Discovering Relations among GO-annotated Clusters by Graph Kernel Methods*

I. Zoppis[1], D. Merico[1], M. Antoniotti[1], B. Mishra[2], and G. Mauri[1]

[1] Dipartimento di Informatica, Sistemistica e Comunicazione
Universitá degli Studi di Milano Bicocca
Via Bicocca degli Arcimboldi 8, U7, I-20126 Milano, ITALY
[2] Bioinformatics Group, Courant Institute of Mathematical Sciences
New York University, 715 Broadway, New York, NY, 10003, USA

Abstract. The biological interpretation of large-scale gene expression data is one of the challenges in current bioinformatics. The state-of-the-art approach is to perform clustering and then compute a functional characterization via enrichments by Gene Ontology terms [1]. To better assist the interpretation of results, it may be useful to establish connections among different clusters. This machine learning step is sometimes termed *cluster meta-analysis*, and several approaches have already been proposed; in particular, they usually rely on enrichments based on flat lists of GO terms. However, GO terms are organized in taxonomical graphs, whose structure should be taken into account when performing enrichment studies. To tackle this problem, we propose a kernel approach that can exploit such structured graphical nature. Finally, we compare our approach against a specific flat list method by analyzing the cdc15-subset of the well known Spellman's Yeast Cell Cycle dataset [2].

1 Introduction

The biological interpretation of large-scale gene expression data is one of the challenges in bioinformatics [3]. The state-of-the-art approach is to perform clustering, in order to group together genes with similar expression profiles across experiments; then, in order to provide a functional characterization [4], enrichments of Gene Ontology [1] terms are computed for each cluster. In fact, it is expected that groups of genes which perform a common function also behave in a coordinated fashion. In addition, since different processes may contribute to a common function, they could be associated to the same cluster (for instance, both DNA repair and cell cycle arrest genes are both induced after DNA damage). To further assist the result interpretation, it may be useful to establish connections between different clusters; that is especially useful if they refer to different tissues, or if they have been produced according to different experimental techniques. This machine learning task can be termed *cluster meta-analysis*, and several approaches have already been proposed even if they usually rely on

* This work has been supported by EC "Marie Curie" grant MIRG-CT-2005-031140.

comparing enrichments built out of flat lists of GO terms [5]. However, Gene Ontology terms are not mutually independent, but organized according to a graph of taxonomical relations, and thus a flat list comparison approach may fail to exploit this specific structured nature.

A particularly interesting problem of cluster meta-analysis arises when studying time series expression data coming from microarray experiments (as in [6, 7]). In this case, clustering the entire profiles may not allow some temporally localized relationships among genes to be detected. It may happen indeed that some processes are associated (that is, their genes behave in a coordinate fashion) only within a limited sequence of time-steps. Splitting microarray gene expression time series into shorter time-windows which can be clustered in separated groups has been proposed in [8–10] and implemented in the GOALIE system [11]. The GOALIE system provides a number of visualizers for navigating the set of relationships induced by the GO enrichments of the clustering of the time-windows. In order to compare these time-windows clusters, which are obtained in a manner very similar to the one implemented in GOALIE, it is necessary to take into account the different graph structures of their GO enrichments (*GO window graph*), we propose to use a *kernel measure* of similarity. Because of their theoretical potential as well as wide-range of applicability, kernel methods [12, 13] have proven to be among the currently most successful learning algorithms. These methods work by embedding the data into a new *features* space and then looking for relations between the data in that space. In this way complex relations can be simplified and then used, for example, for classification, regression, clustering, etc.
The paper is organized as follows: in Section 2 we give a brief overview of the Kernel Methods. In Section 3, we more formally address the underlying biological problem and apply a valid kernel function to measure the similarities among the objects of our model. In Section 4, we discuss the numerical results of our evaluation experiments and finally in Section 5 we conclude and discuss some directions for future work.

