BIOINFORMATICS

Vol. 00 no. 00 2006
Pages 1–6

Inferring gene annotations in Gene Ontology from gene expression data

Xuerui Wang, David Kulp, Andrew McCallum

Department of Computer Science, 140 Governors Drive,
University of Massachusetts, Amherst, MA 01003, USA

ABSTRACT

Motivation: The Gene Ontology (GO) project develops a standard way to describe gene products in terms of their associated biological processes, cellular components and molecular functions. However, it is far from complete. Due to lacking biological knowledge or other technical difficulties, many gene products do not have GO annotations, and the annotations for many other gene products are not specific enough to be useful. Previous studies show that, to some extent, both sequence similarity and expression similarity indicate functional relationship. The ability to infer unknown annotations or to refine known (but not specific) annotations is fundamentally important in discovery of various new biological knowledge.

Results: In this paper, we focus on gene expression data and biological process annotations. The present work describes a new probabilistic framework based on the GO hierarchy where specific (biological process) annotations of genes are inferred from gene expression data. We apply the method to the gene expression data of [5] and the GO annotations from *Saccharomyces Genome Database* (SGD). We show the results in predicting (or refining) already known GO annotations under leave-1-out cross-validation. By comparing the results with previous non-generative models (such as kNN), we also conclude that genes having close annotations to each other in GO are not necessarily more similar in expression data than genes are further away. Additionally, we propose a new rank-based distance metric which is provably more robust than well-known metrics for gene similarity.

Availability: Available upon request.

Contact: xuerui@cs.umass.edu

1 INTRODUCTION

Rapid advancements in high-throughput methods for measuring levels of gene expression for tens of thousand of of genes in parallel have led to revolutionary changes in bioinformatics research. The abundance of available microarray data enables us to conduct large-scale inference using computational approaches regarding the roles of genes and relations between genes in cellular context.

Biological knowledge of gene and protein role in cells is being accumulated in an explosive fashion. To address the need for consistent descriptions of gene products, the Gene Ontology (GO) project [4], among other efforts using controlled vocabularies, organizes biological terms into three ontologies: cellular component, biological process, and molecular function. The GO annotations are very powerful in searching all available aggregated biological knowledge regarding specific topics, for example, finding all the gene products that are involved in bacterial protein synthesis when searching for new targets for antibiotics [4].

However, the Gene Ontology is far from complete. Many genes discussed in the literature do not have GO annotations due to lagged updates and many genes do not have specific enough annotations because of lack of related biological knowledge. There is a great need to automatically identify unknown annotations or refine known (but not specific) annotations, and thus to accumulate and discover biological knowledge in a more systematic way.

The huge amount of gene express data make it possible to conduct gene annotation inference computationally. Previous studies showed that genes with similar expression profiles (under some conditions) have similar annotations in GO [9, 7, 2, 10, 8, 17]. However, it is not yet clear whether, in the DAG structure of the Gene Ontology, genes that have close annotations to each other are more similar in expression than genes are further away. To computationally prove/disprove the validity of the above statement is also an objective of this paper.

The GO annotations are organized in a species-independent manner. The yeast genome has been well studied and thus its GO annotations are relatively accurate and complete. To make it easier to assess the methods, in this paper, we will focus on the *Saccharomyces Genome Database* (SGD) annotations.

Among the three GO ontologies (biological processes, cellular components and molecular functions), for our analysis of gene expression data, we only focus on Biological Process in this paper. Molecular Function and Cellular Location are conventionally assumed to be more amenable to sequence-based comparisons. For example, conserved motifs indicate how the protein interacts with other molecules, but not what a biological process suggests. Similarly, common molecular

function and cellular location are assumed to not necessarily be co-expressed. On the other hand, biological processes may have reasonably high-correlated gene expression level. Several previous studies [7, 2] also share this observation.

2 MATERIALS

In all experiments conducted in this paper, we use the gene expression data collection described in [5], available on-line at http://genome-www.stanford.edu/yeast_stress/data.shtml. The data represent the normalized, background-corrected log2 values of the Red/Green ratios measured on the DNA microarrays. All missing values in the dataset are estimated by us using KNNimpute [16] with 17 nearest neighbors under Euclidean distance metric. In total, we utilize the gene expression data of 6,152 genes under 173 experimental conditions.

