2009-6-3

Seeing the Forest for the Trees: Using the Gene Ontology to Restructure Hierarchical Clustering

BIOINFORMATICS ORIGINAL PAPER Vol. 25 no. 14 2009, pages 1789–1795
doi:10.1093/bioinformatics/btp327

Gene expression

Seeing the forest for the trees: using the Gene Ontology to restructure hierarchical clustering

Dikla Dotan-Cohen[1,*], Simon Kasif[2,3,4,5] and Avraham A. Melkman[1]

[1]Department of Computer Science, Ben-Gurion University, Beer Sheva, Israel 84105, [2]Department of Biomedical Engineering, [3]Center for Advanced Genomic Technology, [4]Bioinformatics Program, Boston University, MA 02215 and [5]Children's Hospital Boston, Harvard/MIT Program in Health Sciences and Technology, 300 Longwood Avenue, Boston, MA 02115, USA

Received on December 7, 2008; revised on April 28, 2009; accepted on May 15, 2009

Advance Access publication June 3, 2009

Associate Editor: David Rocke

ABSTRACT

Motivation: There is a growing interest in improving the cluster analysis of expression data by incorporating into it prior knowledge, such as the Gene Ontology (GO) annotations of genes, in order to improve the biological relevance of the clusters that are subjected to subsequent scrutiny. The structure of the GO is another source of background knowledge that can be exploited through the use of semantic similarity.

Results: We propose here a novel algorithm that integrates semantic similarities (derived from the ontology structure) into the procedure of deriving clusters from the dendrogram constructed during expression-based hierarchical clustering. Our approach can handle the multiple annotations, from different levels of the GO hierarchy, which most genes have. Moreover, it treats annotated and unannotated genes in a uniform manner. Consequently, the clusters obtained by our algorithm are characterized by significantly enriched annotations. In both cross-validation tests and when using an external index such as protein–protein interactions, our algorithm performs better than previous approaches. When applied to human cancer expression data, our algorithm identifies, among others, clusters of genes related to immune response and glucose metabolism. These clusters are also supported by protein–protein interaction data.

Contact: dotna@cs.bgu.ac.il

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Cluster analysis is an important data mining tool for investigating high-throughput biological data such as expression data. There is a growing interest in improving the cluster analysis of expression data by incorporating into it prior knowledge, given in terms of gene annotations such as the Gene Ontology (GO) annotations, in order to improve the biological relevance of the clusters that are subjected to further scrutiny.

Traditionally, the known annotations are utilized only as a second step, after the genes have been clustered according to their expression pattern. Only those clusters in which many genes are annotated with the same annotation (e.g. the same biological process), are then selected for further analysis (Buehler, 2004; Curtis, 2005; Doherty, 2006; Toronen, 2004; and others). Fang *et al*. (2006) took the opposite approach, first mapping the genes involved in the expression dataset to the GO hierarchy, and then looking only at those GO terms for which the mapped genes show high expression similarity. Khatri and Draghici (2005) compared 14 different tools for such secondary analysis.

In the past few years, several methods were introduced, that combine these two steps. We focus here on distance-based methods; for other approaches see the review of Bellazzi and Zupan (2007). Hanisch *et al*. (2002) constructed a distance function which combines information from expression data and the proximity of the proteins in a metabolic pathway network. Cheng *et al*. (2004) took a similar approach, though their graph is based on the GO structure. Huang and Pan (2006) proposed shrinking the distances between pairs of genes that share a common annotation, by a shrinkage parameter $r \leq 1$, and then clustering the genes using the new distance function. Speer *et al*. (2004) and Kustra and Zagdanski (2007) modified the similarity measure between two genes to be a linear combination of the similarity of their expression profiles and their functional similarity. Recently, we proposed to modify the popular hierarchical clustering method by 'snipping' the hierarchical tree to obtain clusters that are as consistent as possible with the known annotations (Dotan-Cohen *et al*., 2007). Most of these clustering methods utilize only the annotations provided by the GO. A further aspect of the ontology, its hierarchical structure, can be taken advantage of through the use of similarity measures between terms, derived from the ontology, (Jiang and Conrath, 1997; Lin, 1998; Lord *et al.*, 2003; Resnik, 1999; Schlicker *et al.*, 2006).

There are several motivating considerations for integrating semantic similarity measures into clustering methods. One is the potential enhancement in the performance of the clustering, a result of the good correlation between semantic similarity and gene co-expression (Wang *et al*., 2004). Another important motivation for the integration is that it enables the analysis to retain all the annotations of genes, including those that are higher up in the hierarchy. To elaborate, if in the analysis a gene is given all its annotations, all the way up to the root annotation, it makes no sense to scrutinize a cluster for annotations that are shared by the genes in the cluster. The simple reason is that by definition all genes are annotated with

*To whom correspondence should be addressed.

the most general GO term (the root of the hierarchy, e.g. the GO term 'biological process'). Therefore, algorithms that only consider having or not having some common annotation can only be useful when genes are annotated with GO terms that do not subsume one another.

To our knowledge Speer *et al*. (2004) and Kustra and Zagdanski (2007) are the only ones to propose algorithms that integrates semantic similarity directly into the clustering. Their methods cluster genes using a similarity measure that is a linear combination of semantic similarity and expression similarity between genes. However, genes that are currently unannotated are either excluded altogether, or are handled as exceptional cases in an *ad hoc* fashion.