2 Kernel Functions and Graph Kernel

Kernel methods have been successful in solving different problems in machine learning. The idea behind these approaches is to map implicitly the input data (i.e. training set) into a new feature (Hilbert) space $\mathcal{F}$ in order to find there some suitable hypothesis: in this way complex relations in the input space can be simplified and more easily discovered. The feature map Φ in question is defined by a kernel function k which allows to compute the inner product in $\mathcal{F}$ using only objects of the input space, hence without carrying out the map Φ. This is sometimes referred as the *kernel trick*.

Definition 1 (Kernel function). *A kernel is a function $K : X \times X \rightarrow \mathbb{R}$ capable of representing through $\Phi : X \rightarrow \mathcal{F}$ the inner product of $\mathcal{F}$ i.e.*

$$K(x, y) =< \Phi(x), \Phi(y) > \tag{1}$$

To assure that such equivalence exists a kernel must satisfy Mercer's Theorem. Hence, under certain conditions (for instance semi-definiteness of K), by fixing a kernel one can assure the existence of a mapping Φ and a Hilbert space $\mathcal{F}$ for which (1) holds. These functions can be interpreted as similarity measures of data objects into a (generally non linearly related) feature space, therefore, given K one can always induce a (non Euclidean) distance $d : X \times X \rightarrow \mathbb{R}$ such that:

$$d^2(x,y) = K(x,x) + K(y,y) - 2K(x,y) \tag{2}$$

While working with spaces whose objects are more structured, one may choose one of the many suitable kernels that exploit the underlying structure; e.g., for the set $\mathcal{G}$ of undirected labeled graphs [3] a suitable measure for the similarity between G_1 and G_2 counts the number of matching labeled random walks. These measures were proposed by different authors (see for instance [14–17]). Given a match, being obtained by comparing the label values associated to a pair of nodes (or edges), the (kernel) similarity between two random walks is then the product of the similarity values corresponding to the nodes and edges encountered along the walk. The kernel value of two graphs is then the sum over the kernel values of all pair of walks within these two graphs:

$$k_{graph}(G_1, G_2) = \sum_{walk_1 \in G_1} \sum_{walk_2 \in G_2} k_{walk}(walk_1, walk_2) \tag{3}$$

An elegant approach to construct such a similarity measure uses the direct product graph [15]:

Definition 2 (Direct product of two labeled graphs). *Given two labeled graphs $G_1 = (V, E), G_2 = (W, F)$ the direct product is denoted by $G_1 \times G_2$. The vertex set $V_\times$ and edge set $E_\times$ of this direct product are respectively defined as:*

$$\begin{aligned} V_\times(G_1 \times G_2) &= \{(v_1, w_1) \in V \times W : label(v_1) = label(w_1)\} \\ E_\times(G_1 \times G_2) &= \{((v_1, w_1), (v_2, w_2)) \in V_\times^2(G_1 \times G_2) : \\ &\qquad (v_1, v_2) \in E \\ &\qquad \wedge (w_1, w_2) \in F \\ &\qquad \wedge label(v_1, v_2) = label(w_1, w_2)\} \end{aligned} \tag{4}$$

[3] Here we use the following notation: a graph $G = (V, E)$ consists of a finite set of n vertices V denoted by $\{v_1, v_2, \ldots, v_n\}$, and a set of directed (possibly, weighted) edges $E \subseteq V \times V$. A walk w on G is a sequence of indices $(w_1, w_2, \ldots, w_{t+1})$ where $(v_{w_i}, v_{w_{i+1}}) \in E$ for all $1 \leq i \leq t$. A random walk is a walk where $\mathbb{P}(w_{i+1}|w_1, \ldots, w_i) = \mathbb{P}(w_{i+1}|w_i) = A_{w_i, w_{i+1}}$, i.e., the probability at w_i of picking w_{i+1} is directly proportional to the weight of the edge $(v_{w_i}, v_{w_{i+1}})$.

