Functional Distances for Genes Based on GO Feature Maps and their Application to Clustering

Nora Speer, Holger Fröhlich, Christian Spieth, and Andreas Zell
University of Tübingen, Centre for Bioinformatics Tübingen (ZBIT),
Sand 1, D-72076 Tübingen, Germany
Email: nspeer@informatik.uni-tuebingen.de

***Abstract*— With the invention of high throughput methods, researchers are capable of producing large amounts of biological data. During the analysis of such data, the need for a functional grouping of genes arises. In this paper, we propose a new functional distance measure for genes and its application to clustering. The proposed distance is based on the concept of *empirical feature maps* that are built using the Gene Ontology. Besides, our distance function can be calculated much faster than a previous approach. Finally, we show that using this distance function for clustering produces clusters of genes that are of the same quality as in our previous publication. Therefore, it promises to speed up biological data analysis.**

I. Introduction

In the past few years, DNA microarrays have become major tools in the field of functional genomics. In contrast to traditional methods, these technologies enable researchers to collect tremendous amounts of data, whose analysis itself constitutes a challenge. On the other side, these high-throughput methods provide a global view on the cellular processes as well as on their underlying regulatory mechanisms and are therefore quite popular among biologists.

During the analysis of such data, researchers use different approaches in order to deal with the huge amounts of data that they gathered. Some use statistics to find significantly regulated genes that may be involved in the underlying process due to their change in expression. Others apply pattern recognition methods to cluster the genes according to their expression profiles. The hypothesis is, that genes with expression patterns similar to those of genes known to be involved in the examined biological process, may play a role in the process, too. In both cases, researchers often end up with long lists of interesting candidate genes that need further examination. At this point, a second step is almost always applied: biologists categorize these genes by known biological functions and thus try to combine a pure numerical analysis with biological information.

In this paper we address the problem of finding functional gene clusters only based on Gene Ontology terms. The advantage of such a method is that no *a priori* knowledge about relevant pathways is necessary except a mapping from genes to their ontological information. The latter is often available in public databases. Given the GO terms we are able to compute a functional distance between genes [13]. This information is fed into a clustering algorithm. To our best knowledge, so far there exists no automatic method that produces a biologically plausible functional clustering of genes just based on the GO apart from our earlier publications [22], [21]. In [21], we represented each gene by its functional distance to all other genes. This encoding allowed us to construct a valid mathematical distance measure between genes and the incorporation of all GO annotations into the distance function. Both was not given in [22]. Now, we apply a similar representation as in [21], but instead of representing each gene by its distance to all other genes, we only utilize the distance of each gene to a subset of the GO terms that occur in the dataset. The advantage is, that the number of *prototypes* (GO terms) that are necessary to construct such a feature vector is quite small which makes the method much faster than the one in [21]. Besides, we provide a method to automatically select those prototypes. Again, there is also a deeper connection to *Kernel Methods* [19], which will be discussed later on in this paper. The final grouping of the genes can then be performed by any clustering method.

The organization of this paper is as follows: section I gives a general introduction, discusses related work and provides a brief introduction to the Gene Ontology. Section II explains our method in detail. In section III, an application of our distance function to clustering is shown on real world datasets. Finally, in section IV, we conclude.

A. Related Work

While GO analysis is an increasingly important field, existing techniques suffer from some weaknesses: Many methods consider the GO simply as a list of terms, ignoring any structural relationships [2], [5], [16], [20], [25]. Others regard the GO primarily as a tree and convert the GO graph into a tree structure for determining distances between nodes [11]. Again others use a pseudo-distance that does not fulfill all metric conditions and relies on counting path lengths [9]. This is a delicate approach in unbalanced graphs like the GO, whose subgraphs have different degrees of detail.

Besides, the aim of some methods is primarily either to use the GO as preprocessing [1] or as visualization tool [4]. Only few approaches utilize its structure for computation. Many methods are scoring techniques describing a list of genes annotated with GO terms [2], [4], [5], [11], [16], [20], [25]. But to our knowledge and apart from our earlier publications [22], [21], [23], there exists no automatic functional GO-based clustering method. One method is related to clustering and can be used to indicate which clusters are present in the data [9].

0-7803-9387-2/05/$20.00 ©2005 IEEE

However, it suffers from the weaknesses that come along with using pseudo-distances as mentioned earlier.

