology, and CliXO considers the order of edges in its clustering process rather than their absolute weights, our discretization procedure promotes gene pairs that are ontologically similar to be clustered earlier by CliXO.

We designed a searching algorithm to identify discretizations with a good objective score (Fig S4). Initially, all training gene pairs are sorted based on their edge weights and random partition points are added to divide them into k (set to 200 by default) ordered levels, where gene pairs with the same edge weights must be put in the same level. The algorithm then repeatedly refines the levels by randomly either moving some top gene pairs of a level (i.e., gene pairs with the largest original edge weights) to the next higher level, or moving some bottom gene pairs to the next lower level. If the objective score is improved, the new discretization is kept; Otherwise, the discretization is kept with a probability that reduces over time ($min\{0.1/iteration\_number, 0.001\}$), an idea similar to simulated annealing. The searching process stops after a maximum number of iterations (set to 10,000 by default), and the whole process is repeated multiple times using different random initial partitions. The discretization with the best objective score is then retrieved, and all edges in a level are given a new weight equal to the average of their original weights. Finally, some neighboring levels are combined to form 10-20 levels to avoid over-fitting in data integration step.

Integration of multiple biological networks (Step 5 and 6)

After discretizing the edge weights of each network individually, we integrate the networks by finding a linear combination of them that maximizes the Pearson correlation coefficient (*PCC*) with the semantic similarity values of the training gene pairs. Specifically, we find the coefficient vector $\mathbf{a}$ that maximizes $PCC(\mathbf{Ma}^t, \mathbf{s})$ subject to the constraint that $\sum_{i=1}^{N} a_i = 1$, where N is the total number of merged discrete levels in the different networks, $\mathbf{M}$ is a $|T| \times N$ matrix of discretized edge values of the $|T|$ training gene pairs, and $\mathbf{s}$ is a vector of the ontological similarity values of these training gene pairs. Each column in matrix $\mathbf{M}$ corresponds to one merged discrete level of one of the networks. Element (i, j) takes the value of the discretized edge weight of gene pair i if it belongs to the level represented by j, or 0 otherwise.

Since this optimization problem is a special case of canonical correlation analysis (CCA)[29] for finding the most correlated linear combinations of two sets of variables, we use a standard routine for CCA in Matlab to determine the coefficients $\mathbf{a}$.

After this step, all the biological networks are integrated into a single network with a new edge weight assigned to every (training and non-training) edge. This integrated network is then used as the input of CliXO to infer a DAG in which each node is formed by a cluster of genes and corresponds to a potential term in the target sub-ontology. CliXO is a hierarchical clustering method for grouping genes into potential terms. It starts by treating each gene as a node. Different nodes are merged to form a parent node of them if the genes contained in these nodes all have a similarity higher than a threshold with each other, where gene-gene similarity is defined based on the input biological network. If an edge is drawn between every two genes with a similarity higher than the threshold, each node is essentially a clique (i.e., a complete graph), which explains the name of the method (CliXO - Clique Extracted Ontology). The similarity threshold is set at a large value at the beginning, and is reduced progressively in rounds to allow more and more nodes to be merged together. CliXO also has additional steps to prune uninformative cliques and to allow for errors in the similarity values or imperfect cliques. These extra steps make the final output of CliXO not necessarily a tree, but a DAG in general.

Data and Experiment Settings (Step 7)

Biological networks

We used four public yeast networks that had also been used for inferring GO in previous studies[11,12], namely 1) correlation network of genetic interactions from DRYGIN(http://drygin.ccbr.utoronto.ca/DOWNLOAD/sgadata_costanzo2010_correlations.txt.gz)[16], co-expression network from Stanford Microarray Database (SMD)(Provided by Michael Kramer)[17], probabilistic functional gene network from YeastNet (v3)(http://www.inetbio.org/yeastnet/download.php?type=1)[18], and network of physical interactions (of types "direct interaction" and "physical association") from BioGRID(http://thebiogrid.org/downloads/archives/Release%20Archive/BIOGRID-3.3.122/BIOGRID-ORGANISM-3.3.122.mitab.zip)[19]. We considered only genes with at least one GO annotation. Some statistics of the four resulting networks are given in Table S1.

Since the edges in the BioGRID network were binary, we used a diffusion kernel[20] to produce numeric edge weights between 0 and 1, which resulted in larger weights for genes more (directly or indirectly) connected to each other.

Gene ontology definition and annotation files

We downloaded the gene ontology and annotation files from the Gene Ontology Web site (http://geneontology.org/). We processed these files in the same way as in previous studies[11,12].(Supplementary materials)

We downloaded two versions of GO ontology and annotation files. The first version (Ontology: 2-Dec-2014; Annotation: 29-Nov-2014) was the most updated version at the time we started the project and downloaded the files, which will be referred to as the 2014 version. The second version (Ontology: 31-Mar-2009; Annotation: 14-Mar-2009), which will be referred to as the 2009 version, represents an older version of GO that we used to test whether we could infer terms in the new version by combining the information in the old version and the biological networks. Some statistics of these two GO versions are given in Table S2. In addition to these two versions, in the part of our work that studied novel terms inferred by Unicorn, we also checked whether some of these terms were included in the latest version of GO at the time of paper writing (Ontology: 31-May-2016). This version will be referred to as the 2016 version.

For the 2014 version of GO, using the criteria we defined for selecting training parts described in Section ***Selection of training part***, we got 12, 12 and 9 training parts from BP, CC and MF, respectively.

Ontology alignment and performance evaluation

We used a slightly modified version (explained below) of the method described previously[11] to align an ontology inferred by Unicorn with the actual GO sub-ontology. Briefly, a mapping of the terms in the two ontologies was produced to align highly similar terms based on the genes they annotate, with the constraints that 1) each term in the inferred ontology could be aligned to at most one term in the GO sub-ontology, and 2) the aligned term pairs could not crisscross. We used a false discovery rate of 5% as the cutoff to define a pair of terms to be aligned.