We take the annotations in the SGD database distributed on February 15, 2006, and for consistent consideration, the DAG structure of the Gene Ontology is directly constructed from the ontology definitions on February 15, 2006. For simplicity, we do not distinguish "is_a" and "part_of" relationships. We assume that the annotations also follow the "true path rule", that is, if a gene is annotated at a node, it is implicitly annotated at all nodes in the pathway from this node all the way up to its top-level parent(s).

The term GO:0000004 (biological process unknown) was removed since it is not a real annotation. Many GO nodes have no genes or only a few genes associated with them. It is often the case that these nodes are not well studied. For simplicity, only the biological process GO terms associated with more than 10 genes in the above dataset are considered. In this way, we end up with 725 GO terms. In comparison with previous studies (e.g., [17]), we study the problem in a much larger scale (725 vs. 48 GO terms).

3 METHODS

We propose a Bayesian framework to conduct annotation inference in GO from gene expression data. Unlike the previous methods, the framework explicitly incorporates the knowledge about the structures of the gene taxonomy into consideration by assuming that genes that are closer in GO have similar expression data. Thus, we not only offer a new way to infer gene annotations in GO, but validate the assumption indirectly by comparing it with other models which do not take such an assumption. The DAG structures of GO and the fact that a gene may have multiple most specific annotations make it possible that there are generally multiple paths from the root to a gene's most specific annotation(s). Traditional data-driven clustering techniques do not respect either the GO structure or multiple annotations, and we need a model that take both into account.

In our framework, we associate a probabilistic distribution over gene expression profiles with each node in GO. On one hand, we want to learn a good model at each node to fit the associated data. One the other hand, to observe the GO hierarchical structures, the distribution distance between neighboring nodes should be small, more exactly, the distributions with neighboring nodes should be close. Instead of the ad-hoc distance metrics used in hierarchical clustering, a natural choice of distance measure between distributions is the Kullback-Leibler (KL) divergence, as used in the probabilistic abstraction hierarchies (PAH) model [14]. Unlike in our framework in which the "true path rule" of annotations is followed, the PAH model generates a data points only from *one leaf* node of a hierarchy. There is a trade-off between fitting individual distributions and making neighboring node close in distribution. We introduce a penalty factor λ for distance between distributions of neighboring nodes, and treat the distance penalty as a prior of the hierarchy. Notation used in the model is shown in Table 1.

Our model also share some insight of Bayesian Hierarchical Clustering (BHC) [6], which constructs a binary-tree-structured hierarchy using Bayesian hypothesis testing, without specifying a distance metric. The BHC model also generates data points at every node in the path from the most specific cluster to the root. However, any data point, in the BHC model, is only allowed to be associated with only *one leaf* cluster.

The PAH model mainly focuses on learn a better tree-structured hierarchy instead of incorporating an existing DAG-structured hierarchy like GO. For the purpose of inferring gene annotations, our framework is more flexible than the original PAH model in that we can ask multiple annotations at arbitrary levels instead of one annotation at the leaf node level. More importantly, our framework defines a probabilistic model of the data which allows us to ask questions like how well the GO structures match the gene expression data. It can also be used to calculate the probability of a gene belonging to any node in the DAG structures. When we have more evidence on other genes that share the similar expression profile of the test gene, we are able to infer more specific annotation(s) for it; otherwise, only less specific annotation(s) are possible.

A relatively general GO term might be the most specific annotation only for a few genes that may have very different gene expression profiles, as reported in [11]. By allowing a gene's expression value to be generated at all nodes in the pathway from its most specific annotation to the root, our model naturally avoids the danger of overfitting and focuses on all genes having corresponding annotations.

We learn the individual distributions at each node by maximum likelihood estimation and simultaneously make distributions at close node close (smaller KL divergence in this paper). So we want to maximize that $P(\text{GO}|\text{Data}) \propto P(\text{Data}|\text{GO})P(\text{GO})$.

