DATA-DRIVEN ONTOLOGIES

JAMES C COSTELLO [1,2]* DAN SCHRIDER [1] JEFF GEHLHAUSEN [1]
MEHMET DALKILIC[1,3]
[1] *School of Informatics,*
[2] *Department of Biology,*
[3] *Center for Genomics and Bioinformatics,*
Indiana University, Bloomington IN, 47405

Gene networks are important tools in studying gene-gene relationships and gene function. Understanding the relationships within these networks is an important challenge. Ontologies are a critical tool in helping deal with these data. The use of the Gene Ontology, for example, has become routine in methods for validation, discovery, *etc.* Here we present a novel algorithm that synthesizes an ontology by considering both extant annotation terms and also the connections between genes in gene networks. The process is efficient and produces easily inspectable ontologies. Because the relationships drawn between terms are heavily influenced by data, we call these "Data-Driven" Ontologies. We apply this algorithm to both discover new relationships between biological processes and as a tool to compare sets of genes across microrarray experiments. Supplemental data and source code are available at: `http://www.ddont.org`

1. Introduction

Researchers have unprecedented access to biological data organized in community supported and publicly available repositories; however, leveraging this enormous amount of data is neither routine nor simple.[1] Graph-theoretic and statistical approaches (often termed network approaches) are among the most popular means of studying complex biological relationships. Specific implementations of networks include, but are not limited to, coexpression networks,[2] protein-protein interaction networks,[3,4,5] and in the form of integrated functional linkage networks.[6,7,8] Extracting meaningful information from these networks has now become a challenge.

Ontologies are increasingly being employed to systematically organize data, thereby facilitating interpretation and exchange of information. The

*to whom correspondence should be sent. *email*: jccostel@indiana.edu purpose of an ontology is, from observing the real world, to identify the pertinent existing instances and then to codify the relationships that exist between them.[9,10] In the context of biology, the Gene Ontology (GO)[11] is a well-known example; however, the GO is one of many potential and valid ontologies to describe any and all properties of genes. In short, there is no *canonical* ontological form. Indeed, multiple ontologies reveal relationships otherwise absent from a single ontology. Outside the GO, there are examples of ontologies that relate genes to diseases,[a] genes to organism anatomy,[b] and genes to gene expression data.[c]

In this study we present an algorithm that constructs an ontology by taking both gene network relationships and extant annotations on genes as inputs. We call this a "data-driven ontology" (hereafter abbreviated as DDOnt) to distinguish the process from building an ontology solely from human design. The DDOnts are "data-driven" in the sense that the relationships drawn between terms are influenced by the gene network connections. Our motivation is to use DDOnts to reveal potentially novel and interesting biological relationships that would not arise without the influence of gene networks. The complete deterministic, recursive algorithm is presented in Sec. 2.2.

We apply our method to three distinct, biologically motivated applications. First, given any gene network (*e.g.* relevance network,[2] integrated functional gene network,[7,6]) we synthesize a DDOnt to reveal novel relationships between biological processes. Second, given a set of genes of interest, we use gene network data to construct a DDOnt to find associated biological processes. Third, given a predefined set of genes, we explore how their associated biological processes are affected by varying experimental conditions as measured through microarray gene expression assays.

2. Methods

2.1. *Problem Statement*

Given a graph $G\langle V, E\rangle$ and a lexicon L, where lexical element $l_i \in L$ is mapped to 1 or more vertices of $G\langle V, E\rangle$, can we synthesize an ontology $\mathcal{O}$ where the relationships between lexical elements in L are determined through both annotation frequency and the edges in $G\langle V, E\rangle$?

$G\langle V, E\rangle$ is a set of genes $d_i \in V$ connected as edges $\{d_i, d_j\} \in E$ that

[a] http://diseaseontology.sourceforge.net/
[b] http://flybase.org/
[c] http://www.evocontology.org/

are determined from a dataset (*e.g.* coexpression correlation). Throughout all examples in this study we define the lexicon L as the set of terms in the GO category "biological process." When constructing $\mathcal{O}$, we follow two rules-of-thumb to draw connections between elements in L. 1) Lexical terms that are annotated to tightly connected genes in $G\langle V, E\rangle$ should be closely related in $\mathcal{O}$. 2) The prevalence of l_i is related to the distance of that term to a root, *i.e.* increasing prevalence means nearer to a root, decreasing prevalence means further from a root. We define a term's prevalence as a combination of two properties: (i) the count of genes annotated with l_i, and (ii) the connectivity of the genes annotated with l_i in $G\langle V, E\rangle$.