B. The Gene Ontology

The Gene Ontology (GO) is one of the most important ontologies within the bioinformatics community and is developed by the Gene Ontology Consortium [24]. It is specifically intended for annotating gene products with a consistent, controlled and structured vocabulary. Gene products are for instance sequences in databases as well as measured expression profiles. The GO is independent from any biological species. It represents terms in a Directed Acyclic Graph (DAG), covering three orthogonal taxonomies or "aspects": *molecular function*, *biological process* and *cellular component*. The GO-graph consists of over 18,000 terms, represented as nodes within the DAG, connected by relationships, represented as edges. Terms are allowed to have multiple parents as well as multiple children. Two different kinds of relationship exist: the "is-a" relationship (for example *photoreceptor cell differentiation* is a child of *cell differentiation*) and the "part-of" relationship (*regulation of cell differentiation* is part of *cell differentiation*).

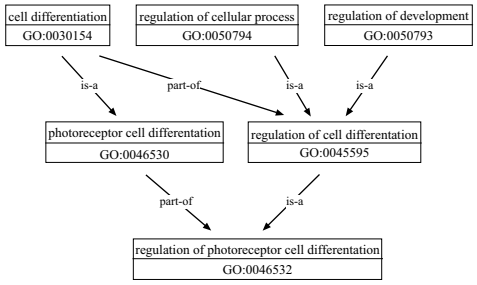

Fig. 1. Relations in the Gene Ontology. Each node is annotated with a unique accession number.

Providing a standard vocabulary across any biological resources, the GO enables researchers to use this information for automatic data analysis done by computers and not by humans.

II. Methodology

To find functional gene clusters, one needs to compute functional distances between genes which can only be computed due to functional annotation. Here, we use Gene Ontology terms. Since the functional distances are computed between GO terms and since each gene can be annotated with more than one GO term, we face the problem of combining all the different possible functional distances to compute one distance between a pair of genes. Taking the smallest distance is a possible solution of this problem and was presented in [22]. It has the disadvantage that this approach looses the properties of a metric and that only a dissimilarity matrix is available and thus means cannot be calculated which reduces the amount of data mining techniques that can be applied. Besides, when taking the minimum distance, genes can by definition only be grouped according to one function.

Therefore, in [21], we introduced the concept of representing each gene by a feature vector describing the function of a gene. We proposed to represent each gene by its functional distance to all other genes. This encoding allowed us to construct a valid mathematical distance measure between genes and the usage of mean calculation. In addition, the grouping according to more than one function was possible. In this paper, we apply a similar representation, but instead of using all genes as prototypes as in our earlier publication [21], we only utilize a subset of the GO terms that occur in the dataset. This has all advantages of the feature vector representation, plus that the computation of the feature vector is less expensive and consequently much faster.

Our method consists of different steps that will be explained separately in this section: the mapping of the genes to the Gene Ontology (sec. II-A), the calculation of functional distances on GO terms (sec. II-B), the concept of the feature vectors (sec. II-C) and the selection of prototype terms to construct the feature vector (sec. II-D).

A. Mapping the Genes to the Gene Ontology

The functional distance measure is based on distances on pairs of GO nodes in a DAG, whereas in general, researchers are dealing with database ids of genes or probes. Therefore, a mapping $\mathcal{M}$ that relates the genes of a microarray experiment to nodes in the GO graph is required. Many databases (e.g. TrEMBL (GOA-project)) provide GO annotation for their entries, companies like Affymetrix provide GO mappings to their probeset ids and the GO Consortium also makes mappings to other databases and ontologies available. For one of our datasets, we used GeneLynx [12] to map the gene to GO ids. For the other dataset the mapping to the GO was done by Hvidsten *et al.* [6] and is publicly available.

B. Distances between GO terms

To calculate functional distances between GO nodes, we rely on a technique that was originally developed for other taxonomies like WordNet to measure semantic distances between words [8].

Following the notation in information theory, the information content (IC) of a term t can be quantified by the probability of occurrence of this term or any child term in a dataset [15]:

$$IC(t) = -\ln P(t) \tag{1}$$

where $P(t)$ is the probability of encountering an instance of term t in the data.

In the case of a hierarchical structure, such as the GO, where a term in the hierarchy subsumes those lower in the hierarchy, this implies that $P(t)$ is monotonic as one moves towards the root node. As the node's probability increases, its information content or its informativeness decreases. The root node has a probability of 1, hence its information content is 0. As the three aspects of the GO are disconnected subgraphs, this is still true if we ignore the root node "Gene Ontology" and take, for example, "biological process" as our root node instead. $P(t)$ is simply computed using maximum likelihood estimation [21].