Or equivalently,

Table 1. Notation used in this paper

SYMBOL	DESCRIPTION
G	number of genes that have annotations in GO
N	number of nodes in the GO hierarchy
D	number of experimental conditions
E	all edges in the GO hierarchy
M_n	model (distribution) associated with the n^{th} node in the GO hierarchy
G_n	number of genes having annotations in node n
S_n	set of nodes that are neighbors of node n
$\|S_n\|$	number of nodes that are neighbors of node n
e_g	expression of gene g under all experiments
a_g	node(s) with the most specific annotation(s) of gene g
g_n	number of genes at node n
λ	penalty factor for distribution distance
ρ	a distance measure between the distribution of two nodes (using symmetric KL *distance*)
GO	$M_1, M_2, ..., M_N$
Data	$e_1, e_2, ..., e_G$

$$\begin{aligned} J &= \log P(\text{Data}|\text{GO}) + \log P(\text{GO}) \\ &= \sum_{g=1}^{G} \log P(e_g|M_{a_g}) - \sum_{(i,j)\in E} \lambda\rho(M_i, M_j) \end{aligned}$$

Next we define the concrete distribution at each node M_i and the distance measure ρ. The dataset is a $G \times D$ matrix which encodes the expression level measurements of G different genes under D different experimental conditions. For our purpose, we need a continuous distribution over $\Re^D$. A basic choice is a multivariate Gaussian with a diagonal covariance matrix, $P(e_i|M_j) = \prod_{k=1}^{D} P(e_{ik}|M_{jk})$ where $M_{jk} = \text{Normal}(\mu_{jk}, \sigma_{jk}^2)$

We take ρ as symmetric KL divergence, that is, $\rho(M_i, M_j) = \frac{KL(M_i,M_j)+KL(M_j,M_i)}{2}$. It is not hard to show that

$$KL(M_i, M_j) = \sum_{k=1}^{D} (\log\sqrt{\frac{\sigma_{jk}^2}{\sigma_{ik}^2}} + \frac{\sigma_{ik}^4 + \sigma_{ik}^2\mu_{ik}^2 - \sigma_{ik}^2\sigma_{jk}^2 - 2\sigma_{ik}^2\mu_{ik}\mu_{jk} + \sigma_{ik}^2\mu_{jk}^2}{2\sigma_{ik}^2\sigma_{jk}^2})$$

$$\rho(M_i, M_j) = \frac{KL(M_i, M_j) + KL(M_j, M_i)}{2}$$

$$= \sum_{k=1}^{D} \left(\frac{(\sigma_{ik}^2 - \sigma_{jk}^2)^2 + (\sigma_{ik}^2 + \sigma_{jk}^2)(\mu_{ik} - \mu_{jk})^2}{4\sigma_{ik}^2\sigma_{jk}^2} \right)$$

If we assume that all Gaussian components have the equal variance σ^2, we can further simplify the above formula and get:

$\rho(M_i, M_j) = \frac{KL(M_i,M_j)+KL(M_j,M_i)}{2} = \sum_{k=1}^{D} \frac{(\mu_{ik}-\mu_{jk})^2}{2\sigma^2}$

It is very satisfying to see that J is concave. To estimate the parameters of the above model, we can simply use maximum likelihood estimation by taking derivatives of J with respect to the parameters of the distributions at each node, due to the convexity of $-J$. Iteratively, we can optimize the parameters of each M_i given other parameters in a round robin fashion until convergence. The convexity of $-J$ guarantees that the global maximum of J can be obtained. In each iteration, the computation only involves the genes at M_i, and ρ for M_i and its direct neighbors (parents and children). Parameters can be updated as follows:

$$\mu_{id} = \frac{\sum_{j=1}^{G_i} e_{jd} + \lambda \sum_{k\in S_i} (\sigma_{id}^2 + \sigma_{kd}^2)\mu_{kd}/2\sigma_{kd}^2}{G_i + \lambda \sum_{k\in S_i} (\sigma_{id}^2 + \sigma_{kd}^2)/2\sigma_{kd}^2}$$

$$\sigma_{id}^2 = \frac{\sqrt{G_i^2 + 2\lambda \sum_{k\in S_i} A_{id}/\sigma_{kd}^2} - G_i}{\lambda \sum_{k\in S_i} 1/\sigma_{kd}^2}$$

Where, $A_{id} = \lambda \sum_{k\in S_i} (\sigma_{kd}^2 + (\mu_{id} - \mu_{kd})^2)/2 + \sum_{j=1}^{G_i} (e_j d - \mu_{id})^2$. If we assume that all σ_{id}^2 equal an unknown constant, the above formulae could be substantially simplified.

$$\mu_{id} = \frac{\sum_{j=1}^{G_i} e_{jd} + \lambda \sum_{k\in S_i} \mu_{kd}}{G_i + \lambda|S_i|}$$

As shown above, all update equations are in close form, which makes the model inference highly efficient. In the non-parametric kNN framework, to predict a gene's annotation, its distances to all genes need to be calculated. In our new framework, no matter how many genes there are in the training set, only limited GO terms need to be visited for prediction. More importantly, annotation prediction can be conducted in an efficient batch fashion.