We have informally typed all directed edges that are drawn between lexical elements $\{l_i, l_j\}$ as: `is-related-to`. These edges are parent-child relationships where the parent is more prevalent (defined above) than the child.

2.2. *Ontology Construction Algorithm*

BuildOntology is a deterministic, recursive algorithm that takes four inputs and produces an ontology $\mathcal{O}$ in the form of a tree.

The first input to BuildOntology is $\mathbf{M}$. Given an annotated graph $G_{D,L}\langle V, E\rangle$ two matrices are constructed: one to reflect the relationships between nodes in the graph and the other to reflect the lexical annotations on the nodes. Matrix $\mathbf{M}_{D,D}$ is an adjacency matrix where $[a_{i,j}] = 1$ if $\{d_i, d_j\} \in E$; 0 otherwise. Matrix $\mathbf{M}_{D,L}$ is an incidence matrix where $[a_{i,j}] = 1$ if instance d_i is annotated with lexical term ℓ_j; 0 otherwise. We then set $\mathbf{M} \leftarrow \mathbf{M}_{D,D}\,\mathbf{M}_{D,L}$. We note that $\mathbf{M}$ captures the degree of connectivity of terms *via* the network. The second input to BuildOntology is the set of candidate parent nodes P. This set can also be thought of as the complete set of leaf nodes in $\mathcal{O}$. Only leaf nodes are considered as potential parents to terms not added to $\mathcal{O}$, thus ensureing a tree-structured ontology is synthesized. The third input to BuildOntology is $\mathcal{O}$ itself. On initialization $\mathcal{O}$ is empty and thus seeded with an artificial root node, termed "root," setting $P = \{\text{"root"}\}$. Upon completion of BuildOntology, $\mathcal{O}$ will contain all the lexical terms from $G_{D,L}$. The fourth input is the *percent* value. As the algorithm progresses, the largest value within a matrix is found (*maxval* at line 6). Applying *percent* establishes the range $[(1 - percent) \times maxval, maxval]$ and any columns in $\mathbf{M}$ that have a value falling within this range are selected (columns represent lexical terms).

```
BuildOntology($\mathbf{M}$, $P$, $\mathcal{O}$, $percent$)
 1  $p' \leftarrow \emptyset$, $C' \leftarrow \emptyset$, $maxval' \leftarrow 0$
 2  for each $p \in P$
 3        do
 4           $\hat{\mathbf{M}} \leftarrow \mathbf{M}/(P/\{p\})$
             /*$\mathbf{M}$ with all parent nodes in $P$ (except for $p$) removed[d]*/
 5           $\mathbf{M}' \leftarrow (\hat{\mathbf{M}}^{\top}\hat{\mathbf{M}})^2$
 6           $maxval \leftarrow$ Max($\mathbf{M}'/\{p\}$)
             /*returns the maximum value in $\mathbf{M}'$ without column $p$[d]*/
 7           $C \leftarrow$ FindCloseColumns($\mathbf{M}', maxval, percent$)
             /*returns columns with a value $> (1 - percent) \times maxval$*/
 8           if ($maxval > maxval'$)
 9              then
10                    $maxval' \leftarrow maxval$, $C' \leftarrow C$, $p' \leftarrow p$
11  AddChildren($\mathcal{O}, p', C'$)
    /*add the set of children $C'$ to the parent $p'$ in ontology $\mathcal{O}$*/
12  $P \leftarrow (P/\{p\}) \bigcup C'$
13  /*remove $p$, add children of $p$[d]*/
14  $\mathbf{M} \leftarrow \mathbf{M}/\{p'\}$
    /*remove column $p'$ from $M$[d]*/
15  if (all labels have been assigned)
16     then return $\mathcal{O}$
17     else  BuildOntology($\mathbf{M}, P, \mathcal{O}$)
```