To compute a similarity between two terms, one can use the IC of their common ancestor. As the GO allows multiple parents for each term, two terms can share ancestors by multiple paths. We take the minimum $P(t)$, if there is more than one ancestor. This is called P_{ms}, for *probability of the minimum subsumer* [13]. Thereby, it is guaranteed, that the most specific parent term is selected:

$$P_{\text{ms}}(t_i, t_j) = \min_{t \in S(t_i, t_j)} P(t) \tag{2}$$

where $S(t_i, t_j)$ is the set of parental terms shared by both t_i and t_j.

Based on Eqn. 1 and 2, Jiang and Conrath developed a distance measure [8], which is the inverse of similarity. They defined the distance of two nodes (in our case GO terms) t_i, t_j as follows:

$$d(t_i, t_j) = 2 \ln P_{ms}(t_i, t_j) - (\ln P(t_i) + \ln P(t_j)) \tag{3}$$

One should note, that the probability of a term as well as the resulting distance between two terms differs from dataset to dataset, depending on the distribution of terms. Therefore, in the end, also the clustering differs from a general clustering/categorization of the GO and a subsequent mapping of the genes to such a general categories. Due to our approach, we are able to arrange the resulting cluster boundaries depending on the distribution of the GO terms in the data either more specific (if the terms concentrate on a specific part of the GO) or more general (if the terms are widely spread).

C. Distances between Genes Using Feature Vectors

For each gene g_i we construct a feature vector $\phi_p(g_i)$ relative to some *prototypes* $\mathbf{p} = (p_1, ..., p_N)^T$

$$\phi_p(g_i) = (d(g_i, p_1), ..., d(g_i, p_N))^T. \tag{4}$$

This construction is known as an *empirical feature map* [18], [19]. In our case prototypes are a subset of GO terms that are present in the dataset: $\mathbf{p} = (t_1, ..., t_N)^T$. Since each gene can be annotated with a different number of GO terms, each feature of $\phi_p(g_i)$ corresponds to the smallest distance between all GO term annotations of the gene and the corresponding prototype $p_i = t_i$. That means each gene g_i is represented by its smallest functional distance to each of the prototypes. Now, the distance between two genes g_i and g_j is simply given by

$$\hat{d}(g_i, g_j) = \|\phi(g_i) - \phi(g_j)\|. \tag{5}$$

There exists a deep connection to the construction of so called *kernel functions*, which can be viewed as a general similarity measure $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ with the property of being symmetric and positive definite: More specifically, we have the equality (c.f. [19])

$$\begin{aligned} \hat{d}^2(g_i, g_j) &= \|\phi(g_i) - \phi(g_j)\|^2 \\ &= \langle \phi(g_i), \phi(g_i) \rangle - 2\langle \phi(g_i), \phi(g_j) \rangle + \langle \phi(g_j), \phi(g_j) \rangle \\ &= k(g_i, g_i) - 2k(g_i, g_j) + k(g_j, g_j). \end{aligned} \tag{6}$$

That means by defining $\phi : \mathcal{X} \rightarrow \mathcal{H}$ we map our data into some Hilbert space $\mathcal{H}$. The scalar product in this space defines a kernel $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ and hence a similarity measure between two genes g_i and g_j in our original input space $\mathcal{X}$. If we take the normalization $\phi_{norm}(g_i) = \frac{\phi(g_i)}{\|\phi(g_i)\|}$, we recover the normalized kernel [19]:

$$\begin{aligned} k_{norm}(g_i, g_j) &= \langle \phi_{norm}(g_i), \phi_{norm}(g_j) \rangle \\ &= \frac{k(g_i, g_j)}{\sqrt{k(g_i, g_i) k(g_j, g_j)}}. \end{aligned} \tag{7}$$

D. Selecting the Prototypes

As already mentioned earlier, we use a set of GO terms as prototypes. Since each gene is annotated with one or more GO terms, the selected prototypes should fulfill certain criteria: At first, the prototypes should be well distributed over the GO space, not necessarily over the whole GO with its 18,000 terms, but over those terms that are present in the dataset. Additionally, since our GO term distance function (see Eqn. 3) uses the probability of occurrence of the smallest common parent term, a certain tradeoff between specificity and generality of the prototypes must be guaranteed. This is important, because if the prototypes are too specific, nearly every other term has a large distance to them and discrimination becomes impossible. On the other hand, too general prototypes (e.g. GO Slim terms) may cause the same effect, especially, when the GO annotations of the dataset are mostly very specific.