The nodes and edges $G_1 \times G_2$ have the same labels as the corresponding nodes and edges in G_1 and G_2. Based on this definition the *Random Walk Kernel* is then defined as follows.

Definition 3 (Random Walk Kernel).

$$k_\times(G_1, G_2) = \sum_{i,j=1}^{|V_\times|} \left[\sum_{n=0}^{\infty} \lambda_n A_\times^n \right]_{i,j} \tag{5}$$

where $A_\times$ is the adjacency matrix of the product graph:

$$[A_\times]_{i,j} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E_\times \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Therefore the sum in (5) converges for a suitable choice of $\lambda_0, \lambda_1, \lambda_2 \ldots$ [15]. In this paper, as in [18], we compute the random walk kernel only for walks up to a predetermined length.

3 Method

Groups of genes may behave in a coordinate manner, but over a period of time, such coordination may be confined within some limited interval. Therefore, the usual clustering of the entire profiles, while useful (as exploited in [6, 7]) may not allow some relationships among genes to be detected. To overcome this problem, it has already been suggested to split the time-series into shorter, partially-overlapping time-windows [8–10]. In this section we design *GO Window Graphs*, a kind of graphs where each node represents the functional enrichment of a cluster of genes in a specific time-window, and then we provide patterns of relations across time, by assembling the adjacent cluster pairs possessing minimal dissimilarity. Apart from the time-windows breakdown, our approach is in the vein of other works, such as [19, 20].

3.1 GO Graph Model

In more details, our analysis is conducted on the sets $S_i = \{\mathcal{C}_{i,u} : u = 1, \ldots, N\}$ of N gene clusters obtained at step $i \in \{1, \ldots, M\}$ by splitting each time series in M time-window intervals. For each of these clusters we compute a labeled GO graph by attributing for each node v its GO term value (accessed as $\mathsf{label}_{GO}(v)$) and its enrichment (log) p-value (accessed as $\mathsf{label}_p(v)$).

3.2 A kernel for GO Window Graphs

The graph kernel defined in section 2 is designed for discrete attributes; in that case two labeled nodes match whenever they share the same label values (i.e.

their attribute). In our case labels are almost never completely identical since they contain the (*log*) p-values of the enrichment computations. More precisely, we apply the distance (2) induced from a specific kernel function that measures (dis)similarity between the associated functional processes, in order to determine a relationship $\mathcal{L} \subseteq S_1 \times S_2 \times ... \times S_M$ which can be used to link similar GO Window Graphs (i.e. clusters). One such suitable function can be constructed

- by considering in (4) a match between two vertex v_1 and v_2 if $\mathsf{label}_{GO}(v_1) = \mathsf{label}_{GO}(v_2)$, and
- by modifying as in [18] the adjacency matrix (6).

We have the following:

Definition 4. *Given two graphs $G_1 = (V, E)$ and $G_2 = (W, F)$ and two walks $walk_1 = (v_1, v_2, \ldots, v_n) \in G_1$ and $walk_2 = (v_1, v_2, \ldots, v_n) \in G_2$, with $v_i \in V$, $w_i \in W$. The walk kernel is defined as*

$$k_{walk}(walk_1, walk_2) = k_{step}((v_i, v_j), (w_i, w_j)) \tag{7}$$

for each i and j.

The random kernel is still the sum over all kernel on pairs of walk as in [18] and it can be computed with following adjacency:

$$[A_\times]_{((v_i,w_i),(v_j,w_j))} = \begin{cases} k_{step}((v_i, v_j), (w_i, w_j)) & \text{if } ((v_i, v_j), (w_i, w_j)) \in E_\times \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

with $E_\times = E_\times(G_1 \times G_2)$ and $(v_i, v_j) \in E$ and $(w_i, w_j) \in F$.