To objectively evaluate the performance of our inferred ontology using information not involved in the training process, we designed the following evaluation procedure (Fig S1). Given a target GO sub-ontology O_G and a chosen sub-tree of it O_T, we first inferred an ontology $O_{G'}$ from all genes using Unicorn with O_T as the training part. Next, we considered only the genes annotated by terms in O_T to infer another ontology $O_{T'}$ using O_T as the training part, and aligned it with $O_{G'}$. For any term in $O_{G'}$ aligned to a term in $O_{T'}$, we considered it a term inferred due to information directly from the training part. Finally, we aligned $O_{G'}$ and O_G, and used only the aligned terms in $O_{G'}$ not considered to be due to the training part to evaluate the performance of the inferred ontology. Specifically, if $A(O_x, O_y)$ is the set of aligned term pairs from ontologies O_x and O_y, we defined Hit as the number of terms in $O_{G'}$ aligned to O_G but not due to the training part, i.e., $Hit = |\{(t_{G'}, t_G) \in A(O_{G'}, O_G) : t_G \notin O_T \wedge \neg \exists t_{T'} s.t. (t_{G'}, t_{T'}) \in A(O_{G'}, O_{T'})\}|$. Three performance metrics were then defined accordingly:

$$Precision = \frac{Hit}{|O_{G'}| - |A(O_{G'}, O_{T'})|}, \tag{4}$$

$$Recall = \frac{Hit}{|O_G - O_T|}, \tag{5}$$

$$F-score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{6}$$

where $|O_x|$ is the number of terms in an ontology x and $|O_G - O_T|$ is the set of terms in the GO sub-ontology not in the training part.

In the original alignment algorithm[11], the similarity between two terms from the two ontologies is based on both the genes they annotate (their "intrinsic similarity") and their parent and child terms (their "hierarchical similarity"). In our case, when we aligned $O_{G'}$ and O_G, some of the parent/child terms were those considered to be due to the training part. In order to remove any effects of the training part in our performance evaluation, we modified the alignment algorithm to consider only the intrinsic similarity between two terms in all our experiments.

Results

Edge filtering increased fraction of informative edges

In the filtering step of Unicorn (Step 3), some edges are removed such that among the training gene pairs with a retained edge, a larger fraction of them are informative (i.e., having an ontological similarity larger than threshold t_s) after the filtering. We checked whether the filtering also increased the fraction of informative edges among the left-out genes as judged by their actual ontological similarity according to the GO sub-ontology (which was not disclosed to Unicorn). As shown in Table 1, indeed for all three GO sub-ontologies and all four biological networks, the filtering increased the fraction of informative edges among left-out gene pairs, thus verifying the effectiveness of the filtering step.

		DRYGIN	SMD	YeastNet	BioGRID
BP	Before filtering	11.75%	19.20%	40.44%	40.39%
	After filtering	47.79%	70.43%	67.84%	53.41%
CC	Before filtering	3.47%	7.17%	21.43%	26.75%
	After filtering	41.38%	73.46%	59.62%	58.00%
MF	Before filtering	3.73%	5.46%	16.05%	13.42%
	After filtering	15.66%	74.70%	57.23%	32.90%

Table 1. Average fraction of informative edges in the biological networks among the left-out genes before and after edge filtering. These values were obtained by averaging over all the training parts of each sub-ontology.

Unicorn improved accuracy of ontology inference

We then checked the accuracy of ontology inference of Unicorn based on the left-out parts. CliXO contains two key parameters, namely α (for reducing noise by adding a margin to the similarity threshold when forming cliques) and β (for inferring missing edges by allowing near-complete graphs as new terms). We set β to 0.5 as previously suggested[12], and varied the value of α such that each set of results contained points from one extreme (high precision, low recall) to the other extreme (high recall, low precision).

Fig 2 shows the overall F-measure of Unicorn as compared to running CliXO on individual networks and a simple benchmark method, averaging over the parameter values. In this benchmark method, the weight of an edge in the integrated network is simply the summation of its weight in the original networks, which assumes equal importance of the input networks.

It is seen that when inferring BP, information from BioGRID was most useful followed by YeastNet. On the other hand, when inferring CC, YeastNet was most useful followed by DRYGIN and BioGRID. Finally, when inferring MF, YeastNet was most useful followed by BioGRID. These results indicate that the different networks should be integrated differently when inferring the three sub-ontologies. Indeed, a simple summation of the four networks led to improved F-measure only for CC but not in the cases of BP, MF and the overall average. On the other hand, by having a semi-supervised framework that processes and integrates the networks specific to the target GO sub-ontology, Unicorn was able to achieve better average F-measure values both when inferring each sub-ontology and averaging over all three sub-ontologies overall, as compared to using individual networks as input.

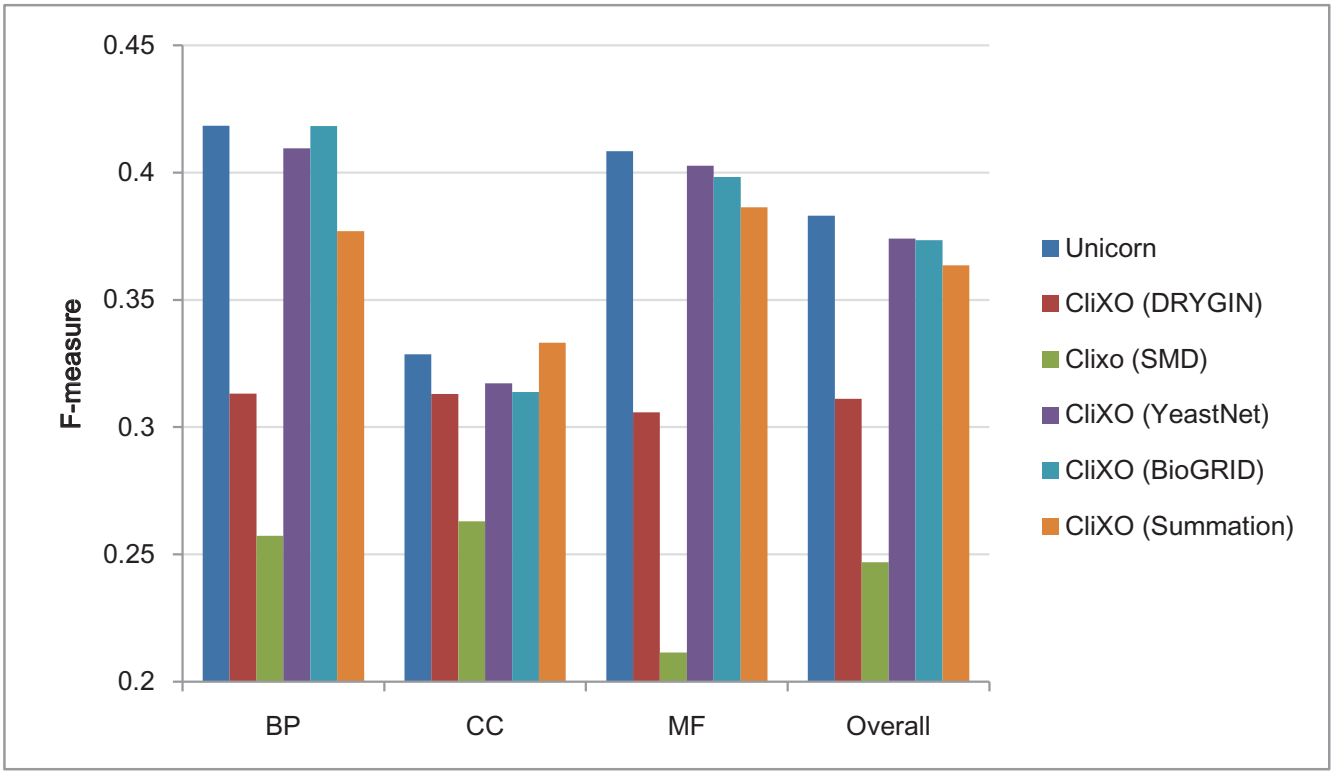
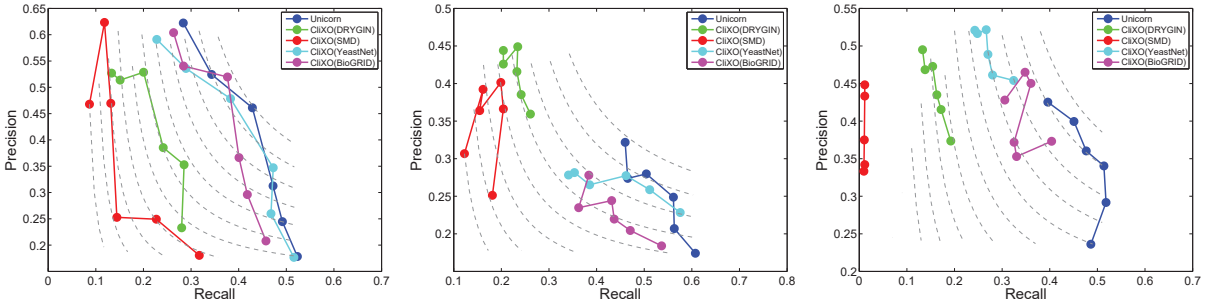