To address the uneven regularity in the hierarchy, we want to introduce a prior to make Bayesian prediction by getting a posterior distribution of a gene belonging to a node in the hierarchy. In the paper, the prior we adopt is the normalized counts of genes at each node in the hierarchy, with Laplace smoothing, that is, the prior probability that a gene belongs to a node i is $P(M_i) = \frac{G_i+1}{\sum_{i=1}^{N} G_i+N}$. To predict the annotations of unknown genes, we can easily get the posterior probability that it belongs to any node in the GO hierarchy, $P(M_i|e_j) \propto P(e_j|M_i)P(M_i)$. To make predictions, we can either select the nodes with highest posterior probability (topN in Table 2) or with posterior probabilities larger than a given threshold (Threshold in Table 2).

4 RELATED WORK

There have been quite a few research works in gene function inference using computational methods. For example, a

learning technique based on rough sets [9, 7] is used to learn simple rules for GO terms for genes of unknown function. Their model allows learning and classification of multiple biological process roles for each gene and can predict participation of genes in a biological process even though the genes of this class exhibit a wide variety of gene expression profiles including inverse co-regulation.

The rule-based method performed well on some temporal data which, however, are not nearly as common as stationary data. Many other works for stationary data include: [2] uses BLAST and mutual information to perform statistical tests and to provide accurate categorization; predicting human protein functions has been reported [8] using supervised learning methods (support vector machines); Supervised neural networks [10] and k nearest neighbors [17] are also presented in function annotation; some work in visualizing GO-based clustering of gene expression data [1]; and so on.

Among the above methods, the ksNN method [17] gives the best performance in inferring gene annotations in the literature. It first identifies k nearest neighbors of the gene to be annotated, and then calculate a representative score for each candidate GO term, which is a measurement of the taxonomy similarity between the GO term in consideration and the group of annotated nodes. Finally, m classes with highest representative scores are selected to be the gene's annotations. Two different taxonomy similarity measures are explored: PK-TS [12] and SB-TS [17]. In all of ksNN experiments we conduct for comparison in this paper, we use the experimental setting recommended in [17] (SB-TS, $k = 20$ and $m = 3$).

However, all the methods discussed here work for very limited number of GO terms, ranging from a few to dozens of nodes, corresponding to level 2 and 3 in the GO hierarchy. In this paper, we study the problem in a much larger scale, 725 GO terms and up to level 19 in the hierarchy.

5 EXPERIMENTAL RESULTS

Our method was evaluated on a data set provided by Gasch et al.[5]. Traditional accuracy measure is not appropriate here since a gene may have several correct annotations/predictions. Also, because of the existence of the ontology taxonomy, the basic precision/recall is not particularly good for our task, either. A prediction is possible to be partially correct, and taxonomy similarity (e.g., [12]) has to be incorporated into the measurement. We adopt the criteria in [11] shown as follows:

Let A_g be the set of annotations of a gene g and P_g be the corresponding set of predictions for g. We say $a \simeq p$ if annotation a and prediction p are on the same branch in the hierarchy; $a \succeq p$ if $a \simeq p$ and a is closer to the root than p in the hierarchy. Depth(a) is defined as the length of the shortest path from the root to a. We also define:

$$d(a,p) = \begin{cases} \text{Depth}(p)/\text{Depth}(a) & \text{if } p \succeq a \\ 0 & \text{otherwise} \end{cases}$$

Table 2. Annotation inference performance on SGD

Measures	Bayesian (topN)	Bayesian (Threshold)	ksNN
RA	0.7652	0.7768	0.7998
RP	0.5676	0.6165	0.6335
DA	0.4967	0.4944	0.5084
DP	0.4392	0.4696	0.4773

RA_g is the ratio of annotations of a gene which get predicted (similar to recall). $RA_g = |MA_g|/|A_g|$, where $MA_g = \{a \in A_g | p \in P_g, a \simeq p\}$.

RP_g is the ratio of correct predictions (similar to precision). $RP_g = |MP_g|/|P_g|$, where $MP_g = \{p \in P_g | a \in A_g, p \simeq a\}$.

DA_g gives the average relative depth of the best prediction for each annotation, indicating how detailed annotations are predicted. $DA_g = \sum_{a \in A_g} \max_{p \in P_g} d(a,p)/|MA_g|$.