In this article, we describe a novel clustering method which takes semantic similarities into account, and demonstrate that doing so improves the quality of the clustering. The method is a modification of the tree-snipping algorithm introduced in Dotan-Cohen *et al.* (2007). That algorithm takes as input a tree, the dendrogram as constructed in hierarchical clustering. Instead of constructing the clusters by the standard partitioning, which results from making a horizontal cut through the entire tree, the algorithm constructs clusters by snipping the tree—cutting selected edges at possibly different levels. The selection of the snips is guided by an objective function which aims to construct a partition that is maximally consistent with the partially available background knowledge. Specifically, each cluster of genes is given its majority annotation, and each gene whose annotation differs from its cluster annotation is called 'misclassified'. The objective of the algorithm is then to find a partition with the overall minimum number of misclassified genes.

The Minimum discrepancy algorithm introduced here also constructs clusters by snipping the dendrogram, and assigns each cluster an annotation. It departs from the method described above in assigning each gene a discrepancy score between 0 and 1, whose value is smaller the greater the semantic similarity between the gene's annotations and its cluster annotation is thus, for example, the penalty for assigning a gene which is annotated with 'mRNA capping' to a cluster labeled with 'mRNA cleavage' is less severe than the penalty for assigning to the same cluster a gene annotated with 'glycolysis'. The rationale for doing so is that 'mRNA capping' and 'mRNA cleavage' are both mRNA processing-related processes, which is reflected in the fact that they are semantically more similar to each other than are 'mRNA cleavage' and 'glycolysis'.

The aim of the Minimum discrepancy algorithm is to find a partition whose total discrepancy score is minimal. We demonstrate that the clustering capability of the Minimum discrepancy algorithm is indeed improved as compared with the original algorithm, as well as with two additional methods. The improvement manifests itself in the increased percentage of protein–protein interactions within the clusters, and in the heightened accuracy of cross-validation tests.

Another novel idea set forth here, and used in the testing phase of our algorithms, is to employ semantic similarity in the assessment of annotation predictions. We observe that there is a difference between a false prediction that is slightly wrong and one that is very wrong. Consider, for example, a cross-validation test in which a gene, whose actual annotation is withheld, is predicted to participate in 'DNA replication'. If the actual annotation of the gene is 'DNA recombination', the standard accuracy score will rate the prediction as completely wrong, without distinguishing it from the case where the actual annotation is 'reproduction'. Thus, the test makes no distinction between the second case, which is out of the ballpark, and the first case which is not far off in that both 'DNA replication' and 'DNA recombination' are DNA metabolic processes. Another disadvantage of such a cross-validation test is that a true prediction of a specific term and a true prediction of a general term both receive a score of 1, even though the latter prediction is much less informative. Consequently, we propose to evaluate a prediction using a *similarity weighted* accuracy score, equal to the semantic similarity between the actual annotation of the gene and the predicted process term.

2 METHODS

2.1 Expression similarity

For a given dataset, the expression similarity between two genes g_1 and g_2 was taken to be $(1+\rho(g_1, g_2))/2$, where $\rho(g_1, g_2)$ is the Pearson correlation between the expression profiles of g_1 and g_2.

2.2 Semantic similarity measures

2.2.1 Semantic similarity between two GO terms Many definitions of the semantic similarity between two terms in an ontology were introduced in the past few years. The definitions use $P(t)$, the probability of term t to occur, which is estimated as the number of gene products annotated with this term or any more specific term (child term) in the database, divided by the total number of annotated genes.

We adopt here the relevance measure of Schlicker *et al.* (2006):

$$\mathrm{Sim}_{\mathrm{Relevance}}(t_1,t_2)=\max_{t\in S(t_1,t_2)}\left(\left(\frac{2^{*}\log P(t)}{\log P(t_1)+\log P(t_2)}\right)*\left(1-P(t)\right)\right)$$

where $S(t_1,t_2)$ is the set of common ancestors of terms t_1 and t_2. This similarity measure is a modified version of the definition given by Lin (1998):

$$\mathrm{Sim}_{\mathrm{Lin}}(t_1,t_2)=\max_{t\in S(t_1,t_2)}\left(\frac{2^{*}\log P(t)}{\log P(t_1)+\log P(t_2)}\right)$$

The latter definition compares the information content of the two terms with that of their lowest common ancestor. It concentrates, therefore, on the closeness of the terms to their common ancestor, but it is insensitive to the level of detail, in the sense that two general terms are given the same similarity score as two specific terms provided the two pairs are equally close to their lowest common ancestors. In contrast, the relevance measure is sensitive also to the level of detail and takes on values in the interval [0, 1).