Our step kernel has a simpler formulation having the goal of comparing only the (log) p-values of the original node, the destination nodes and their respective GO terms. More formally:

Definition 5 (Step kernel for GO graphs). *For $i = 1, \ldots, n-1$, the* step kernel *is defined as*

$$\begin{aligned} & k_{step}((v_i, v_j), (w_i, w_j)) \\ & = k_{nodepv}(v_i, w_i) * k_{nodeterm}(v_i, w_i) * k_{nodepv}(v_j, w_j) * k_{nodeterm}(v_j, w_j) \end{aligned} \tag{9}$$

where for k_{nodepv} we use the Brownian bridge kernel [21]

$$k_{nodepv}(x, x') = \max(0, c - |\mathsf{label}_p(x) - \mathsf{label}_p(x')|) \tag{10}$$

and for the kernel $k_{nodeterm}$ on the GO terms, a Dirac function Kernel:

$$k_{nodeterm}(x, x') = \begin{cases} 1 & \text{if } \mathsf{label}_{GO}(x) = \mathsf{label}_{GO}(x') \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Now, by defining $U = \bigcup_i S_i$ and by following the same proof scheme of [18], we can show that

$$k(\mathcal{C}_{i,u}, \mathcal{C}_{i+1,v}) = \sum_{j=1}^{|V_\times|} \sum_{n=0}^{\infty} \lambda_n A_\times^n \tag{12}$$

is still a valid kernel on $U \times U$. Thus, we have the following lemma.

Lemma 1. *Let k be defined as in (12). Then it is a positive definite kernel function.*

Proof. The node kernel is a Brownian bridge kernel that is known to be positive definite [21]. Since pointwise multiplication preserves positive definiteness, step kernel is consequently positive definite. By fixing now $\tilde{k}^j_{walk}$, for all these pairs of walks of length j only and zero otherwise [16] we have that $\tilde{k}^j_{walk}$ is a tensor product of step kernels for walks which is zero extended to the whole set of pair of walks, hence is positive definite. Again, K_{walk} is a sum over all $\tilde{k}^j_{walk}$ and is valid as well. The modified random walk kernel follows being a convolution kernel proven to be positive definite. Hence (12) can measure the similarity among objects in U and specifically can perform the similarity for each $x \in S_i$ and $y \in S_{i+1}$.

Since each kernel induces a distance (2), here we take $d_i : U \times U \to \mathbb{R}$:

$$d(\mathcal{C}_{i,u}, \mathcal{C}_{i+1,v})^2 = k(\mathcal{C}_{i,u}, \mathcal{C}_{i,u}) + k(\mathcal{C}_{i+1,v}, \mathcal{C}_{i+1,v}) - 2k(\mathcal{C}_{i,u}, \mathcal{C}_{i+1,v}) \tag{13}$$

Therefore it becomes quite natural to link the cluster $\mathcal{C}_{i,u}$ at time i with the cluster $\mathcal{C}_{i+1,v}$ at time $i+1$ on the base of the minimal distance value i.e.

$$\mathcal{C}_{i,u} \sim \mathcal{C}_{i+1,v} \text{iff } d(\mathcal{C}_{i,u}, \mathcal{C}_{i+1,v}) = \min_{\mathcal{C}_{i,m} \in S_i, \mathcal{C}_{i+1,n} \in S_{i+1}} d(\mathcal{C}_{i,m}, \mathcal{C}_{i+1,n}) \tag{14}$$

4 Numerical Results

The purpose of the following analysis is mainly to compare the results of our application against a specific flat list approach. The n-tuples in $\mathcal{L} \subseteq S_1 \times S_2 \times \ldots \times S_M$ are expected to contain clusters with functional homogeneity among each other and maximum separation of functional annotations across clusters of other n-tuples. Therefore we conducted numerical evaluations to assess two quality indexes: (I) maximum density with minimum diversity within a cluster and (II) maximum separation between clusters. This reflects one of the main approaches in quality validation tests for a clustering technique. In general, DNA microarray expression data-sets are grouped with the expectation that genes with similar functional features group together. In order to fulfill this expectation, we have applied the indexes for cohesiveness from [22] while performing the clustering at each "window-time interval". We briefly report all these indexes to provide a better understanding of our results.