An easy and straightforward way to chose prototype terms that fulfill the above requirements is to select a well distributed subset of of those terms that occur in the dataset. Such an approach has several advantages: first it reduces the number of considered terms from about 18,000 to a couple of hundred, and second, it automatically guarantees that the prototypes are not completely beyond the space covered by the dataset.

To find a subset of points that are well distributed over a dataset, one can cluster the data and use the cluster *centers*. If the precondition is that the subset should only contain points also present in the dataset or if mean calculation is not possible (such as in the GO), one could take the cluster *medoids* instead. A *medoid* is defined as the most centrally located item in a cluster that is, the item in the cluster whose average dissimilarity to all other items in the cluster is minimal [10]. Thus, medoids can easily be computed from dissimilarity data.

Referring to our problem, we cluster all GO terms that are present in the dataset using a method that was published in [23]. It is based on a Spectral Clustering algorithm by Ng *et al.* [14], that we also briefly review in sec. III, since it is also used later on in this paper. In contrast to our work in [23], in this paper, we apply a gaussian kernel to construct the affinity matrix

$$A_{t_i t_j} = \exp(\frac{-d(t_i, t_j)^2}{2\sigma^2}), \tag{8}$$

with $d(t_i, t_j)$ denoting the GO distance according to Eqn. 3 between term t_i and t_j. The parameter σ was tuned automatically such that the average distortion of the points in Eigenvector space becomes minimal as proposed in [14].

After clustering, we compute the medoids of each cluster as described above and utilize them as prototypes to construct the feature vector as explained in sec. II-C.

III. Application: Clustering

One possible scenario where researchers would like to group a list of genes according to their function is when they received lists of up- or down-regulated genes from the analysis of an DNA microarray experiment. Thus, we chose two publicly available microarray data sets, annotated the genes with the GO and used them for functional clustering (see III-C). We only use the taxonomy *biological process*, because we are mainly interested in gene function in a more general sense. However, our method can be applied in the same way for the other two taxonomies.

Given our representation of each gene as a feature vector, we can choose any clustering algorithm to group our data. Here, we use three different cluster algorithms: Spectral Clustering, K-means and Single Linkage clustering. The last two are standard methods, but Spectral Clustering isn't and will therefore be explained briefly in the next section.

A. Spectral Clustering

A set of objects (in our case genes) to be clustered will be denoted by X, with $|X| = n$. Given an affinity measure $A_{ij} = A_{ji} \geq 0$ for two objects i, j, the affinities A_{ij} can be seen as weights on the undirected edges ij of a graph G over X. Then, the matrix $A = [A_{ij}]$ is the real-valued adjacency matrix for G. Let $d_i = \sum_{j \in X} A_{ij}$ be called the degree of node i, and D be the diagonal matrix with d_i as its diagonal. A clustering $C = \{C_1, C_2, \ldots, C_K\}$ is a partitioning of X into the nonempty mutually disjoint subsets $C_1, C_2, \ldots, C_K$. In the graph theoretical paradigm a clustering represents a multiway cut in the graph G.

In general, all Spectral Clustering algorithms use Eigenvectors of a matrix (derived from the affinity matrix A) to map the original data to the K-dimensional vectors $\{\gamma_1, \gamma_2, \ldots, \gamma_n\}$ of the spectral domain $\Re^K$. Then, in a second step, these vectors are clustered with standard clustering algorithms. Here, we use K-means. We chose the newest Spectral Clustering algorithm by Ng *et al.* [14] and we will now explain it briefly:

1) From the affinity matrix A and its derived diagonal matrix D, compute the Laplacian matrix $L = D^{-1/2} A D^{-1/2}$.
2) Find $v^1, v^2, \ldots, v^K$, the Eigenvectors of L, corresponding to the K largest Eigenvalues.
3) Form the matrix $V_{n \times k} = \left[v^1, v^2, \ldots, v^K\right]$ with these Eigenvectors as columns.
4) Form the matrix Y from V by renormalizing each of V's rows to have unit norm.
5) Cluster the rows of $Y = [\gamma_1, \gamma_2, \ldots, \gamma_n]$ as points in a K-dimensional space using standard methods.
6) Finally assign the original object i to cluster j if and only if row γ_i of the matrix Y was assigned to j.

Since Spectral Clustering relies on the affinity matrix A, affinities are often computed with a kernel function, e.g. Eqn. 8, with $d(i, j)$ denoting the distance between object i and j and σ denoting the kernel width. Since in our case, the objects are genes, $d(i, j)$ is the Euclidean distance of the feature vectors for each gene. For the final clustering in the Eigenvector space, we choose the K-means algorithm by Zha *et al.* [26], which leads to a unique and global optimal solution. This has the advantage that no restarts are necessary. The parameter σ can be tuned automatically such that the average distortion of the points in Eigenvector space becomes minimal [14].