2.2.2 Semantic similarity between a gene and a GO term The discrepancy measure also requires the definition of the similarity between a GO term and a gene. A difficulty that has to be resolved is that a gene g is usually annotated with a list of GO terms, GO(g), because it participates in more than one biological process, or simply because a gene that participates in GO process t, also participates in every process that is more general than t. We adopt the usual approach of setting the similarity equal to the maximum possible similarity between the term and any of the annotations of the gene:

$$\mathrm{Sim}(t_1,g)=\max_{t_2\in GO(g)}\left(\mathrm{Sim}(t_1,t_2)\right)$$

2.2.3 Semantic similarity between two genes The semantic similarity between two genes is used in two competitor algorithms, the Semantic Similarity-based Shrinkage algorithm and the Linear Combination algorithm. Let GO(g_1) and GO(g_2) be the sets of GO terms with which g_1 and g_2 are annotated. The semantic similarity between two genes could be set equal to the maximum similarity between any pair of terms $t_1 \in$ GO(g_1) and $t_2 \in$ GO(g_2). This is the kind of definition adopted by Resnik (1999) for the similarity between words with multiple meanings in a lexical database, on the premise that a word has usually a single meaning in a given context. It was pointed out by Lord *et al.* (2003) that in the context of GO annotations

and expression data this definition may not be appropriate, because of the co-occurrence of several processes. We followed Schlicker *et al*. (2006) in defining the similarity between genes g_1 and g_2 as the average of the following row and column score:

$$\text{columnScore} = \frac{1}{|\text{GO}(g_2)|} \sum_{t_2 \in \text{GO}(g_2)} \max_{t_1 \in \text{GO}(g_1)} \big(\text{Sim}(t_1, t_2)\big)$$

$$\text{rowScore} = \frac{1}{|\text{GO}(g_1)|} \sum_{t_1 \in \text{GO}(g_1)} \max_{t_2 \in \text{GO}(g_2)} \big(\text{Sim}(t_1, t_2)\big)$$

2.3 Unannotated genes

Any clustering method that wishes to exploit the biological knowledge imparted by the annotations of the genes has to deal with the sizable percentage of genes that are currently unannotated. One option, adopted by Speer *et al*. (2004), is to exclude them from the clustering and thereby lose the information to be gained from their expression profiles. Another option, mentioned by Kustra and Zagdanski (2007), is to base the semantic similarity between an unannotated gene and an annotated one on the expression similarity between them. We propose instead to give such genes the most general annotation, the root of the ontology hierarchy (e.g. 'biological process', GO:0000004). This has the further advantage of not drawing a distinction between a gene whose annotation is completely unknown and one that has a known annotation that is not very specific.

2.4 Algorithms

2.4.1 Standard hierarchical clustering The first step in the widely used hierarchical agglomerative clustering algorithm is to create a binary tree, the dendrogram, each leaf of which corresponds to a different gene. The dendrogram is created by recursively and greedily joining the nodes that are closest to each other according to some criterion. Subsequently, k clusters are produced by making a horizontal cut through the dendrogram, right below the k - 1 highest merge nodes, thereby partitioning it into subtrees, and assigning all genes in a subtree to the same cluster.

2.4.2 Minimum misclassification clustering The standard hierarchical clustering method cannot take into account any additional information about genes, such as their known GO annotations. In a previous paper (Dotan-Cohen *et al*., 2007), we proposed therefore to partition the dendrogram in a biologically more meaningful manner by allowing partitions that snip selected edges at varying heights to take advantage of the partially available annotation of genes. We briefly describe this approach. Denote the set of annotations of a gene g by GO(g). In addition to generating a partition, we now wish to assign each resulting cluster c a label, *label*(c), from among a set of GO terms. A gene g belonging to a cluster c is said to be misclassified if it is not annotated with *label*(c). The problem is to find, among all possible K-partitions of the dendrogram and all possible labeling of the resulting clusters, one for which the total number of misclassified genes is minimized.

The dynamic programming Minimum misclassification algorithm solves the problem traversing the tree in a bottom-up fashion, while computing for each node v, functional label l and number of snips k ($0 \le k < K$) the value minMis(v,l,k), which equals the minimal number of misclassified leaves when node v is labeled with label l and it is permitted to snip k edges of the subtree rooted at node v, creating a $(k+1)$-partition. The computation uses a recursion formula for *minMis*(v,l,k) which considers three cases:

- Case 1: neither of the edges from the node v to its children left and right are snipped;
- Case 2: the edge from v to right is snipped;
- Case 3: the edge from v to left is snipped.

The minimum number of misclassified leaves for an optimal K-partitioning of the tree is the minimum of *minMis*(root,l,$K-1$) over all possible labels l. Once this number is computed, the appropriate snips can be found by the usual traceback, from the root of the tree down to the leaves.

2.4.3 Minimum discrepancy clustering In the Minimum misclassification clustering problem a gene was either misclassified or not. We propose here to soften this distinction by using instead a measure of the extent of misclassification of a gene g with respect to a label l, its discrepancy denoted *disc*(g,l), which varies between 0 and 1. Namely, we set *disc*(g,l) equal to the minimum of $(1-\text{Sim}_{\text{Relevance}}(t,l))$ over all t in GO(g). We emphasize that GO(g) of an unannotated gene contains simply the most general label, the root of the ontology.

The modified problem is now to find a partition P of the tree into a set of K clusters, and to find a labeling 'label' that assigns each cluster c in P a cluster label *label*(c), such that the following measure, the similarity discrepancy, is minimized:

$$SD\ (P, label) = \sum_{c \in P} \sum_{g \in c} \text{disc}\big(g, label(c)\big)$$

The main recursive formula at the heart of the Minimum discrepancy algorithm for computing *SD* is similar to the one described for the Minimum misclassification algorithm. It differs only in that the initial value of *minMis*(v,l,k) for each leaf v is *disc*(v,l). The resulting algorithm runs in $O(nLK^2)$ time, and uses $O(nLK)$ space where n is the number of genes, L is the total number of possible labels and K is the requested number of snips.