DP_g gives the average relative depth of each prediction compared to the closet annotation, indicating the coherence of predictions. $DP_g = \sum_{p \in P_g} \max_{a \in A_g} d(a,p)/|MP_g|$.

Under leave-1-out cross-validation, we report average RA, RP, DA, and DP for all left out genes. The results are compared to the ksNN method [17] in Table 2.

As shown in Table 2, the results for the new framework are not as good as the ones for the ksNN method, but the results are still very competitive compared to other methods such as rule-based methods and neural networks. More importantly, our framework is much more computationally efficient than ksNN and other methods. From the results above, we can conclude that genes having close annotations to each other in GO are not necessarily more similar in expression data than genes are further away, thus possibly the gene expression data at a node can be modeled by a multi-modal distributions (e.g., mixture of Gaussians). We discuss possible improvements in detail in Section 7.

6 A NEW DISTANCE METRIC FOR GENE SIMILARITY

The ksNN method has given the best performance, however, there is still some space to improve within its framework. First, like all methods in the kNN family, it is very prone to noise. Particularly, we know gene expression data are indeed very noisy. Some ways to deal with the noisy essence of the data can be incorporated to improve the performance. Second, the uneven regularity in the GO hierarchy makes it difficult to find a universal value for k good for all test genes. For certain genes, there are simply not many close genes around them. Trying to get a fixed number of nearest neighbors not only increases the already expensive computation, but worsen the results by adding unnecessary far neighbors as well.

We can certainly adopt a scheme for variable k values according to the gene expression profile and the similarity to all genes in the dataset. For example, we only take the genes

Table 3. Annotation inference performance on SGD of the ksNN method using different distance metrics

Measures	Rank-based Correlation	Pearson Correlation
RA	0.8022	0.7998
RP	0.6414	0.6335
DA	0.5110	0.5084
DP	0.4804	0.4773

among the k nearest neighbors of gene g whose k' nearest neighbors include gene g, to make sure the selected neighbors are indeed close to the test gene. We can also consider the union of the k' nearest neighbors of the k nearest neighbor of gene g, to give the algorithm an opportunity to evaluate relatively not well studied GO terms. However, we will focus on developing new robust similarity measures in this section, which is more fundamental and generally applicable to other problems in the field of bioinformatics.

An extreme value in a gene expression profile may severely change the score of similarity. The rank-based Top Scoring Pair (TSP) classifier [15] is a new machine learning technique which works entirely on relative gene expression values. The TSP classifier is specifically designed to conduct pair-wise comparisons between any two gene expression levels, that is, to classify experiments (tissues). Inspired by the TSP classifier, we introduce a new distance metric for gene similarity as follows: 1) For each experimental condition, gene expression values are approximated by the ranks of them under that condition; 2) Calculate the Pearson correlation r between any two genes in terms of rank profiles (distance = $1 - r$). Obviously, this metric is (monotonical) transformation invariant, so it can be calculated directly on raw data, i.e., without normalization, thus avoids additional noise often introduced in normalization process. It is also more robust to noise due to the approximation introduced by ranks. Although it loses some detailed information about gene expression values, the benefit gained from reducing the effect of noise is more noticeable and important. As shown in Table 3, consistently better results are obtained using the new metric than the original Pearson correlation.

7 FUTURE WORK

In Section 5, we found out that genes having close annotations to each other in GO are not necessarily more similar in expression data than genes are further away. The unimodal Gaussian distribution assumed at each node is thus a drawback. The great efficiency of the framework make it possible to associate some multi-modal distributions at all nodes. In particular, mixture of Gaussians seems promising in that it clearly represents possible clusters of genes at each GO node while each cluster is still Gaussian distributed as usual.

To GO hierarchy is based on the biological knowledge people accumulated in a long history. A particular dataset is only relevant to some small part of the hierarchy. However, a data-driven hierarchy can be learned without the knowledge of the GO hierarchy. Using the techniques in [13], it is possible to leverage the learned hierarchy into the GO taxonomy in various ways, e.g., as a prior.

Purely probabilistic clustering methods do not always produce meaningful clusters. For example, one might wonder why two different regulated genes are placed in the same cluster, which might imply unknown knowledge about the genes, but, more often, are meaningless. By incorporating the knowledge from the GO structure, we want to ask how "good" a clustering is.