The probability of selecting a gene associated to a functional group (identified by a certain GO term) i within a cluster r can be estimated knowing the total number of genes in r i.e. $p_{ir} = \frac{n_i}{n_r}$ and $\sum p_{ir} = 1$. One can model the functional cohesiveness within a cluster using Shannon's information theory. Higher value of cohesiveness of a cluster is measured by a high degree of certainty that the genes in a cluster belongs to a functional group. Hence, the cohesiveness of a cluster is its information content:

$$CC = -\sum_{i=1}^{k} p_{ir} \log_2(p_{ir}) \tag{15}$$

In our application the relation $\mathcal{L} \subseteq S_1 \times S_2 \times ... \times S_M$ is discovered step by step by evaluating the kernel-induced distance between pair of clusters $\mathcal{C}_{i,u}, \mathcal{C}_{i+1,v}$ for each time-window interval i. That is, discharging the interval steps we can consider this distance as a way to group together genes of the respective clusters in the pair, where these genes share the same functional processes in an ideal case. It seems, indeed quite natural to consider the cluster cohesiveness index CC (15) when a cluster is defined as $\mathcal{C}_{i,u} \cup \mathcal{C}_{i+1,v}$. Therefore, the total cluster cohesiveness can be defined as

$$TCC = -\sum_{r=1}^{r=m} \sum_{i=1}^{k} p_{ir} \log_2(p_{ir}) \tag{16}$$

The functional separation across different clusters can be measured by estimating the probability b_{ir} of selecting a gene of functional group i in cluster r among all genes belonging to the functional group i, i.e. $b_{ir} = \frac{n_{ir}}{N_i}$ where n_{ir} is the total number of genes of functional group i in cluster r, N_i the total number of genes in the behavioral group i and $\sum b_{ir} = 1$. The information content of a functional group i in all the clusters reflects the specificity of the functional group and thereby indicates the separation property, more formally:

$$GC = -\sum_{r=1}^{m} b_{ir} \log_2(b_{ir}), \tag{17}$$

while the total cluster separation can be defined as

$$TGC = -\sum_{i=1}^{k} \sum_{r=1}^{m} b_{ir} \log_2(b_{ir}) \tag{18}$$

For a simple flat list approach we first removed from each cluster those terms whose p-values was below a detection threshold and then applied as in (13) the Jaccard distance index:

Jaccard	1-2	2-3	Total
TCC	253.99	333.65	587.64
TGC	2134.9	2513.4	4648.30
Kernel	1-2	2-3	Total
TCC	276.41	303.45	579.86
TGC	2298.3	2341.3	4639.6

Table 1. The comparison of the cluster cohesiveness and separation indexes for connected clusters between time windows 1 and 2, and between time windows 2 and 3. The Kernel induced distance produces better results, although it is more expensive to compute.

$$J(X, Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|} \tag{19}$$

for each X and $Y \in \bigcup_i S_i$.

4.1 Preliminary Analysis of the Yeast Cell Cycle Data-set

The Spellman's Yeast Cell Cycle data-set [2] includes three main experiments: cdc15, alpha-factor, elutriation (where the names correspond to the three different methods employed for cell synchronization). We have analyzed only the cdc15 subset, which is 18 time-points long.

GO annotations of *S. cerevisae* genes have been downloaded from the SGD database (`http://www.yeastgenome.org`). The GO DAG has been derived from R package GO 1.14.1. Functional Enrichment p-values have been calculated according to the hypergeometric distribution approximating Fisher's exact test (a standard in existing resources, such as [23–25]). GO terms annotating less than 4 genes of all genes from the experiment ("universe-set") have been excluded from the analysis. The p-value of GO terms with less than 5 genes in sample has been arbitrarily set to 1 (not relevant at all).