2.4.4 The balance algorithm Any k-partition P of the dendrogram is obtained by snipping an edge to a child at $k-1$ nodes, while at each of these *excluded* nodes the other child is made into a child of the parent of the node; call these nodes snipping nodes, and denote their set by $S(P)$. Denote by $h(v)$ the height of a node v in the hierarchical tree. Since the latter is constructed using complete linkage, for internal nodes $h(v)$ is the largest distance between the expression patterns of two genes clustered in the subtree rooted at v. Denote by $H(P)$, the sum of the heights of the snipping nodes,

$$H(P) = \sum_{v \in S(P)} h(v)$$

Alternatively $H(P)$ can be defined as the difference between the sum of all node heights in the tree and the sum of the node heights in the clusters induced by P. Intuitively, it is desirable to maximize $H(P)$ (i.e. to minimize the sum of the node heights in the clusters induced by P). By itself, this maximum is achieved when $S(P)$ consists of the $k-1$ nodes that have the largest possible heights. Hence, the resulting partition is the same as the one obtained by standard hierarchical clustering. Another, possibly conflicting, desirable goal is to minimize the similarity discrepancy. Consequently, we develop an algorithm that strikes a balance between minimizing *SD* and maximizing H, by minimizing the function

$$\lambda\ SD(P, label) - (1-\lambda)H(P)$$

For a fixed λ, the algorithm computes for all v,l,k, *minDiscH*(v,l,k), the minimum value of $\lambda SD(P,\text{label}) - (1-\lambda)H(P)$ for any k-partition of the subtree $T(v)$ with the constraint that the component of v is labeled l. The recurrence formula is given in the Supplementary Material. The initial value of *minDiscH*(v,l,k) for a leaf v is $\lambda * disc(v,l)$. This modification does not change the time- or space-complexity of the algorithm. The parameter λ is used to balance between the two goals: when $\lambda = 1$, the balance algorithm becomes the Minimum discrepancy algorithm described previously; and, when $\lambda = 0$ the algorithm becomes the standard hierarchical clustering, as discussed above. The experiments presented in Section 3 demonstrate that the best results are achieved when λ equals 1.

2.4.5 Comparison algorithms: shrinkage and linear combination We compared the performance of our algorithms (Minimum discrepancy and balance) with that of four other algorithms. The first, the Standard algorithm, is the standard hierarchical clustering algorithm described previously, which ignores the annotation information. Two of the other algorithms incorporate the available annotation information about the genes by first modifying the distance function $d(i,j)$ between genes i and j derived from the expression

patterns of genes i and j, and then applying standard hierarchical clustering. In all cases the hierarchical tree was constructed using complete linkage.

The first of these was proposed by Kustra and Zagdanski (2007), and takes the overall similarity between two genes to be a fixed linear combination of the similarity of their expression profiles and their semantic similarity; we use the average of the two similarity measures, and call it the Linear Combination algorithm. In case at least one of the genes is unannotated their overall similarity is set equal to the similarity of their expression profiles, as suggested by Kustra and Zagdanski (2007). We adopt this suggestion, instead of assigning the root annotation to the unannotated gene, because the semantic similarity between an annotated and an unannotated gene is zero, and consequently the overall similarity between these genes would be small, even if they are tightly co-expressed.

The third algorithm is our Minimum misclassification algorithm, described previously.

The fourth is the algorithm of Huang and Pan (2006). Their algorithm is given a shrinkage parameter $r \leq 1$, and for any two genes i and j that share an annotation the distance between them is reduced to $r^*d(i,j)$. Although it is possible to integrate semantic similarity into this framework, we have found that it provides marginal improvements in this setting. This issue is discussed further in the Supplementary Material.

2.5 Performance evaluation

The performance of the proposed clustering algorithm and the competitor algorithms was evaluated in two different ways. The first test used as a validation metric the average percentage of gene pairs in the clusters which are known to interact according to the protein–protein interaction (PPI) database (cf. Kustra and Zagdanski, 2007). The second consisted of a series of cross-validation tests in which a biological process (BP) annotation was determined for each cluster, and that annotation was predicted to be the annotation of those genes belonging to the cluster whose annotation had been withheld (cf. Huang and Pan, 2006). Note that the algorithms under discussion are not designed for functional prediction *per se*, so that the cross-validation served only to evaluate the quality of the clusterings.

2.5.1 Expression data, yeast cell cycle As the test dataset for the evaluation of the algorithms we used the well-known yeast time-series dataset of Spellman *et al*. (1998), which consists of the expression values of *Saccharomyces cerevisiae* genes at different time points along the cell cycle. Missing values in the log-transformed data were replaced with zeros. GO-BP annotations were downloaded from http://www.geneontology.org (Revision: 1.1310).

2.5.2 Validation with protein–protein interaction data, yeast For each cluster of n genes, we computed the percentage of the number of interactions found in the PPI data, downloaded from DIP (version March 2007) available at http://dip.doe-mbi.ucla.edu/, relative to the total number of *possible* interactions, $\binom{n}{2}$.

The PPI figure of merit was then taken as the average percentage of PPI pairs over all clusters. We included in this analysis only those genes that were in the PPI data found to have at least five interactions with other proteins, in order to reduce the effect of missing PPI data as in Kustra and Zagdanski (2007).