Figure 2. The average F-measures of CliXO with either Unicorn-produced integrated network, a single biological network, or a simple summation of the input networks. Each reported F-measure is the average among the results from all the training parts of one GO sub-ontology (in the case of "BP", "CC" and "MF") or across all three sub-ontologies (in the case of "Overall"), over all parameter values of CliXO.



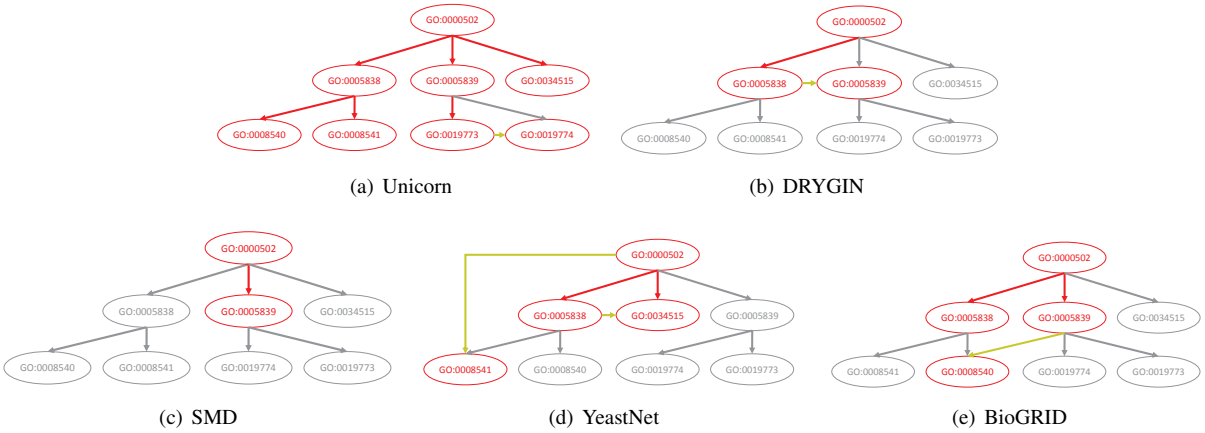
(a) BP (b) CC (c) MF

Figure 3. Ontology inference left-out accuracy of Unicorn and single biological networks. The training parts were the sub-trees rooted at GO:0051179 (localization), GO:0016020 (membrane) and GO:1901363 (heterocyclic compound binding) for BP, CC and MF, respectively.

Fig 3 shows some examples of the comparison results in the form of precision-recall graphs. In each graph, each approach

is represented by a curve joining different points that correspond to the results when running CliXO with different α values. The dotted curves in the background are contour lines that connect points with the same F-measure score. From the graphs, the left-out parts of the ontologies inferred by Unicorn have higher F-scores in general, as seen by their positions closer to the upper-right corner.

Fig 4 gives an example illustrating the importance of integrating the biological networks. It shows the ability of different approaches in inferring the sub-tree of the CC sub-ontology rooted at the CC term GO:0000502 (proteasome complex) when the sub-tree rooted at the term GO:0043226 (organelle) was used as the training part. As seen in the figure, while each individual network was sufficient to infer part of the sub-tree, only when the networks were integrated was it possible to infer all the terms.



(a) Unicorn (b) DRYGIN

(c) SMD (d) YeastNet (e) BioGRID

Figure 4. Ability to infer the sub-tree rooted at GO:0000502 (proteasome complex) by single biological networks and Unicorn with the sub-tree rooted at GO:0043226 (organelle) as the training part. The colors represent successfully inferred (red), missed (gray) and novel (yellow) terms and term-term relationships as compared to the 2014 version of GO.

Re-discovering terms in new version of GO by combining information in an old version of GO with biological networks

While the above results have confirmed the accuracy of Unicorn using left-out parts of GO not involved in the training process, the ultimate use of Unicorn is to infer novel terms not already contained in GO. The first way we attempted to test this possibility was to combine the information in the biological networks and an old (2009) version of GO, to see if Unicorn could infer terms that were only in a new (2014) version of GO.

By running Unicorn with the 2009 version of GO as input, the inferred DAG contained nodes that could not be aligned to any term in this version of GO. Based on the CliXO procedufre, each of these nodes contained a set of genes and was connected to other nodes in the inferred DAG. Each such node can therefore be considered a potential novel term that annotates these genes and are related to other existing terms in the 2009 version of GO based on their connections in the inferred DAG. We then aligned all the terms in the inferred DAG with the 2014 version of GO, and found some of the nodes not aligned to the 2009 version of GO actually aligned to some nodes in the 2014 version. Specifically, we identified 3-19, 6-10 and 1-6 cases in BP, CC and MF, respectively for different values of α when running CliXO. Fig 5 and Fig S2 show some of the examples.