The BuildOntology algorithm proceeds by applying the rules-of-thumb mentioned in Section 2.1 until $\mathcal{O}$ is completely constructed. The algorithm proceeds by choosing the parent p' with the greatest value and the associated set of children C' (Lines 2 through 10). To do this, we first remove all potential parents in set P excluding p from $\mathbf{M}$, which produces $\hat{\mathbf{M}}$(Line 4). Next we calculate $\mathbf{M}'$ by multiplying the transpose of $\hat{\mathbf{M}}$ by $\hat{\mathbf{M}}$ to get a symmetric matrix, then square $\mathbf{M}'$ to amplify the disparities among relationships in $\mathbf{M}'$ (Line 5). From $\mathbf{M}'$, we find the maximum value present and return it as the variable $maxval$ (Line 6). The set of children C of parent p are selected from $\mathbf{M}'$ if the column in $\mathbf{M}'$ has a value in the interval $[(1 - percent) \times maxval, maxval]$ (Line 7). This procedure is done

[d] We write set difference as A/B. To denote a similar matrix operation that removes a set of columns S from matrix $\mathbf{X}$, we write $\mathbf{X}/S$. For example, if $\mathbf{X} = [1\ 2\ 3\ 4]$, then $\mathbf{X}/\{2, 4\} = [1\ 3]$ (matrices are in bold text).

for each potential parent in P (Line 2) and the parent p with the greatest value is carried on through the algorithm (Lines 8 through 10). Upon completion of this loop, the selected parent p' and its set of children C' are added to $\mathcal{O}$ (Line 11). Once added to $\mathcal{O}$, p' is then removed from P and the children in C' are promoted to potential parents and appended to P (Line 12). p' is also removed from matrix $\mathbf{M}$ (Line 14). If all of the lexical terms have been assigned, then $\mathcal{O}$ is returned, otherwise we recursively apply this algorithm until all the terms are exhausted (Lines 7 through 17).

The complexity of this algorithm is $O(n^3)$ and performs well on the datasets tested in this study.

2.3. *Data*

2.3.1. *Microarray Coexpression Data*

The following 9 *S. cerevisiae* microarray datasets were downloaded from the Stanford Microarray Database (SMD): cell cycle (synchronized using CDC15 and alpha factor),[12] exposure to dithiothrietol (DTT),[13] diauxic shift,[14] exposure to gamma radiation,[15] sporulation,[16] heat stress,[13] starvation,[13] and exposure to H_2O_2.[13] All datasets were taken in their normalized form as log-transformed ratios and mapped to a unique yeast ORF ID. Spots not flagged as being problematic were averaged for both technical and biological replicates. Gene expression profiles with greater than 25% missing values for one gene across all conditions were removed, and any remaining expression profiles with missing values were imputed using KNNimpute.[17] For each gene expression profile per experiment, the difference between the maximum and minimum ratio value was calculated. Expression profiles that showed a difference less than 0.5 were removed.

The Pearson correlation coefficient was calculated using the expression profiles of each gene pair over in a given experiment. Significant correlations were determined through permutation testing, where the gene expression values within each condition are shuffled between genes. Correlation coefficients were calculated on the shuffled data to produce an empirical null distribution. The positively correlated gene pairs with a p-value < 0.01 were used and the other gene pairs were removed.

2.3.2. *Integrated Yeast Networks*

Two integrated yeast networks were used, namely YeastNet [6,18] and bioPIXIE (biological Process Inference from eXperimental Interaction

Evidence).[19] Both networks were built under a probabilistic framework using both yeast experimental data (*i.e.*, microarray, genetic interactions, protein interactions) and GO annotations. The value for a gene pair within the bioPIXIE network is a posterior probability from the Bayesian framework developed by Troyanskaya *et al.*,[20] while the values within YeastNet are weighted sums of log-likelihood scores as developed by Lee *et al.*.[6,18]

The bioPIXIE data[e] and YeastNet[f] data were downloaded June 2008. Any feature in the bioPIXIE or YeastNet integrated networks that did not match an ORF id in the *Saccharomyces* Genome Database gene registry were removed. Edges with values less than 0.3 in the bioPIXIE network and edges less than 1.0 in the YeastNet network were removed.