2.5.3 Determination of the cluster annotation in cross-validation Once the genes are clustered into distinct clusters, and in order to use these clusters for functional prediction cross-validation tests, it is required to label each cluster with a GO term. This label is predicted to be the annotation of those genes in the cluster whose actual annotation was withheld. In practical usages, a cluster might be labeled with more than one label, but for comparison purposes it is preferable to predict a single annotation for each tested gene, to assure that the number of predictions in each of the tested methods is equal. Since in some of the tested procedures, cluster labeling is not part of the clustering itself, we separated the labeling phase from the clustering procedure. All clusters obtained by each of the clustering procedures were labeled with the one GO term that was statistically most enriched among the terms of the annotated genes in the cluster, according to the hypergeometric distribution. An alternative approach is to assign each cluster the majority label of its annotated genes. This approach was used in Huang and Pan (2006). However, it is impossible to use such a labeling when the clustered genes are annotated with GO terms from different levels of the GO tree.

2.5.4 Accuracy of cross-validation prediction In the standard cross-validation test, a gene is predicted to participate in the process with which the cluster it belongs to has been annotated. The accuracy score of the clustering is defined to be the ratio of the number of predictions that are true to the total number of predictions. In the following we call this the *strict* accuracy measure.

We propose and examine also an alternative figure of merit for evaluating the quality of the predictions, the *similarity weighted* (*sw*-) accuracy measure. For the latter measure, we assign each prediction a sw-accuracy score equal to the semantic similarity between the actual annotation of the gene and the predicted process term. Note first of all that a false prediction can still receive a high sw-accuracy score if the predicted term is close to the actual one. Furthermore, a true prediction can have a low sw-accuracy score if the predicted term is a general one, since the relevance score rates $0 < \text{relevance}(t,t) < 1$ for any GO term t. The sw-accuracy of the test is defined to be the average sw-accuracy score over all predictions.

2.5.5 Setup of 5-fold cross-validation test In V-fold cross-validation test, the base dataset is randomly partitioned into V equally sized subsets. The prediction accuracy is calculated as the average accuracy over V tests. In each test, the annotations of all genes from one subset are hidden. Only the annotations of the other genes are considered in the clustering procedure, and the subset of genes with hidden annotations are then predicted according to the cluster to which they are assigned. We used $V = 5$.

2.5.6 First experiment The first experiment described in Section 3, involved all the 273 GO-BP annotations that are not too specific in that at least 50 genes are known to participate in it, and all those 1628 genes that (i) have a functional annotation that is more specific than the most general one, e.g. 'biological-process', and that (ii) are known to interact with at least five different proteins. A 5-fold cross-validation test was performed, as described above. Thus, in each of the five subtests, only 80% of the genes had known labels (the remaining 20% of the genes were treated as unannotated genes). For each of the clusters, we also noted the percentage of pairs that are present in the PPI data, for the PPI-percentage test. The results described in Section 3 are the average over the five sub-tests.

2.5.7 Second experiment In the second experiment described in Section 3, the 5-fold cross-validation test was repeated 10 times, each time with a different base dataset of genes consisting of all genes participating in 10 processes chosen at random from among the 215 processes in which at least 50 and at most 200 genes participate; the latter restriction was imposed to prevent bias due to very general or too specific process terms. The number of genes in the base dataset varied from one repetition to the next, the smallest being 474 and the largest 769. The same clusters were used for PPI-percentage tests as well. The results described in Section 3 are the average over the 5*10 sub-experiments.

2.5.8 Expression data, Human breast cancer To further test our algorithm, we used the breast cancer human expression data of van 't Veer *et al.* (2002) which consists of the expression values of human genes at primary breast tumors of 117 young patients. The samples of 20 of these patients, who have BRCA1 or BRCA2 mutation, were excluded from further analysis. For the expression of each gene, we used the log-transformed ratio of the disease sample expression to a reference mRNA pool expression

measured by Agilent two-colored cDNA platform. The expression values of multiple probes representative of the same gene were summarized by averaging.

Our input data consequently includes the expression values of 10 503 genes for 97 breast tumor samples.

2.5.9 Validation with protein–protein interaction data, human PPI data for *Homo sapiens* was downloaded from two sources: the Human Protein Reference Database (release 7) available at http://www.hprd.org and the Molecular INTeraction database (version April 2008) available at http://mint.bio.uniroma2.it.

After mapping the interactions to gene symbols, our full list of interactions consists of 41 757 interactions. Of these interactions, 31 765 occur between 6953 of the genes tested in the human expression data described above. For each cluster of co-expressed genes, we included for further analysis only those n genes that appear in the PPI data, and computed the percentage of the number of interactions found in the PPI data relative to the total number of possible interactions, $\binom{n}{2}$. The average of this quantity over all clusters was taken to be the overall PPI percentage.

3 RESULTS

3.1 First experiment: using all the GO terms

Figure 1 compares the results of the Minimum discrepancy algorithm, the Standard algorithm and the Linear combination algorithm. Figure 1A presents the percentage of PPI pairs out of all possible pairs, present on average in the different clusterings. The results of the cross-validation tests are shown in Figure 1B. The accuracy measure is the ratio of the number of true predictions to the total number of predictions, averaged over the five subtests.