In these examples, we see that Unicorn is able to infer both general (upper-level) and specific (lower-level) terms present only in the 2014 version of GO. It is possible that some of the Unicorn-inferred terms that cannot be aligned to either the 2009 or 2014 version of GO (the ones in gray) are biologically meaningful novel terms. In fact, for some of them (the nodes in gray with GO term IDs) we actually find nodes in the 2016 version of GO connecting to the corresponding parent and child terms as in our inferred DAG. For the remaining novel terms, we further explore their potential meanings in the next section.

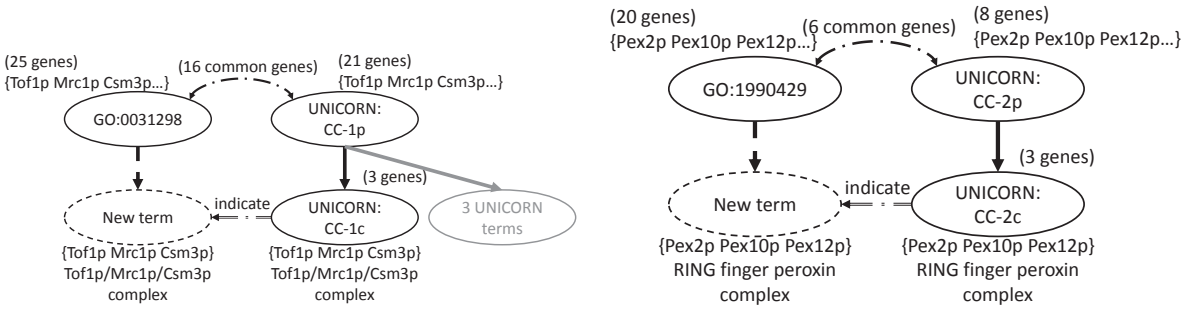
Discovery of biologically meaningful novel terms

Unicorn inferred a large number of novel terms not contained in either the 2009, 2014 or 2016 version of GO. To investigate their potential meanings, we extracted the list of genes annotated by them and looked for descriptions of these gene groups in the literature. Some examples with supports from the CYC 2008 protein complex database[21] are shown in Fig 6.

In the first example (Fig 6a), Unicorn identified a sub-complex of the replication fork protection complex (GO:0031298) involving three proteins Csm3p, Mrc1p and Tof1p. These three proteins form the replication fork-pausing complex (FPC)[22,23], which is associated with replication sites and prevents genomic instability through mediating checkpoint signaling in stationary-phase cells[24].

(a) (b)

Figure 5. Some terms inferred by Unicorn by combining the information in the biological networks and the 2009 version of GO. The colors represent terms and term-term relationships only present in the 2014 version of GO but not the 2009 version (red), present in both the 2009 and 2014 versions of GO (blue), and absent in both the 2009 and 2014 versions of GO (gray), but present in the 2016 version. These two terms were all inferred from BP.



(a) (b)

Figure 6. Some biologically meaningful novel terms inferred by Unicorn. In each panel, the terms on the right were inferred by Unicorn.

In the second example (Fig 6b), Unicorn identified a sub-complex of the peroxisomal importomer complex (GO:1990429) involving three proteins Pex2p, Pex10p and Pex12p. These three proteins form the RING finger peroxin complex[25–27], which was considered to function in peroxisomal matrix protein import by recycling receptors[25].

Six additional novel terms are shown in Fig S3 and their literature supports are given in the supplement. We also provide on our supplementary Web site a list of unverified novel terms with either a parent or child term aligned to a GO term with score $>$ 0.8.

Discussion

In this paper, we proposed a semi-supervised framework to integrate multiple biological networks for better automatic inference of Gene Ontology. The Results based on the left-out parts of GO not involved in training confirmed the accuracy of the inferred ontologies. The better performance of Unicorn as compared to CliXO in some of the experimental results were due to the semi-supervised nature of Unicorn, which allowed it to integrate both the information in the biological networks and in the training part of GO. These training data helped Unicorn to 1) determine the most relevant network edges to retain, 2) discretize network edges such that multiple heterogeneous networks can be easily integrated, and 3) determine the best way to integrate these networks by maximizing the correlation between the edge weights in the resulting integrated network and the ontological similarity of the training part. All these novel components contributed to the construction of an integrated network more suitable for CliXO to infer GO from.

We were also able to rediscover terms in a new (2014) version of GO based on information in an old (2009) version, and discover novel terms that were shown to be biologically meaningful. Unicorn can thus be used to propose new terms for further manual validation and curation.

We selected four biological networks in our study based on the successful use of them in inferring GO in some previous work[11,12]. We showed that these four networks contributed unequally, and for each GO sub-ontology the way to use them should be customized, which highlights the advantage of a supervised or semi-supervised approach as compared to previous unsupervised approaches.

It is useful to explore the integration of more types of biological network such as those based on the evolutionary relationships of the genes, and the possibility to apply Unicorn to other species and other types of biological ontology.

One of the main uses of GO is functional enrichment analyses. The DAGs constructed by Unicorn provide a putative set of terms potentially useful for explaining the functional relationships between some genes. An advantage of using these Unicorn-constructed terms is that the molecular basis of them can be easily traced back from the similarity of the genes in the integrated network, with the importance of each network indicated by its respective coefficient in the integration formula.

Acknowledgement

We would like to thank Michael Kramer for providing the source code of CliXO and data files of SMD. We also thank Michael Kramer, Chuan Luo and Yuxi Wang for helpful discussions. KYY is supported by a CUHK VC discretionary fund.

Author contributions statement

LL and KYY conceived the study, designed the methods and experiments, analysed the results, and wrote the manuscript. LL implemented the methods and conducted the experiments.

Additional information

Competing Financial Interests

The authors declare no competing financial interests.

References

1. Ashburner, M. *et al.* Gene ontology: Tool for the unification of biology. *Nature Genetics* **25**, 25–29 (2000).
2. Bettembourg, C., Diot, C. & Dameron, O. Semantic particularity measure for functional characterization of gene sets using gene ontology. *PLOS One* **9**, e86525 (2014).
3. Mistry, M. & Pavlidis, P. Gene ontology term overlap as a measure of gene functional similarity. *BMC Bioinformatics* **9**, 327 (2008).
4. Schlicker, A., Domingues, F. S., Rahnenführer, J. & Lengauer, T. A new measure for functional similarity of gene products based on gene ontology. *BMC Bioinformatics* **7**, 302 (2006).