B. Cluster Validity

We selected the number of clusters K in our data according to the maximal Average Silhouette Index [17]. The Silhouette value for each point is a measure of how similar that point is to points in its own cluster vs. points in other clusters, and ranges from -1 to +1. It is defined as:

$$S(i) = \frac{\min(\bar{d_B}(i,j)) - \bar{d}_W(i)}{\max(\bar{d}_W(i), \min(\bar{d_B}(i,j))} \tag{9}$$

where $\bar{d}_W(i)$ is the average distance from the j-th point to the other points in its own cluster, and $\bar{d_B}(i, j)$ is the average distance from the i-th point to points in another cluster j.

C. Datasets

The authors of the first dataset examined the response of human fibroblasts to serum on cDNA microarrays in order to study growth control and cell cycle progression. They found 517 genes whose expression levels varied significantly, for details see [7]. We used these 517 genes for which the authors provide NCBI accession numbers. The GO mapping was done using GeneLynx [12]. After mapping to the GO, 238 genes showed one or more mappings to *biological process* or a child term of *biological process*. These 238 genes were used for the clustering.

In order to study gene regulation during eukaryotic mitosis, the authors of the second dataset examined the transcriptional profiling of human fibroblasts during cell cycle using microarrays [3]. Duplicate experiments were carried out at 13 different time points ranging from 0 to 24 hours. Cho *et al.* found 388 genes whose expression levels varied significantly. Hvidsten *et al.* [6] provide a mapping of the dataset to GO. 233 of the 388 genes showed at least one mapping to the GO *biological process* taxonomy and were thus used for clustering.

D. Results

In the experiments, we first compare our new distance function to the one proposed in [21]. Then, we present some results using this distance function by showing their application to clustering.

The advantage of using only a small set (e.g. 20, 30 or 40) of GO terms as prototypes instead of all genes of the dataset as proposed in [21] is quite obvious considering the number of distance calculations that are necessary to construct the feature vector: For each feature, one has to compute the smallest distance between all GO annotations of the respective gene and the respective prototype, which is either given by a

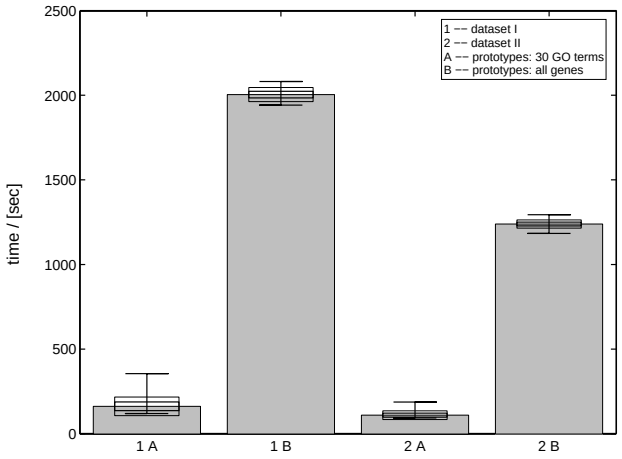

Fig. 2. Running times in sec. averaged over 20 runs for the calculation of the feature maps with 30 GO terms (A) and all genes (B) [21] as prototypes). The plot shows the minimum/maximum, the standard deviation and the 90 percent confi dence interval.

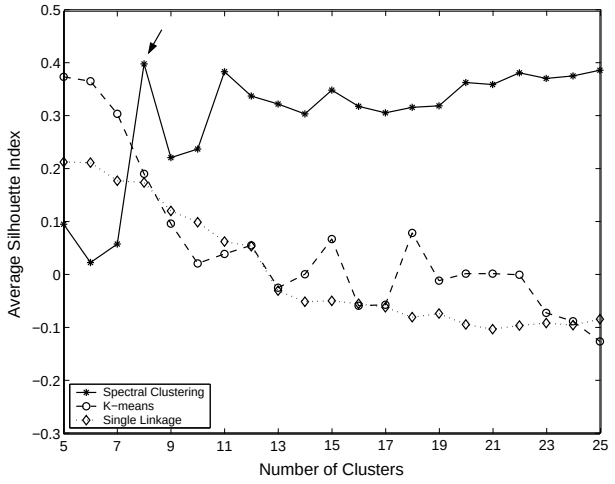

Fig. 3. Average Silhouette index of dataset I. The arrow indicates the solution with the best Silhouette index.