The Minimum discrepancy algorithm performs better than all its competitors in the cross-validation test, while on the PPI percentage test it is slightly outperformed only by the Linear combination method. One should bear in mind, however, that the main purpose of the PPI percentage test is to qualitatively support the coherency of the clustering. Indeed, since protein–protein interactions are more correlated with semantic similarities than with any other linear combination of semantic similarity and expression similarity, as shown in Kustra and Zagdanski (2007), a clustering using only semantic similarity would achieve the highest scores in this test. A further interesting finding was that the GO terms found to be most enriched in the clusters obtained by the Standard clustering, are significantly more specific than the ones obtained by Minimum discrepancy algorithm, which in turn are more specific than the terms found to be most enriched in the Linear combination clustering. Observe that a more specific prediction is a priori less likely to be true. What's worse, even when the predicted annotation is semantically very similar to the actual, more general, annotation of the gene it still receives a score of zero, so that no points are awarded for being close to the mark.

In order to take into account the semantic similarity between the predicted annotation and the actual one in the cross-validation results, we evaluated the sw-accuracy scores, see Section 2, for the same clusters and their annotations.

The results are summarized in Figure 1C. The scores for all methods are higher, due to the fact that many 'false' predictions that were scored '0' by the strict accuracy score are scored higher than '0' by the sw-accuracy score. However, the improvement of the standard clustering is the most significant: it now performs better than the Linear combination method, whose performances improve

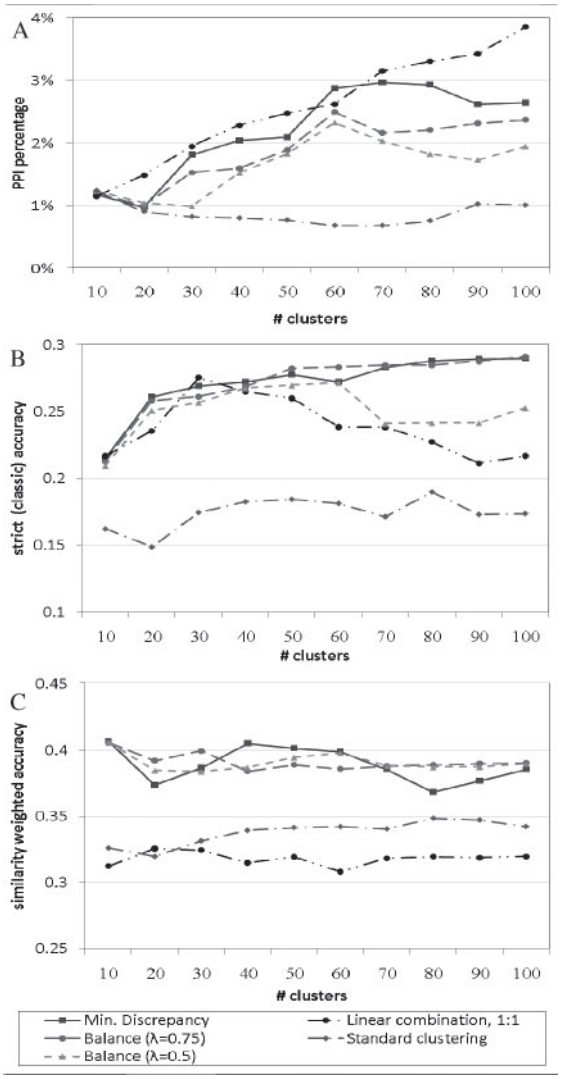

Fig. 1. Clustering performances as a function of the number of clusters. (**A**). Overall PPI percentage in the obtained clusters. (**B**) The 5-fold cross-validation tests, strict accuracy. (C) The 5-fold cross-validation tests, sw-accuracy.

the least. This observation can be explained by the fact that the Linear combination procedure tends to predict GO annotations that are very general. We believe that the poor results of this method can be attributed to the fact that it treats the uncharacterized genes differently than the characterized ones, whereas these are precisely the genes at the focus of the cross-validation tests. The distribution of the accuracy among the different clusters is also shown in the Supplementary Material.

3.2 Second experiment: 10 Go terms

As stated before, some recently published algorithms, such as our Minimum misclassification algorithm (Dotan-Cohen *et al*., 2007) and the Shrinkage algorithm (Huang and Pan, 2006) which only consider having or not having some specific annotation, can be applied only when the set of possible annotations is restricted. Therefore, in the second experiment we used 10 randomly

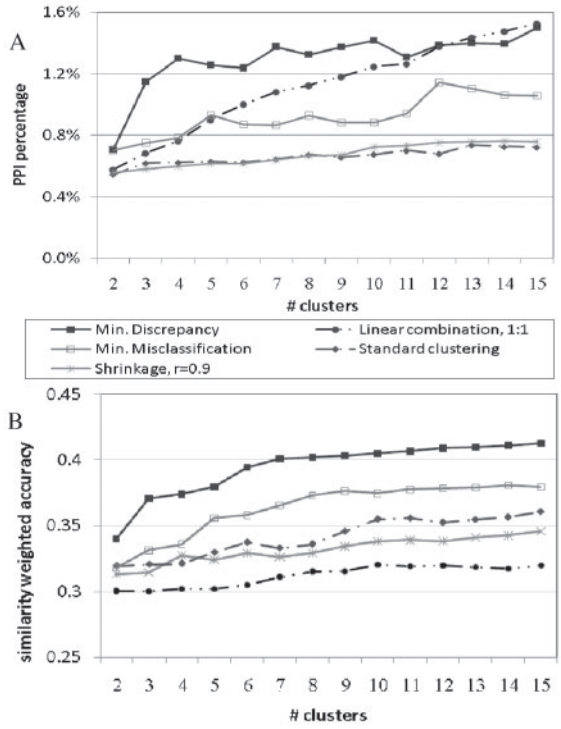

Fig. 2. Clustering performances as a function of the number of clusters: average accuracy over 10 experiments. (**A**) Overall PPI percentage in the obtained clusters. (**B**) The 5-fold cross-validation tests, sw-accuracy.