5. Cheng, L., Lin, H., Hu, Y., Wang, J. & Yang, Z. Gene function prediction based on the gene ontology hierarchical structure. *PLOS ONE* **9**, e107187 (2014).

6. Jensen, L. J., Gupta, R., Staerfeldt, H.-H. & Brunak, S. Prediction of human protein function according to gene ontology categories. *Bioinformatics* **19**, 635–642 (2003).

7. Tao, Y., Sam, L., Li, J., Friedman, C. & Lussier, Y. A. Information theory applied to the sparse gene ontology annotation network to predict novel gene function. *Bioinformatics* **23**, i529–i538 (2007).

8. Reimand, J., Arak, T. & Vilo, J. g:profiler–a web server for functional interpretation of gene lists (2011 update). *Nucleic Acids Research* **39**, W307–W315 (2011).

9. Robinson, P. N., Wollstein, A., Böhme, U. & Beattie, B. Ontologizing gene-expression microarray data: Characterizing clusters with gene ontology. *Bioinformatics* **20**, 979–981 (2004).

10. Zhang, B., Schmoyer, D., Kirov, S. & Snoddy, J. GOTree machine (GOTM): a web-based platform for interpreting sets of interesting genes using gene ontology hierarchies. *BMC Bioinformatics* **5**, 16 (2004).

11. Dutkowski, J. *et al.* A gene ontology inferred from molecular networks. *Nature Biotechnology* **31**, 38–45 (2013).

12. Kramer, M., Dutkowski, J., Yu, M., Bafna, V. & Ideker, T. Inferring gene ontologies from pairwise similarity data. *Bioinformatics* **30**, i34–i42 (2014).

13. Gligorijević, V., Janjić, V. & Pržulj, N. Integration of molecular network data reconstructs gene ontology. *Bioinformatics* **30**, i594–i600 (2014).

14. Peng, J., Wang, T., Wang, J., Wang, Y. & Chen, J. Extending gene ontology with gene association networks. *Bioinformatics* **32**, 1185–1194 (2015).

15. Glass, K. & Girvan, M. Finding new order in biological functions from the network structure of gene annotations. *PLOS Computational Biology* **11**, e1004565 (2015).

16. Costanzo, M. *et al.* The genetic landscape of a cell. *Science* **327**, 425–431 (2010).

17. Hubble, J. *et al.* Implementation of genepattern within the stanford microarray database. *Nucleic Acids Research* **37**, D898–D901 (2009).

18. Kim, H. *et al.* YeastNet v3: A public database of data-specific and integrated functional gene networks for saccharomyces cerevisiae. *Nucleic Acids Research* **42**, D731–D736 (2013).

19. Stark, C. *et al.* BioGRID: A general repository for interaction datasets. *Nucleic Acids Research* **34**, D535–D539 (2006).

20. Kondor, R. I. & Lafferty, J. D. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 315–322 (2002).

21. Pu, S., Wong, J., Turner, B., Cho, E. & Wodak, S. J. Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Research* **37**, 825–831 (2009).

22. Bando, M. *et al.* Csm3, tof1, and mrc1 form a heterotrimeric mediator complex that associates with DNA replication forks. *The Journal of Biological Chemistry* **284**, 34355–34365 (2009).

23. Nedelcheva, M. N. *et al.* Uncoupling of unwinding from DNA synthesis implies regulation of MCM helicase by tof1/mrc1/csm3 checkpoint complex. *Journal of Molecular Biology* **347**, 509–521 (2005).

24. Alver, B., Kelly, M. K. & Kirkpatrick, D. T. Novel checkpoint pathway organization promotes genome stability in stationary-phase yeast cells. *Molecular and Cellular Biology* **33**, 457–472 (2013).

25. Brown, L.-A. & Baker, A. Shuttles and cycles: Transport of proteins into the peroxisome matrix (review). *Molecular Membrane Biology* **25**, 363–375 (2008).

26. El Magraoui, F. *et al.* The RING-type ubiquitin ligases pex2p, pex10p and pex12p form a heteromeric complex that displays enhanced activity in an ubiquitin conjugating enzyme-selective manner. *FEBS Journal* **279**, 2060–2070 (2012).

27. Prestele, J. *et al.* Different functions of the C3HC4 zinc RING finger peroxins PEX10, PEX2, and PEX12 in peroxisome formation and matrix protein import. *Proceedings of the National Academy of Sciences of the United States of America* **107**, 14915–14920 (2010).

28. Resnik, P. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research* **11**, 95–130 (1999).

29. Krzanowski, W. *Principles of Multivariate Analysis: A User's Perspective* (Oxford University Press, 1988).

Supplementary materials for Integrating Information in Biological Ontologies and Molecular Networks to Infer Novel Terms

Le Li[1] **and Kevin Y. Yip**[1,2,3,4*]

[1]Department of Computer Science and Engineering,
[2]Hong Kong Bioinformatics Centre,
[3]CUHK-BGI Innovation Institute of Trans-omics, and
[4]Hong Kong Institute of Diabetes and Obesity, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong
[*]kevinyip@cse.cuhk.edu.hk

Supplementary methods

Following Dutkowski et al. (2013)[1], the Gene Ontology term definition and gene annotation files were processed as follows:

- Terms labeled as 'is_obsolete' were removed.
- For any pair of terms (x, y), if their relationship was specified as either 'x is_a y', 'x part_of y', 'y has_part x', 'x regulates y', 'x positively_regulates y' or 'x negatively_regulates y', an edge was drawn from the node that represents x to the node that represents y.
- Annotations labeled as 'NOT' were removed.
- Only genes contained in at least one of the biological networks were considered.
- If a gene was annotated by a term, it was also considered as annotated by all ancestors of the term[2] according to the True Path Rule[3].
- Terms annotating zero or only one gene were discarded since there would be no way to recover such terms by CliXO[4].
- If a term x and a child term of it y annotated exactly the same set of genes, x would be removed and the y would inherit all its informative relations. All other child terms of x would become child terms of y.