3. Results

3.1. *Ontology Properties*

We first demonstrate that the BuildOntology algorithm performs well at recapitulating relationships between GO terms. As inputs, the algorithm takes both gene network data as $\mathbf{M}_{D,D}$ and gene annotation data as $\mathbf{M}_{D,L}$. To test the ability of the algorithm to reconstruct known relationships while ignoring any influence from a gene network, the initialization of $\mathbf{M}$ as input to BuildOntology was changed from $\mathbf{M} \leftarrow \mathbf{M}_{D,D}\,\mathbf{M}_{D,L}$ to $\mathbf{M} \leftarrow \mathbf{M}_{D,L}$.

Using the Cytoscape[21] network visualization software and the MCODE graph clustering plug-in,[22] we extracted the top 20 subnetworks from both the bioPIXIE and YeastNet integrated gene networks. For each of these clusters, we built an ontology ignoring the gene network data. We consider a relationship between two terms in an ontology to be reasonably close if the shortest path length between these terms as defined in the GO is ≤ 3. This is based on the observation that when the shortest path between all pairwise GO biological process terms is calculated, more than 98% of the shortest paths are greater than a length of 3, with the average path length being ≈ 8.2. Roughly 46% of the GO terms across the tested ontologies were within a path length of 3 and the average path length between terms was 4.2 ($\sigma = 2.5$). In comparison, we then constructed DDOnts for the same 20 subnetworks in both the bioPIXIE and YeastNet networks while including the gene network relationships. For this set of ontologies, roughly 17% of the GO term connections were within a path length of 3 and the

[e] http://pixie.princeton.edu/pixie/

[f] http://www.yeastnet.org/

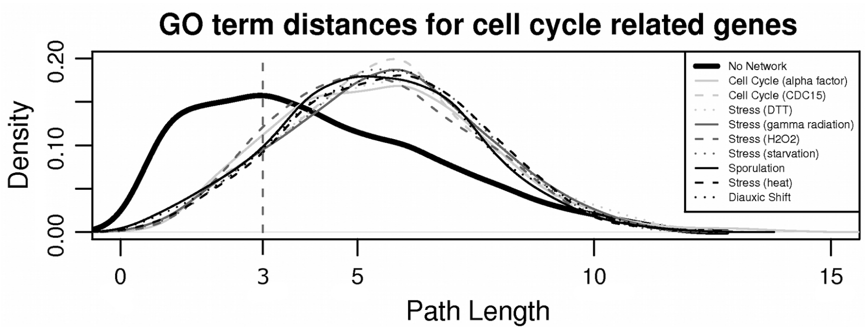

Figure 1. Distribution of the shortest path lengths between terms as measured in the Gene Ontology but assigned as a parent-child relationship in the Data-Driven Ontologies (DDOnt). All DDOnts were built from a common set of cell cycle related genes. The thick, black line represents an ontology built without gene network data, while the rest of the lines represent DDOnts constructed using microarray data. The change in distributions as compared to the ontology built without any network data was due solely to the gene-gene relationships in the microarray datasets. The path length cutoff of 3 is discussed in Section 3.1.

average path length was 5.4 ($\sigma = 2.1$). This comparison shows that the BuildOntology algorithm performed relatively well at recapitulating the term relationships defined in the GO, while including the network data places the terms further from their predefined positions in the GO revealing relationships directly influenced by the gene network data.

As a specific example of how the BuildOntology algorithm performs in relation to including or excluding network data, we first selected all genes annotated under the parent term *cell cycle* (GO:0007049) as listed in the SGD.[g] This set consisted of 386 genes annotated with 648 GO biological process terms. We then constructed an ontology from this set of terms and genes, while ignoring the network data. Next, we used 9 yeast microarray studies that cover a variety of experimental conditions to build gene networks (see Section 2 for datasets). Pearson correlation coefficients were calculated between gene expression profiles for each individual experiment and connections were drawn between a cell cycle gene pair if their coefficient was significant (see Section 2). This process results in 9 separate gene networks, which were used to build 9 DDOnts. The distribution of shortest path lengths for the 9 DDOnts and the ontology built without any gene network data are shown in Fig. 1. Including the gene network relationships

[g] checked on July, 2008

has a dramatic affect on the distribution of GO terms in the DDOnts.