chosen functional annotations to serve as labels to the genes, see Section 2. Figure 2 presents the performances of the five tested algorithms (Minimum misclassification, Minimum discrepancy, Shrinkage, Standard and Linear combination). Figure 2A presents the correlation between the obtained clusters and the PPI data. In Figure 2B, sw-accuracy is measured. The unannotated genes are assigned to the term that is statistically the most enriched one among the annotated genes in the cluster. In both tests, the Minimum discrepancy algorithm performs better than the Minimum misclassification algorithm, the Shrinkage algorithms and the Standard clustering. It is notable that both the Shrinkage algorithm and the Linear combination method are outperformed by the Standard clustering in the cross-validation tests, although in terms of the PPI percentage test this situation is reversed.

3.3 Applying the Minimum discrepancy algorithm to human breast cancer data

As a practical application, the Minimum discrepancy algorithm was used to analyze the breast cancer expression data of van't Veer *et al.* (2002). The 10 503 genes were grouped into 100 clusters, using as labels all 529 GO-BP annotations that are not too specific, in that at least 50 genes participate in the process.

Notably, the tightest observed cluster (average pairwise Pearson correlation of 0.75), was labeled with 'glucose metabolism'. Unusual activity of glucose pathways has been observed in the past in connection with the Warburg effect, the generation of oxygen by fermentation of glucose in cancer cells (Gatenby and Gillies, 2004). However, this small pathway usually avoids detection in standard clustering, so that the prominence of the cluster is suggestive of the sensitivity of our procedure. Moreover, of the seven genes belonging to the cluster, six are known to participate in 'glucose metabolism', while the seventh, SLC4A1, is currently annotated only with processes that are very distant in the Gene Ontology. However, the protein encoded by this gene is known to influence the metabolism of glucose in the RBC (Weber *et al*., 2004).

Another potential benefit of our method is exemplified by three clusters labeled with 'immune response'. There is increasing recognition that gene modules are potentially useful for prognostic and diagnostic application in cancer. In particular, an immune module was shown useful in predicting survival for breast cancer patients (Teschendorff *et al*., 2007). However, the current definition of gene modules is generally believed to be incomplete. We believe that our approach will enable researchers to extend modules with co-regulated genes and perhaps achieve better prognostic accuracy, but this research is beyond the scope of this particular paper. It may be worth mentioning that our 'immune response' clusters include genes like CD68, IL10RA and CD33 which currently lack functional annotation or are annotated only to processes that are very distant in the Gene Ontology, although these three genes are known to participate in the immune response (Crocker *et al.*, 2007; Qi *et al.*, 2005; Simmons and Seed, 1988).

To assess the quality of the clusters by independent means, we used PPI data. In comparison to standard hierarchical clustering, the average PPI percentage of the clusters obtained by our method is improved by 150%, from 0.2% to 0.5%. The complete list of clusters and the genes included in each cluster is available at http://www.cs.bgu.ac.il/∼dotna/treeSnipping2.html.

4 DISCUSSION

Semantic similarity measures extract a significant part of the biological knowledge captured by the GO. Integrating semantic similarity into the clustering of gene expression data should therefore yield clusters that are biologically more germane. Two different perspectives give this intuition a quantitative footing and demonstrate the superiority of the clusters generated by the Minimum discrepancy algorithm presented here. The first, asks how many of the potential interactions between genes in a cluster actually appear in the database of PPI. The rationale here is that interacting genes are strongly constrained, as co-expression might be essential to sustain the normal function of cells and tissues (Yona *et al.*, 2007). The second, exploits the positive correlation between semantic similarity and gene expression (Wang *et al*., 2004) and examines the cohesiveness of the clusters from an annotation perspective: using a cross-validation methodology, we measure the success of predicting the annotation of a gene from the dominant annotation of the cluster it belongs to. Note that the primary purpose of our algorithm is to cluster genes, rather than making functional predictions for which it may be better to employ algorithms such as Kustra *et al*. (2006).

A further advantage of integrating semantic similarities into the clustering procedure is that it enables the retention and exploitation of multiple annotations, as is the case for the majority of genes. In contrast, several algorithms, such as Dotan-Cohen *et al*. (2007) or Huang and Pan (2006), ask only whether a gene does or does not have a given annotation. Such algorithms can only be used when none of the terms that genes are annotated with is a descendant term of another. Still, different clustering methods differ also in the benefit

they derive from the integration of background knowledge. We have found that the Minimum discrepancy algorithm, which integrates semantic similarities into a snipping algorithm, results in clusterings that are superior to the competitor approaches examined here. A possible explanation for this is that in the snipping methodology the semantic similarity is brought into play after the genes have already been clustered by expression similarity. Consequently, the semantic similarity plays a significant role only in borderline cases. Another notable advantage of the Minimum discrepancy algorithm is that it treats the annotations in a uniform and consistent manner. For example, a gene without known annotations is given the root annotation of 'biological process'. This is in contrast to other approaches in which such a gene is handled as an exceptional case, whereas a gene that has a very non-specific annotation, such as 'metabolic process', is treated in the same manner as a gene with a very specific annotation.