Supplementary results

1. Vac14p and Fig4p were shown to form a complex[5,6] that regulates phosphatidylinositol 3.5-bisphosphate synthesis and turnover, which is part of the function of the PAS complex (GO:0070772).
2. Gyp5p and Gyl1p were considered to form a complex[7,8], which co-purifies with post-Golgi vesicles[7] and is thus related to GO:0005798 (Golgi-associated vesicle).
3. Crh1p and Bug1p form a complex at the cis-Golgi network (GO:0005801)[9,10], and it was suggested to contribute to a redundant network of interactions that mediate consumption of COPII vesicles and formation of the cis-Golgi[9].
4. Bi4p and Nam2p form a complex required for splicing bI4 of the yeast COB gene[11], which suggests that the Bi4p/Nam2p complex should be associated with GO:0000372 (Group I intron splicing).
5. Coq3 and Coq4 form the Q-biosynthetic Coq polypeptide complex[12], which was verified to exist in yeast mitochondria for the biosynthesis of coenzyme Q (ubiquinone)[13]. This suggests the inferred term should be related to GO:0006744 (Ubiquinone biosynthetic process).
6. Sur1p and Csg2p form a sub-complex of the Sur1p/Csg2p/Csh1p peroxisomal importomer complex (GO:1990429) with an unknown function, and was required for growth of yeast under high calcium concentrations[14].

Supplementary tables

Method	CliXO	Unicorn
Type	Unsupervised	Semi-supervised
Filtering of network edges	Fixed thresholds	Thresholds learned from training part of GO
Unification of heterogeneous networks	Nil	By means of discretization
Integration of networks	Fixed scheme	Weights learned from training part of GO

Table S1. Major differences between the CliXO and Unicorn methods

Network	DRYGIN	SMD	YeastNet	BioGRID
Genes (before filtering)	3,919	4,627	5,805	5,557
Edges (before filtering)	6,442,244	2,906,334	361,984	69,680
Edge weight type	continuous	continuous	continuous	binary
Edge weight range	(0,1)	(0,1)	(0.934,5.740)	$\{0,1\}$
Genes* (after filtering)	2,142	2,102	4,924	4,589
Edges* (after filtering)	6,349	18,183	54,332	28,916

Table S2. Statistics of the four yeast biological networks (*Average number for the different target sub-ontologies and training parts considered)

Version		2014	2009
BP	Terms	2,850	2,146
	Genes annotated	5,910	5,898
	Term-term relationships	7,078	4,297
CC	Terms	734	617
	Genes annotated	5,912	5,898
	Term-term relationships	1,721	1,363
MF	Terms	1,252	1,619
	Genes annotated	5,909	5,896
	Term-term relationships	1,740	1,998

Table S3. Statistics of the 2009 and 2014 versions of GO used in the experiments

Supplementary figures

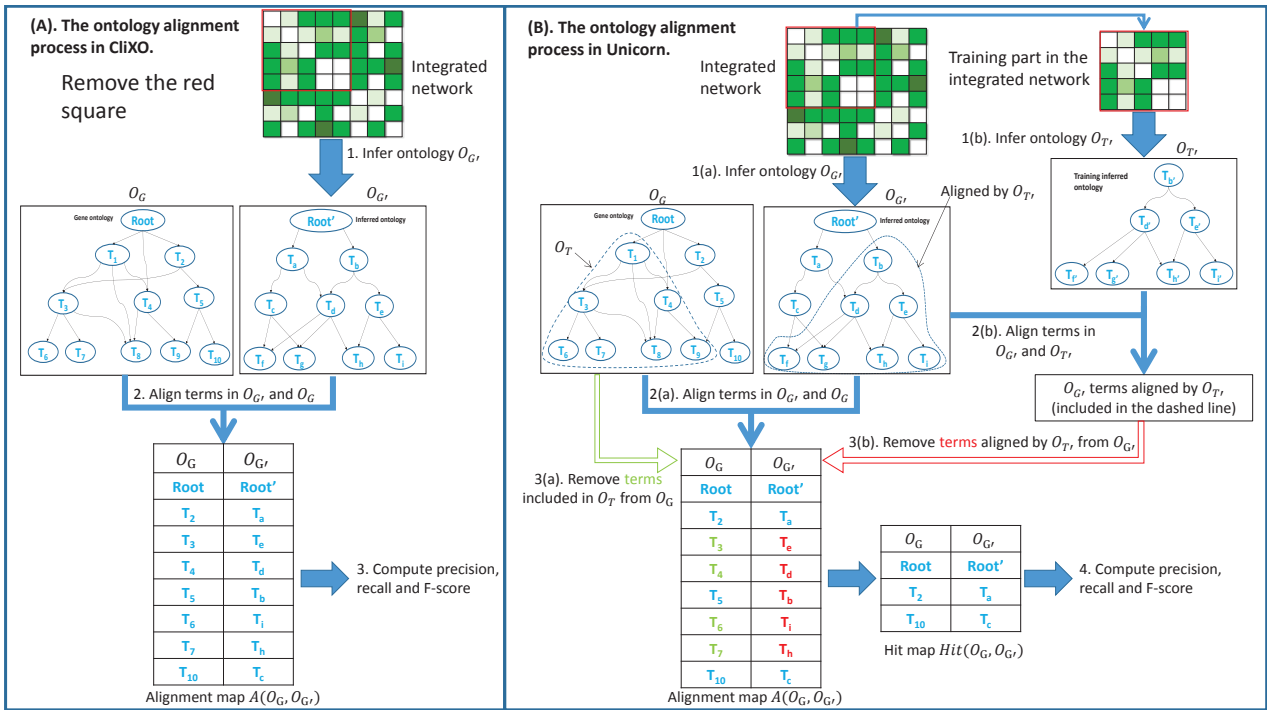


Figure S1. The training-testing procedure for evaluating the effectiveness of Unicorn. Panel A shows the procedure for evaluating an ontology inferred by CliXO without going through the Unicorn process. The input network (either a single biological network or an integrated network) is used to infer an ontology $O_{G'}$, the terms of which are aligned with the terms in the actual GO sub-ontology O_G. Based on the number of aligned terms and the total number of terms in $O_{G'}$ and O_G, precision, recall and F-measure can be computed. In contrast, Panel B shows the modified procedure for ontologies inferred with Unicorn. In this case, the training part is used to learn the parameters for filtering, unifying and integrating biological networks. The resulting integrated network, involving both the training and left-out parts, is used to infer an ontology $O_{G'}$. The terms of it are aligned to the terms in the actual GO sub-ontology O_G. However, instead of using the alignment results to evaluate the effectiveness of Unicorn directly, terms in $O_{G'}$ that are likely due to the training part are first removed (see main text for the details), before the resulting list of aligned term pairs is used to compute the performance metrics.