As described in [5], the experimental conditions in the dataset we used in this paper are far from independent. But almost all of the similarity measures treat them without difference. The simplest modification for that is to change the covariance matrix of Gaussians to be non-diagonal but it would become troublesome when mixture of Gaussians is used. Previous work in biclustering [3] has explicitly studied the dependency of experimental conditions when building gene clusters. To our knowledge, there is no gene ontology consistent biclustering algorithm available yet. We plan to introduce some similar idea into the frame work by examining the experiment similarity, thus some gene-dependent weighting scheme on experiments can be implemented.

For the purpose of reducing the influence of noise in gene expression data, rank-based metric in Section 6 is an right way to proceed. We plan to combine the ranking idea with clustering (particularly, in experiments) mentioned above to do better automatic gene annotation.

8 CONCLUSION

We have presented a new probabilistic framework based on the GO hierarchy where specific (biological process) annotations of genes are inferred from gene expression data. The efficiently of the model makes it possible to handle datasets in a larger scale, with competitive performance. In comparison with other models, we also verified that genes having close annotations to each other in GO are not necessarily more similar in expression data than genes are further away. The new rank-based gene similarity metric can substantially reduce the influence of noise, and help achieve better annotation inference performance.

ACKNOWLEDGMENT

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA). Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the

sponsor. We greatly thank Manjunatha N. Jagalur for helping prepare the data.

REFERENCES

[1]Boris Adryan and Reinhard Schuh. Gene-ontology-based clustering of gene expression data. *Bioinformatics*, 20:2851–2852, Nov 2004.

[2]Izabela Freire Goertzel Ben Goertzel, Cassio Pennachin, Moshe Looks, Murilo Saraiva de Queiroz, Francisco Prosdocimi, and Francisco Lobo. Inferring gene ontology category membership via cross-experiment gene expression data analysis. Biomind LLC, 2004.

[3]Yizong Cheng and George M. Church. Biclustering of expression data. In *American Association for Artificial Intelligence*, pages 93–103, 2000.

[4]The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[5]A.P. Gasch, P.T. Spellman, C.M. Kao, O. Carmel-Harel, M.B. Eisen, G. Storz, D. Botstein, and P.O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11(12):4241–4257, 2000.

[6]K.A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Twenty-second International Conference on Machine Learning*, 2005.

[7]Torgeir R. Hvidsten, Astrid Lægreid, and Jan Komorowski. Learning rule-based models of biological process from gene expression time profiles using gene ontology. *Bioinformatics*, 19:1116–1123, Jun 2003.

[8]L. J. Jensen, R. Gupta, H.-H. Stærfeldt, and S. Brunak. Prediction of human protein function according to gene ontology categories. *Bioinformatics*, 19:635–642, Mar 2003.

[9]A. Lægreid, T.R. Hvidsten, H. Midelfart, J. Komorowski, and A.K. Sandvik. Predicting gene ontology biological process from temporal gene expression patterns. *Genome Research*, 13:965–979, 2003.

[10]A Mateos, J Dopazo, R Jansen, Y Tu, M Gerstein, and G Stolovitzky. Systematic learning of gene functional classes from dna array expression data by using multilayer perceptrons. *Genome Research*, 12:1703–1715, 2002.

[11]Herman Midelfart, Astrid Lægreid, and Jan Komorowski. Classification of gene expression data in an ontology. In *Proc. of the Second International Symposium on Medical Data Analysis, LNCS 2199*, pages 186–194. Springer-Verlag, 2001.

[12]Viktor Pekar and Steffen Staab. Taxonomy learning: Factoring the structure of a taxonomy into a semantic classification decision. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 786–792, 2002.

[13]Suju Rajan, Kunal Punera, and Joydeep Ghosh. A maximum likelihood framework for integrating taxonomies. In *Proceeding of American Association for Artificial Intelligence*, 2005.

[14]E. Segal, D. Koller, and D. Ormoneit. Probabilistic abstraction hierarchies. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[15]Aik Choon Tan, Daniel Q. Naiman, Lei Xu, Raimond L. Winslow, and Donald Geman. Simple decision rules for classifying human cancers from gene expression profiles. *Bioinformatics*, 21:3896–3904, 2005.

[16]Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17:520–525, 2001.

[17]H. Yu, L. Gao, Kang Tu, and Z. Guo. Broadly predicting specific gene functions with expression similarity and taxonomy similarity. *Gene*, 352:75–81, 2005.