3.2. *Data-Driven Ontologies for Discovery*

The bioPIXIE integrated gene network was used to demonstrate a specific application of a DDOnt to identify interesting relationships between biological processes. Using Cytoscape[21] with MCODE,[22] we extracted a series of subnetworks from bioPIXIE. An example subnetwork of 73 ORFs connected by 577 edges is shown in Fig. 2 **B**. Using this gene network and the 241 associated GO biological process annotations, we synthesized a DDOnt (Fig 2 **B**). The edge colors in the DDOnt correspond to the length of the shortest path between two terms within the GO. Edges colored grey are at a shortest path length of 1, 2, or 3, green 4, 5, or 6, red 7, 8, or 9, and blue at a length > 10. Areas of interest within the DDOnt consist of several terms with greater path lengths grouped together, which indicates potential novel relationships. The highlighted rosette in Fig. 2 **B** is an example. The parent node is *response to DNA damage stimulus* (GO:0006974). There are several examples of expected child terms, such as *double-strand break repair via break-induced replication* (GO:0000727), *DNA damage checkpoint* (GO:0000077), and *heteroduplex formation* (GO:0030491). Additionally, there are many terms that are connected to the parent node, but are located at a greater path length in the GO, such as *pre-replicative complex assembly* (GO:0006267), *regulation of chromatin assembly or disassembly* (GO:0001672), and *regulation of heterochromatin formation* (GO:0031445). Though these processes are located far apart in the GO, chromatin assembly and maintenance have long been linked to double-stranded breaks through recombination and cell cycle.[23] The reason these GO terms were put together in the arranged fashion can be seen from the network used in constructing the DDOnt. Nodes colored orange in Fig. 2 **A** are ORFs that are annotated with any of the GO terms contained in the rosette. These terms are highly connected and these relationships are taken into account in the constructing the DDOnt. If an ontology is built using the same algorithm, but excluding the gene network, this rosette is not formed and the GO terms are scattered throughout the ontology under parent nodes that are closer in path length.

We also applied the DDOnt to discover relationships from a set of predefined genes. We selected the 6 yeast ORFs annotated under the GO term *glucose transport* (GO:0015758). Next, we identified the immediate neighbor ORFs in the bioPIXIE network with an edge value > 0.5 and