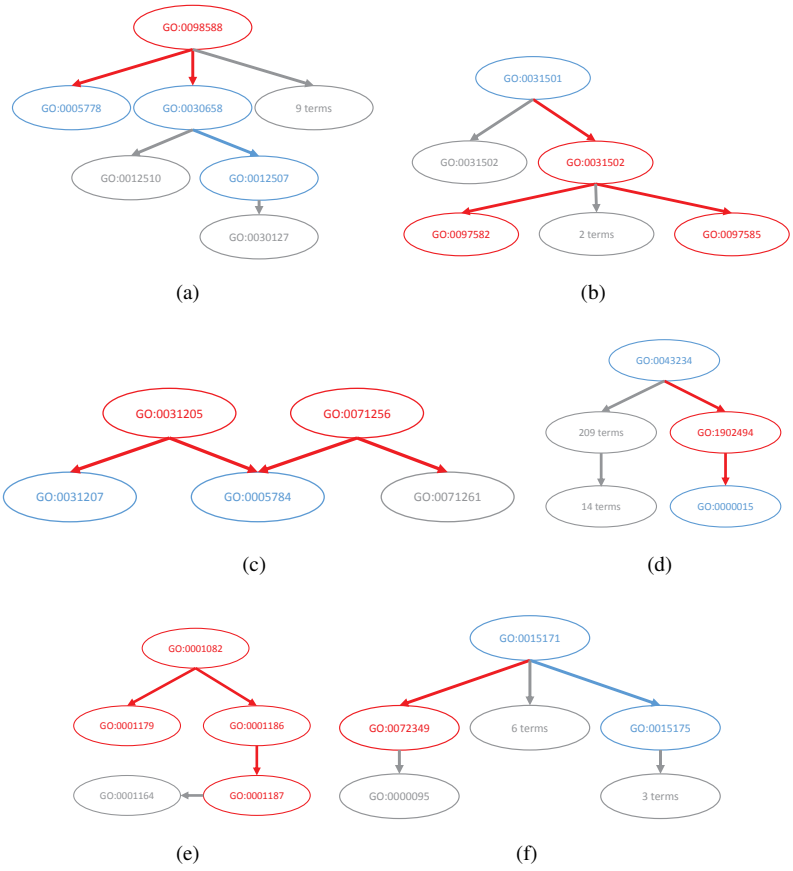


Figure S2. Some terms inferred by Unicorn by combining the information in the biological networks and the 2009 version of GO. The colors represent terms and term-term relationships only present in the 2014 version of GO but not the 2009 version (red), present in both the 2009 and 2014 versions of GO (blue), and absent in both the 2009 and 2014 versions of GO (gray). Some of the terms absent in both the 2009 and 2014 versions are present in the 2016 version, and the corresponding Go term IDs are shown. These terms were inferred from CC (a-d) and MF (e-f), respectively.

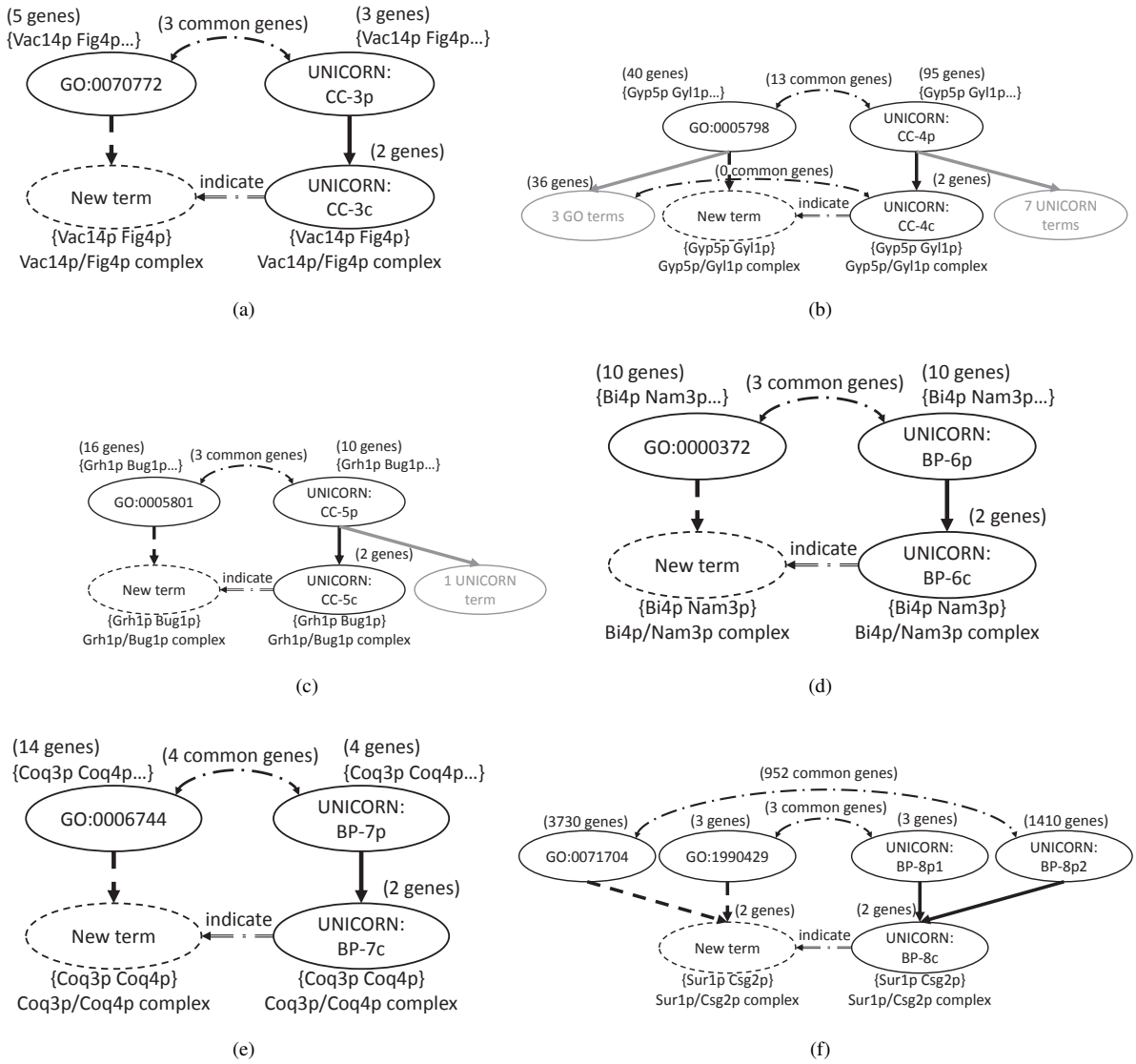


Figure S3. Additional biologically meaningful novel terms inferred by Unicorn. In each panel, the terms on the right were inferred by Unicorn.