The cdc15 data-set was divided into 5 time-windows, with 5 time-steps each, with one overlapping time point. We have computed 15 clusters for each of the first three time-windows using a standard k-means algorithm. Then we identified the "most similar" pairs of adjacent clusters according to (i) the Kernel induced distance and (ii) Jaccard coefficient.

Then, we have merged associated pairs clusters, obtaining new clusters. For these, we have computed cluster cohesiveness and separation indexes, *TCC* and *TGC*, which have been already described elsewhere in the paper. These display a superior performance for the Kernel induced distance over the Jaccard coefficient. Table 1 shows some of these comparisons between the Jaccard coefficient and the Kernel induced distance according to *TCC* and *TGC*.

In addition, we also ran two qualitative benchmarks to test our Kernel and Jaccard performance. We have traced connections between clusters according to:

translation and ribosome subgroup	
4.70e-13	cytosolic large ribosomal subunit (sensu Eukaryota)
4.30e-11	translation
7.43e-11	structural constituent of ribosome
5.40e-07	cytosolic small ribosomal subunit (sensu Eukaryota)
1.22e-06	translational elongation
1.04e-4	ribosomal small subunit assembly and maintenance
2.60e-3	ribosome
cell wall, plasma membrane and vescicular compartments sub-group	
1.04e-04	vacuolar transport
1.77e-03	endoplasmic reticulum
2.70e-03	transporter activity
3.40e-03	integral to membrane

Table 2. c8w1 enrichment reports these terms, ranked by their p-value.

- the absolute value of the intersection between the sets of genes,
- a manual curation.

For each connection found by Jaccard and Kernel, we specified whether it had been found or not according to the other methods.

Again, the superiority of the Kernel approach is generally confirmed, although a substantial disparity occurs between time window 1 to 2 and time window 2 to 3 couplings, with the second one displaying a greater performance difference.

5 Biological Results

The division of the Spellman Yeast Cell Cycle Data into windows of 5 time-points, yields time windows roughly corresponding to two phases each:

- window 1 (1-5): G1, S (and partially G2)
- window 2 (5-9): G2, M (and partially G1)
- window 3 (9-13):G1, S

The discrepancy of window 1 and 2 w.r.t. holding exactly two phases is probably due to synchronization, which alters the regularity in the very initial time-points (see Figure 1 for details).

The demonstration of this statement is provided by the chart in Figure 1, showing the normalized expression levels of a few marker genes.

To provide an example of the results yielded by our method, we consider the maximal-similarity connections found among three adjacent clusters, respectively belonging to time-window 1, 2 and 3 (they will be termed c8w1, c10w2, c13w3 in Tables (2), (3), and (4)).

Therefore, a robust core of terms can be identified: (1) protein synthesis by the cytoplasmic ribosome, (2) glycolysis and glyconeogenesis, and (3) cell wall, plasma membrane and vescicular compartment.

Actually, c8w1 does not include glycolysis and glyconeogenesis genes, which happen to be in a different cluster (c11w1); however, if we compare the profiles of c11w1 and c8w1, we can see that they are quite correlated, displaying a slightly