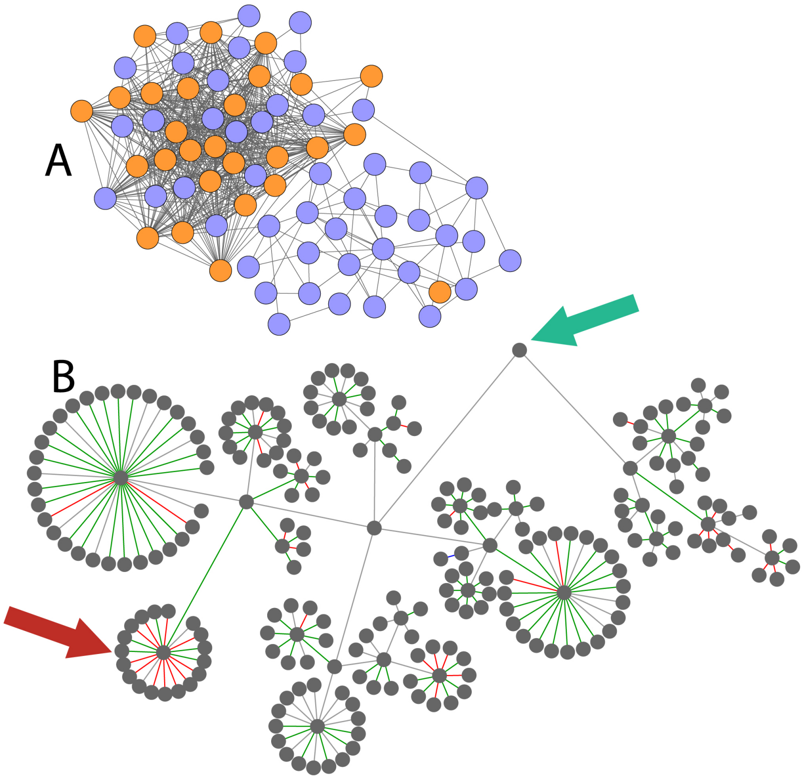

Figure 2. An example of a Data-Driven Ontology (DDOnt). The network of yeast genes (**A**) selected as a subnetwork of the bioPIXIE integrated gene network is used along with the Gene Ontology biological process annotations to synthesize the DDOnt (**B**) using the BuildOntology algorithm. The edge colors in **B** are discussed in Section 3.2. The root node in the ontology is designated by the teal arrow. The orange nodes in **A** are the genes that have been annotated with the rosette of terms in **B** designated by the red arrow. This rosette of terms are potentially novel relationships between biological processes where the parent node is *response to DNA damage stimulus* (GO:0006974).

constructed a network from this set of ORFs. This resulted in a gene network with 54 genes connected by 182 edges. This network and the associated GO terms were used to synthesize a DDOnt. Interestingly, the terms *proteolysis* (GO:0006508), *ubiquitin cycle* (GO:0006512), *phosphorylation* (GO:0016310), and *cell cycle phase* (GO:0022403) were placed under the parent term *monosaccharide transport* (GO:0015749). In the GO, these terms are located far apart, but their connections in the DDOnt reveal the links between cell cycle, proteolysis, phosphorylation, and glucose regulation. Of the set of six *glucose transport* (GO:0015758) annotated genes – RGT1, SKS1, MTH1, HXK2, HXK1, and GLK1 – four are kinases, thus supporting the relationship of phosphorylation. Of the immediate neighboring genes, GRR1, RSP5, and CDC34 are involved in ubiquitin-mediated protein degradation and RPN5 is a regulatory subunit of the 26S proteasome lid, thus supporting the proteolysis relationships. Lastly, several other immediate neighboring genes, CLN1 CLN2, CDC24, and GRR1 are all cell cycle related genes, thus supporting the cell cycle relationships. This is another example of how the construction of a DDOnt places GO terms together only because of the connections in a gene network. Although the relationships discussed are known, this example shows how potentially novel relationships can be discovered.

3.3. *Data-Driven Ontologies for Microarray Analysis*

Fluctuations in the behavior of sets of expressed genes under different experimental conditions is an underlying phenomenon studied through microarray experiments. Disentangling these effects is a major area of study. One property of DDOnts is to place terms closer to the root if they have high values. A term gets a high value from either being highly annotated or from the genes annotated with the term being highly connected. If a set of genes are fixed, thus fixing the GO term counts, and a DDOnt is built using two distinct gene networks, any differences between the two DDOnts is a result of the relationships in the gene networks exclusively. We can then find terms that are placed closer or farther from the root node to identify biological processes that are being affected through the relationships in gene networks.

We selected the 115 verified ORFs defined by the SGD to be involved in sporulation[g]. These 115 ORFs are annotated with 302 GO biological process terms. We first constructed a reference ontology using only the GO terms and ignoring any network data. Next, we constructed DDOnts for all 9 microarray datasets listed in Section 2. We then directly compared the 9 DDOnts to the reference ontology by measuring the distance of a GO term to the root. The GO terms where the distance from the root fluctuated most across the DDOnts were identified. Interestingly, mitosis, as represented by the terms *regulation of mitotic cell cycle* (GO:0007346), *chromosome segregation* (GO:0007059), and *mitosis* (GO:0007059), were

[g] checked on July, 2008

placed closer to the root node across all microarray DDOnts as compared to the reference ontology. An example of a general biological process that consistently moved farther from the root was conjugation; namely, *conjugation* (GO:0000746), *conjugation with cellular fusion* (GO:0000747), and *cellular morphogenesis during conjugation* (GO:0000767). In the reference ontology, these GO terms were all placed at a path length of 3 from the root, but when the microarray data were considered, these terms were consistently placed at a path length of 4 and 5 from the root. Both of these examples show that the relationships derived from gene expression profiles influence how a DDOnt is built, which reflects how biological processes are related in the experimental data.

4. Discussion

In this study we have demonstrated that ontologies (DDOnts) can be synthesized automatically from a gene network and a set of associated annotations. The presented algorithm is tractable and has been shown to effectively run on moderately sized data sets.