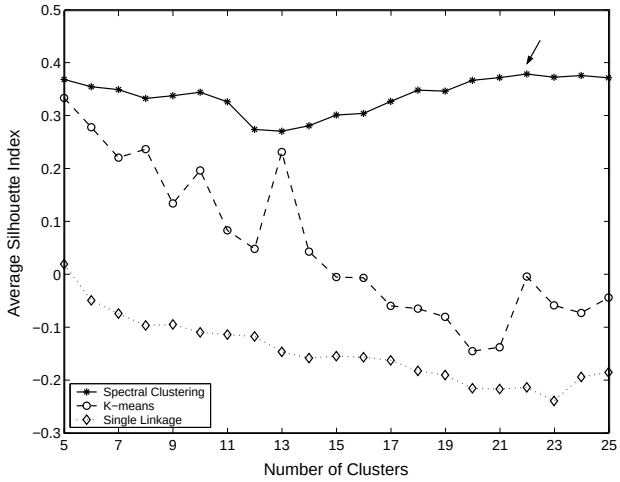

Fig. 4. Average Silhouette index of dataset II. The arrow indicates the solution with the best Silhouette index that was examined in more detail.

GO term or as in [21] again by a gene. This means that one has to calculate $n * m$ distances per prototype with n being the number of GO annotations of the first and m the number of GO annotations of the second gene, in case the prototype is a gene. With our approach, only $n * 1$ distance calculations are necessary, because we use GO terms as prototypes and not genes. Besides, with our approach the total number of prototypes is much smaller. In preliminary experiments, we tested several numbers of prototypes and found that with 30 prototypes the GO space of the datasets seemed to be covered quite well.

Fig. 2 shows the running times for the feature vector calculation for dataset I and II, respectively. The absolute running times depend on the implementation (we use the MySQL database for the GO graph), but the ratio of distance calculations between the two methods are independent of the implementation and remain the same. Our results are averaged over 20 runs. Fig. 2 indicates that with our new approach the feature vector calculation is more than 10 times faster. In addition, which was not shown here, the Euclidean distance calculation with a 30-dimensional vector is also faster than with a vector having a dimension of a couple of hundred. This fact carries even more weight, if the datasets get larger and if distance matrices cannot be kept in memory anymore, but have to be recalculated.

Getting down to the application part, we compare the three cluster methods as described above: Spectral Clustering, K-means and Single Linkage clustering, which are all based on the proposed feature vector representation. We choose the K-means algorithm by Zha *et al.* [26], which leads to a unique and global optimal solution. This has the advantage that no restarts are necessary. We evaluated the three algorithms by means of the Average Silhouette Index (Eqn. 9) and a detailed look at the GO annotations of the genes in each cluster. Unfortunately, due to space limitations, we cannot show all clusters in detail, therefore, we confine ourselves to show selected clusters of dataset II. We simply picked dataset II, since its clusters are smaller than those of the other dataset.

TABLE I

SUMMARY OF THE GO ANNOTATIONS FOR THE CLUSTERS OF DATASET I

Cluster	Summary of the GO annotations
1	apoptosis, cell cycle, protein dephosporylation
2	apoptosis, development, heat response
3	cell adhesion, chemotaxis, G-protein signaling
4	apoptosis, stress response
5	cell-cell signaling, development
6	cell cycle
7	cell adhesion, cell-cell signaling, cell proliferation
8	DNA replication, transcription, cellular metabolism

Figures 3 and 4 show the Average Silhouette Index for cluster numbers $K = 5, ..., 25$ for all three clusterings (Spectral, K-means and Single Linkage). Both figures indicate that the Spectral Clustering method gives significant better results than

TABLE II

SUMMARY OF THE GO ANNOTATIONS FOR THE CLUSTERS OF DATASET II

Cluster	Summary of the GO annotations
1	cell motility, oncogenesis
2	regulation of transcription, signal transduction
3	development, oncogenesis
4	protein amino acid phosphorylation, signal transduction
5	energy pathways
6	apoptosis, development, transcription
8	cellular morphogenesis, cell motility
9	cell motility
10	cell proliferation, signal transduction
11	protein modifi cation, stress response
12	RNA processing
13	development
14	DNA replication
15	cell cycle, oncogenesis
16	transport
17	metabolism
18	oncogenesis, stress response
19	DNA replication, protein modifi cation
20	DNA repair, transcription
21	cell cycle
22	signal transduction