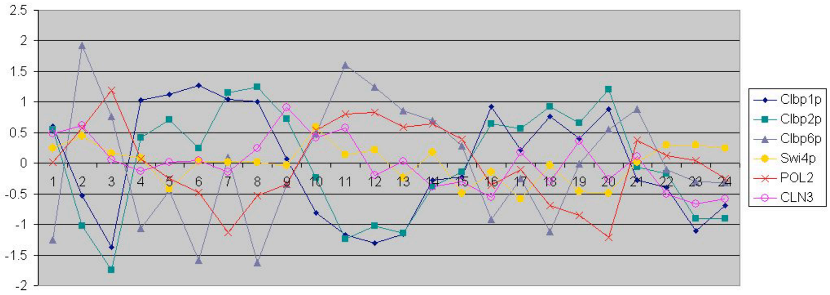

Fig. 1. Transcriptional profiles of cell cycle marker genes in [2] data, cdc15 subset. CLN3p is required for M→G1 transition, and it also indirectly activates the SBF complex. Swi4p is part of the transcription-regulating complex SBF, which binds its targets in early G1, but it is active only in late G1, and is a key player for G1→S transition (together with MBF, whose components are not reported); Clb6p is responsible for an initial inactivation (through nuclear export) of SBF and MBF during S-phase; POL2 has been assumed as a rough predictor of DNA-replication activity under S-phase; Clb1/2p are responsible for switching off SBF and MBF in G2 phase, and therefore are key players of S→G2 transition. Comparing the peaking areas of Clbp6p and POL2, and the depression areas of Clbp1/2p, it is evident that G1 and S phases are "compressed" in the initial time-steps.

increasing profile, although c11w1 is much more noisy; we probably observe this discrepancy because clustering has not been optimized employing functional annotation maximization as the objective for optimal k-means selection. We intend to include this feature in an enhanced version of our method.

The relative stability of functional annotations in these clusters suggests that regulation of basal metabolism and protein synthesis is coupled in the same fashion through all the cell cycle, without any detectable de-coupling event. Please note that these connections were not successfully found employing the alternative method based on Jaccard coefficient.

6 Conclusion

We have presented an application of *graph kernel* methods to group clusters of gene expression measurements organized over a time line. Our main contribution is an initial kernel similarity function that considers the taxonomical graph structure nature of GO terms in the context of an enrichment procedure that takes into account the temporal distribution of biological processes. The preliminary experimental results on the Spellman's Yeast Cell Cycle data-set encourage the use of this application of graph kernels versus a simple flat list approach based on the Jaccard distance index.

Our next concern will be to address the use of different kernel functions and dissimilarity indexes to extend the range of applicability of our approach.

translation and ribosome subgroup	
1.33e-07	translational elongation
9.58e-06	ribosome
1.13e-05	cytosolic large ribosomal subunit (sensu Eukaryota)
1.25e-05	cytosolic small ribosomal subunit (sensu Eukaryota)
2.61e-05	translation
4.70e-05	structural constituent of ribosome
2.97e-03	translational initiation
glycolysis and glyconeogensis subgroup	
2.62e-08	glycolysis
1.19e-06	gluconeogenesis
cell wall, plasma membrane and vescicular compartments subgroup	
3.46e-08	membrane
7.93e-05	transporter activity
3.80e-04	cell wall (sensu Fungi)
7.86e-04	transport
2.46e-03	endoplasmic reticulum
3.00e-03	plasma membrane

Table 3. c10w2 enrichment reports these terms.

translation and ribosome subgroup	
1.25e-05	translational elongation
2.54e-04	ribosome
glycolysis and glyconeogensis sub-group	
4.121e-06	glycolysis
7.36e-06	gluconeogenesis
3.14e-05	hexose transport
cell wall, plasma membrane and vescicular compartments subgroup	
1.89e-05	integral to plasma membrane
2.51e-05	transporter activity
3.14e-04	cell wall (sensu Fungi)
4.71e-04	plasma membrane
6.69e-04	membrane
3.79e-03	integral to membrane

Table 4. c13w3 enrichment reports these terms.