We have also examined a modification of the Minimum discrepancy algorithm, called balance, which takes the edge lengths of the dendrogram into consideration, by minimizing a linear combination of the total discrepancy score and the sum of the node heights in the clusters. Note that minimizing just the sum of the node heights is what the traditional hierarchical clustering approach does. We find, however, that the resulting clusterings are not superior, and at times definitely inferior. We surmise that this is because the utility of semantic similarities lies principally in disambiguating borderline cases, where it is difficult to determine which cluster a gene belongs to on the basis of expression data only; in such cases relying on semantic similarity alone appears to be the best strategy.

ACKNOWLEDGEMENTS

We thank Yair Mazor and Shahar Bar for their help.

Funding: Lynne and William Frankel Center for Computer Science; the Paul Ivanier center for robotics research and production. NIH grant # R01 HG003367-01A1

Conflict of Interest: none declared.

REFERENCES

Bellazzi,R., and Zupan,B (2007) Towards knowledge-based gene expression data mining. *J. Biomed. Inform.*, **6**, 787–802.

Buehler,E.C. *et al.* (2004) The CRASSS plug-in for integrating annotation data with hierarchical clustering results. *Bioinformatics*, **20**, 3266–3269.

Cheng,J. *et al*. (2004) A knowledge-based clustering algorithm driven by Gene Ontology. *J. Biopharm. Stat.*, **14**, 687–700.

Crocker,P.R. *et al.* (2007) Siglecs and their roles in the immune system. *Nat. Rev. Immunol.*, **7**, 255–266.

Curtis,R.K. *et al*. (2005) Pathways to the analysis of microarray data. *Trends Biotechnol.*, **23**, 429—435.

Doherty,J.M. *et al*. GOurmet: a tool for quantitative comparison and visualization of gene expression profiles based on gene ontology (GO) distributions. *BMC Bioinformatics*, **7**, 151.

Dotan-Cohen,D. *et al.* (2007) Hierarchical tree snipping: clustering guided by prior knowledge. *Bioinformatics*, **23**, 3335–3342.

Fang,Z. *et al.* (2006) Knowledge guided analysis of microarray data. *J. Biomed. Inform.*, **39**, 401–411.

Gatenby,R.A. and Gillies,R.J. (2004) Why do cancers have high aerobic glycolysis? *Nat. Rev. Cancer*, **4**, 891–899.

Hanisch,D. *et al*.. (2002) Co-clustering of biological networks and gene expression data. *Bioinformatics*, **18**, 145–154.

Huang,D. and Pan,W. (2006) Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Bioinformatics*, **22**, 1259–1268.

Jiang,J.J. and Conrath,D.W. (1997) Semantic similarity based on corpus statistics and lexical taxonomy. In *Procedings of the International Conference on Research in Computational Linguistics, ROCLING X*, Taiwan.

Khatri,P., and Draghici,S. (2005) Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, **21**, 3587–3595.

Kustra,R. *et al.* (2006) A factor analysis model for functional genomics. *BMC Bioinformatics*, **7**, 216.

Kustra,R. and Zagdanski,A. (2007) Data-fusion in clustering microarray data: balancing discovery and interpretability. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **1**, 1.

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

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

Qi,Z.M. *et al.* (2005) Polymorphism of the mouse gene for the interleukin 10 receptor alpha chain (Il10ra) and its association with the autoimmune phenotype. *Immunogenetics*, **57**, 697–702.

Resnik,P. (1999) Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res.*, **11**, 95–130.

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

Simmons,D. and Seed,B. (1988) Isolation of a cDNA encoding CD33, a differentiation antigen of myeloid progenitor cells. *J. Immunol.*, **141**, 2797–2800.

Speer,N. *et al.* (2004) A memetic co-clustering algorithm for gene expression profiles and biological annotation. *CIBCB*, **2**, 1631–1638 .

Spellman,P.T. *et al*. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell.*, **9**, 3273–3297.

Teschendorff,A.E. *et al*. (2007) An immune response gene expression module identifies a good prognosis subtype in estrogen receptor negative breast cancer. *Genome Biol.*, **8**, R157.

Toronen,P. (2004). Selection of informative clusters from hierarchical cluster tree with gene classes. *BMC Bioinformatics*, **5**, 32.

van 't Veer,L.J. *et al*. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Wang,H. *et al*. (2004) Gene expression correlation and Gene Ontology-based similarity: an assessment of quantitative relationships. In *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2004)*, San Diego, CA, pp. 25–31.

Weber,R.E. *et al*. (2004) Modulation of red cell glycolysis: interactions between vertebrate hemoglobins and cytoplasmic domains of band 3 red cell membrane proteins. *Am. J. Physiol. Regul. Integr. Comp Physiol.*, **287**, 454–464.

Yona,G. *et al.* (2007) Comparing algorithms for clustering of expression data - how to assess gene clusters. In *Computational Systems Biology*. Humana press, Totowa, NJ.