The scope of a constructed DDOnt is dependent on the input lexicon L and gene network. In this study we restricted L to be GO "biological process" terms, therefore the constructed DDOnts reflect biological processes. Similarly, the structure of a DDOnt is dependent on the input gene network. In the extreme cases where the network is either completely or sparsely connected, the gene network will have little effect on a DDOnt. The ideal input to the BuildOntology algorithm is a well annotated gene network with connections between genes forming function specific clusters.

The results of constructing DDOnts from informative input data demonstrate that gene networks do have a major effect on DDOnt construction and edges between terms not closely related in the GO may represent interesting or even novel relationships. We also show that DDOnts constructed from different gene networks yield different perspectives. In summary, DDOnts appears to be useful tools in discovering and understanding biological relationships. Future work will include refining the algorithm, incorporating edge type information, and relaxing structural constraints (*i.e.* expanding directed acyclic graphs beyond the current tree structure).

All results and DDOnt source code can be found at: `http://www.ddont.org`

Acknowledgments

We thank the Center for Genomics and Bioinformatics for computer support, and H. Tang, P. Radivojac, B. Eads, and S. Beason for comments.

References

1. A. Lesk, Ed., *Database Annotation in Molecular Biology: Principles and Practice*, (John Wiley & Sons, LTD, 2005).
2. A. J. Butte, I. S. Kohane, *Pac. Symp. Biocomput.*, 418 (2000).
3. L. Giot, J. S. Bader, C. Brouwer, A. Chaudhuri, B. Kuang, Y. Li, Y. L. Hao, C. E. Ooi, B. Godwin, E. Vitols, et al., *Science* **302**, 1727 (2003).
4. T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, *et al*, *PNAS USA* **98**, 2001 (4569-4574).
5. P. Uetz, L. Giot, G. Cagney, T. Mansfield, R. Judson, *et al.*, *Nature* **403**, 623 (2000).
6. I. Lee, S. V. Date, A. T. Adai, E. M. Marcotte, *Science* **306**, 1555 (2004).
7. C. L. Myers, D. Robson, A. Wible, M. A. Hibbs, C. Chiriac, C. L. Theesfeld, K. Dolinski, O. G. Troyanskaya, *Genome Biol.* **6**, R114 (2005).
8. I. Lee, B. Lehner, C. Crombie, W. Wong, A. Fraser, E. Marcotte, *Nat. Genet.* **40**, 181 (2008).
9. P. Simons, *Parts: A Study in Ontology (new edition)*, (Oxford University Press, USA, 2000).
10. S. Nirenburg, V. Raskin, *Ontological Semantics*, (MIT Press, 2004).
11. Gene Ontology Consortium, *Genome Res.* **11**, 1425 (2001).
12. P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, *et al.*, *Mol. Biol. Cell* **9**, 3273 (1998).
13. A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, *et al.*, *Mol. Biol. Cell* **11**, 4241 (2000).
14. M. J. Brauer, A. J. Saldanha, K. Dolinski, D. Botstein, *Mol. Biol. Cell* **16**, 2503 (2005).
15. A. P. Gasch, M. Huang, S. Metzner, D. Botstein, S. J. Elledge, P. O. Brown, *Mol. Biol. Cell* **12**, 2987 (2001).
16. S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, *et al.*, *Science* **282**, 699 (1998).
17. O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R. Altman, *Bioinformatics* **17**, 520 (2001).
18. I. Lee, Z. Li, E. M. Marcotte, *PLoS ONE* **2**, e988 (2007).
19. C. Myers, D. Robson, A. Wible, M. Hibbs, C. Chiriac, C. Theesfeld, K. Dolinski, O. Troyanskaya, *Genome Biology* **6**, R114 (2005).
20. O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, D. Botstein, *PNAS USA* **100**, 8348 (2003).
21. P. Shannon, A. Markiel, O. Ozier, N. Baliga, J. Wang, D. Ramage, N. Amin, B. Schwikowski, T. Ideker, *Genome Research* **13**, 2498 (2003).
22. G. Bader, C. Hogue, *BMC Bioinformatics* **4**, 2 (2003).
23. F. Paques, J. E. Haber, *Microbiol. Mol. Biol. Rev.* **63**, 349 (1999).