References

1. Gene Ontology Consortium: The Gene Ontology (GO) project in 2006. Nucleic Acid Research (Database issue) **34** (2006) D322–D326
2. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive Identification of Cell Cycle-Regulated Genes of the Yeast Saccharomyces Cerevisiae by Microarray Hybridization. Molecular Biology of the Cell **9** (1998) 3273–3297
3. Li, X., Quigg, R.J.: An Integrated Strategy for the Optimization of Microarray Data Interpretation. Gene Expression **4-6** (2005) 223–230
4. Khatri, P., Draghici, S.: Ontological analysis of gene expression data: current tools, limitations and open problems. Bioinformatics **21** (2005)
5. Doherty, J.M., Carmichael, L.K., Mills, J.C.: GOurmet: a tool for Quantitative Comparison and Visualization of Gene Expression Profiles Based on Gene Ontology (GO) Distributions. BMC Bioinformatics **7**(151) (March 2006)
6. Bar-Joseph, Z.: Analyzing time series gene expression data. Bioinformatics **20**(16) (2004) 2493–2503
7. Ernst, J., Bar-Joseph, Z.: STEM: a tool for the analysis of short time series expression data. BMC Bioinformatics **7**(191) (2006)

8. Antoniotti, M., Ramakrishnan, N., Kumar, D., Spivak, M., Mishra, B.: Remembrance of Experiments Past: Analyzing Time Course Datasets to Discover Complex Temporal Invariants. Technical Report CIMS TR2005-858, Bioinformatics Group, Courant Institute of Mathematical Sciences, New York University (February 2005)
9. Ramakrishnan, N., Antoniotti, M., Mishra, B.: Reconstructing Formal Temporal Models of Cellular Events using the GO Process Ontology. In: Bio-Ontologies SIG Meeting, ISMB, Detroit MI, U.S.A. (2005)
10. Kleinberg, S., Antoniotti, M., Tadepalli, S., Ramakrishnan, N., Mishra, B.: Remembrance of Experiments Past: A Redescription Based Tool for Discovery in Complex Systems. In: Proceedings of the International Conference on Complex Systems, Boston, MA, U.S.A. (June 2006)
11. Antoniotti, M.: GOALIE site. `http://bioinformatics.nyu.edu/Projects/GOALIE/` (2004-2007)
12. Ben-Hur, A., Noble, W.S.: Kernel methods for predicting protein-protein interactions. Bioinformatics **21** (2005)
13. Schölkopf, B., Tsuda, K., Vert, J.P.: Kernel Methods in Computational Biology. MIT Press (2004)
14. Cortes, C., Haffner, P., Mohri, M.: Positive Definite Rational Kernels. In: Proceedings of the 16th Annual Conference on Learning Theory, Springer-Verlag (2003) 41–56
15. Gärtner, P., Flach, P., Wrobel, S.: On Graph Kernels: Hardness Results and Efficient Alternatives. In: COLT/Kernel. Volume 2777 of Lecture Notes in Artificial Intelligence., Springer-Verlag (2003) 129–143
16. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized Kernels between Labelled Graphs. In: Proceedings of ICML. (2003)
17. Kondor, R.S., Lafferty, J.: Diffusion Kernels on Graphs and Other Discrete Structures. In: Proceedings of ICML. (2002)
18. Borgwardt, K.M., Cheng, S.O., Schönauer: Protein Function Prediction via Graph Kernel. Bioinformatics **21** (2005)
19. Joslyn, C.A., Mniszewski, S.M., Fulmer, A., Heaton, A.: The Gene Ontology Categorizer. Bioinformatics **20** (2004)
20. Lord, P.W., Stevens, R., Brass, A., Goble, C.A.: Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. Bioinformatics **19**(10) (2003)
21. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press (2002)
22. Loganantharaj, R., Cheepala, S., Clifford, J.: Metric for Measuring the Effectiveness of Clustering of DNA Microarray Expression. BMC Bioinformatics **7** (2006)
23. Al-Shahrour, F., Diaz-Uriarte, R., Dopazo, J.: FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. Bioinformatics **20** (2004) 578–580
24. Beißbarth, T., Speed, T.P.: GOstat: find statistically overrepresented Gene Ontologies within a group of genes. Bioinformatics **20**(9) (2004) 1464–1465
25. Robinson, P.N., Wollstein, A., Bohme, U., Beattie, B.: Ontologizing gene-expression microarray dat: characterizing clusters with Gene Ontology. Bioinformatics **20**(6) (2004) 979–981