TABLE III

CLUSTER 12 FROM DATASET II: GENES RELATED TO RNA PROCESSING

Acc. number	Gene Ontology terms
D84110_at	**RNA processing**
L28010_at	**RNA processing**
M92843_at	**mRNA catabolism**
U02493_at	**mRNA splicing**
	mRNA processing
U59321_at	**RNA processing**
U75679_at	**mRNA processing**
	histone mRNA 3'-end processing
X14684_at	transcription from Pol III promoter
	tRNA modification
	histone mRNA metabolism

TABLE IV

CLUSTER 14 FROM DATASET II: GENES RELATED TO DNA REPLICATION AND REPAIR

Acc. number	Gene Ontology terms
D26018_at	**DNA dependent DNA replication**
D38073_at	**DNA replication initiation**
D38551_at	**double-strand break repair**
	DNA recombination meiotic recombination
J04611_at	**DNA ligation**
	double-strand break repair
	double-strand break repair via nonhomologous end-joining
	DNA recombination
L07541_at	**DNA strand elongation**
M87339_at	**DNA strand elongation**
U27516_at	**double-strand break repair**
	mitotic recombination
	meiotic recombination
U72066_at	cell cycle checkpoint
	DNA repair
	regulation of transcription from Pol II promoter
X62153_at	**DNA replication initiation**
X74331_at	**DNA replication, priming**

TABLE V

CLUSTER 15 FROM DATASET II: GENES RELATED TO CELL CYCLE AND ONCOGENESIS

Acc. number	Gene Ontology terms
M31423_at	**oncogenesis**
	cell growth and/or maintenance
M86699_at	**regulation of cell cycle**
	oncogenesis
	spindle assembly
	mitotic spindle assembly
	mitotic spindle checkpoint
	positive regulation of cell proliferation
S81914_at	apoptosis
	anti-apoptosis
	embryogenesis and morphogenesis
	cell growth and/or maintenance
U01038_at	**regulation of cell cycle**
	oncogenesis
	mitosis
	cell proliferation
U09579_at	**regulation of cell cycle**
	regulation of CDK activity
	oncogenesis
	cell cycle arrest
	negative regulation of cell proliferation
	induction of apoptosis by intracellular signals
U33203_at	**oncogenesis**
	negative regulation of cell proliferation
U33286_at	nucleocytoplasmic transport
	apoptosis
	cell proliferation
U33761_at	**regulation of cell cycle**
	G1/S transition of mitotic cell cycle
	oncogenesis
	cell proliferation
U58090_at	**G1/S transition of mitotic cell cycle**
	oncogenesis
	cell cycle arrest
	negative regulation of cell proliferation
	induction of apoptosis by intracellular signals
U73737_at	mismatch repair
	oncogenesis
X51688_at	**regulation of CDK activity**
	oncogenesis
	mitotic G2 checkpoint

TABLE VI

CLUSTER 17 FROM DATASET II: GENES RELATED TO METABOLSIM

Acc. number	Gene Ontology terms
D14686_at	**glycine catabolism**
D30037_at	**lipid metabolism**
L39211_at	**fatty acid beta-oxidation**
L42452_at	**glucose metabolism**
M18700_at	**proteolysis and peptidolysis**
	digestion
	cholesterol metabolism
M77836_at	**proline biosynthesis**
S67325_at	**fatty acid catabolism**
U24183_at	**glucose metabolism**
	regulation of glycolysis
U31929_at	**steroid biosynthesis**
	sex determination
U47105_at	**cholesterol biosynthesis**
X07496_at	circulation
	cholesterol metabolism
X92720_at	**glucose metabolism**