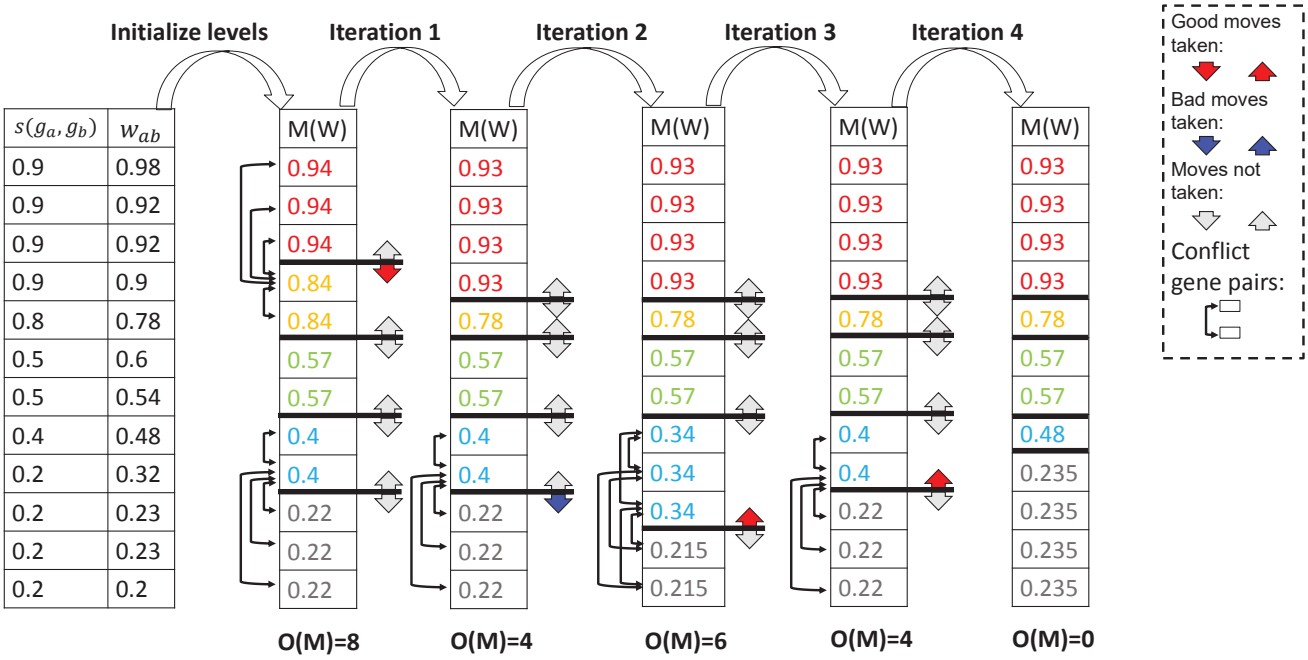


Figure S4. An example illustrating the discretization process. Each row in the tables corresponds to one training gene pair. The thick horizontal bars represent the level partitions. At the beginning, the four randomly added partitions divide the training gene pairs into five levels. Each gene pair in a level receives a new edge weight equal to the average of the original edge weights of all the pairs in that level. This initial partitioning has an objective score of $O(M) = 8$, and the 8 corresponding pairs of conflicting gene pairs are indicated in the figure. Then, each partition can either move up or down (assuming moving step of 1 gene pair in this example, but the step size is arbitrary in the actual algorithm). If a move leads to an improved (i.e., reduced) objective score (indicated by a red arrow), the move will be taken. Otherwise, it will be taken with a certain probability (indicated by a blue arrow), and not taken otherwise (indicated by a gray arrow). In this example, after 4 iterations, the object score improves from 8 to 0. In the actual algorithm, some of the neighboring levels will be further merged, which is not shown in this example.

References

1. Dutkowski, J. *et al.* A gene ontology inferred from molecular betworks. *Nature Biotechnology* **31**, 38–45 (2013).
2. Huntley, R. P., Sawford, T., Martin, M. J. & O'Donovan, C. Understanding how and why the gene ontology and its annotations evolve: the GO within UniProt. *Gigascience* **3**, 4 (2014).
3. Gene Ontology Consortium. Creating the gene ontology resource: Design and implementation. *Genome Research* **11**, 1425–1433 (2001).
4. Kramer, M., Dutkowski, J., Yu, M., Bafna, V. & Ideker, T. Inferring gene ontologies from pairwise similarity data. *Bioinformatics* **30**, i34–i42 (2014).
5. Dove, S. K. *et al.* Vac14 controls ptdins (3,5)p(2) synthesis and fab1-dependent protein trafficking to the multivesicular body. *Current Biology* **12**, 885–893 (2002).
6. Rudge, S. A., Anderson, D. M. & Emr, S. D. Vacuole size control: Regulation of ptdins (3,5)p(2) levels by the vacuole-associated vac14-fig4 complex, a ptdins(3,5)p(2)-specific phosphatase. *Molecular Biology of the Cell* **15**, 24–36 (2004).
7. Chesneau, L. *et al.* Gyp5p and gyl1p are involved in the control of polarized exocytosis in budding yeast. *Journal of Cell Science* **117**, 4757–4767 (2004).
8. Chesneau, L. *et al.* Interdependence of the ypt/rabgap gyp5p and gyl1p for recruitment to the sites of polarized growth. *Traffic* **9**, 608–622 (2008).
9. Behnia, R., Barr, F. A., Flanagan, J. J., Barlowe, C. & Munro, S. The yeast orthologue of GRASP65 forms a complex with a coiled-coil protein that contributes to ER to golgi traffic. *The Journal of Cell Biology* **176**, 255–261 (2007).
10. Čopič, A. *et al.* Genomewide analysis reveals novel pathways affecting endoplasmic reticulum homeostasis, protein modification and quality control. *Genetics* **182**, 757–769 (2009).
11. Rho, S. & Martinis, S. A. The bi4 group i intron binds directly to both its protein splicing partners, a tRNA synthetase and maturase, to facilitate RNA splicing activity. *RNA* **6**, 1882–1894 (2000).
12. Pu, S., Wong, J., Turner, B., Cho, E. & Wodak, S. J. Up-to-date catalogues of yeast protein complexes.

    Nucleic Acids Research **37**, 825–831 (2009).
13. Marbois, B. *et al.* Coq3 and coq4 define a polypeptide complex in yeast mitochondria for the biosynthesis of coenzyme q. *Journal of Biological Chemistry* **280**, 20231–20238 (2005).
14. Jo, W. J. *et al.* Identification of genes involved in the toxic response of saccharomyces cerevisiae against iron and copper overload by parallel analysis of deletion mutants. *Toxicological sciences* **101**, 140–151 (2007).