TABLE VII

CLUSTER 21 FROM DATASET II: GENES RELATED TO CELL CYCLE

Acc. number	Gene Ontology terms
L11353_at	**negative regulation of cell proliferation**
L22005_at	**cell cycle checkpoint**
	DNA replication
	checkpoint G1/S transition of mitotic cell cycle
L26336_at	male meiosis
	spermatid development
M60974_at	**regulation of cell cycle**
	regulation of CDK activity
	DNA repair
	apoptosis
	response to stress
	cell cycle arrest
M81933_at	**regulation of cell cycle**
	regulation of CDK activity
M90657_at	**N-linked glycosylation cell proliferation**
	pathogenesis
U05340_at	**regulation of cell cycle**
	ubiquitin-dependent protein catabolism
	cell cycle
U37426_at	**mitotic spindle assembly**
	mitosis
U40343_at	**regulation of CDK activity**
	cell cycle arrest
	negative regulation of cell proliferation
U47414_at	**cell cycle checkpoint**
U53204_at	cytoskeletal anchoring
U53446_at	**cell proliferation**
U56816_at	**regulation of CDK activity**
	mitosis
	regulation of mitosis
U63743_at	**centromere binding**
	mitosis
	cell proliferation
X05360_at	**regulation of cell cycle**
	start control point of mitotic cell cycle
X54941_at	**regulation of cell cycle**
	regulation of CDK activity
	cell proliferation
X54942_at	**regulation of CDK activity**
	cell proliferation
X58377_at	cell-cell signaling
	cell proliferation
	positive regulation of cell proliferation
X62048_at	**regulation of cell cycle**
X65550_at	**regulation of cell cycle**
	cell proliferation
X66364_at	**cell proliferation**
X67155_at	**mitotic spindle elongation**
X80230_at	**regulation of cell cycle**
	transcription initiation from Pol II promoter
	RNA elongation from Pol II promoter
	cell proliferation
X85137_at	**mitotic spindle assembly**
	mitosis
Z24725_at	**regulation of cell cycle**
	cell proliferation
Z29066_at	**regulation of cell cycle**
	mitosis
	regulation of mitosis
Z29067_at	**cell cycle**
Z36714_at	**regulation of cell cycle**

the other two approaches. Only for dataset I and very small cluster numbers, K-means and Single Linkage are superior. According to these plots, the best solution for dataset I has 8 clusters and for dataset II the best solution with has 22 clusters.

Tables I and II summarize the GO annotations of the genes in each cluster for dataset I and dataset II, respectively. It is notable that the clusterings of both datasets contain clusters with genes that share different functions, e.g. cluster 1 of dataset I contains genes, that are involved in apoptosis, cell cycle and protein dephosphorylation. Many cell cycle genes are also involved in apoptosis, since usually when apoptosis is induced, the cell cycle has to be stopped first.

Additionally, we show five selected clusters of dataset II in detail (Tab. IV-VII): cluster 12, 14, 16, 17 and 21. In all tables, GO terms belonging to the same biological process are printed in bold. Each gene in cluster 12 is related to RNA processing (Tab. III). All genes of cluster 14 have at least one, but in most of the cases more than one GO annotation that is related to DNA replication, either by DNA repair or DNA strand elongation or recombination (Tab. IV). The genes of cluster 15 are mainly involved in cell-cycle and oncogenesis (Tab. V) and those in cluster 17 can be characterized to be involved in all kinds of metabolic processes (Tab. VI). The genes of cluster 21 are mainly related to all kinds of cell cycle processes (Tab. VII).

Other clusters of dataset II (the detailed annotation of all clusters cannot be shown due to space limitations) contain genes that share the functions like protein modification and stress response (cluster 11), development (cluster 13), transport (cluster 16), oncogenesis and stress response (cluster 18), protein modification and DNA replication (cluster 19) and DNA repair and transcription (cluster 20), just to name a few (see Tab. II for more clusters).

One should note that there are many clusters with genes sharing more than one function. Genes that share one function, but differ in another are were placed in different clusters, e.g. cluster 1 (oncogenesis and cell motility), cluster 3 (oncogenesis and development), cluster 15 (oncogenesis and cell cycle) and cluster 18 (oncogenesis and stress response). This discrimination is probably possible due to the feature vector representation of each gene.

IV. CONCLUSION

In this paper we presented a new GO-based distance function for genes and its application to clustering. We showed that using this distance function in the application of functional clustering of genes yields clusters that contain genes, which participate in the same biological processes as indicated by their GO annotation. Additionally, we are able to distinguish between clusters of genes that share one, but differ in a second function, e.g. cell cycle genes also related to oncogenesis and genes also related to oncogenesis but additionally also to stress response. Since our functional distances are based on GO annotations, our approach is quite general and can be applied to any kind of data for which GO annotations are available. This is the case for many entries in public databases.

Besides, the proposed distance function has some theoretical advantages: First, the representation of the genes as feature vectors uses all available GO annotation, in contrast to previously presented methods where only smallest distances

(corresponding only to one single annotation) were used [22]. This means, that we are now dealing with a proper metric space. Second, since each gene is represented by a numerical vector and most pattern recognition methods are developed for numerical data, we can apply nearly all established pattern recognition algorithms. Limitations like the inability to calculate means that was always a problem while calculating with the GO as a graph are now no longer present.

Furthermore, the reduced feature set of 30 GO terms compared to our previous publication [21] makes the distance calculation computationally less expensive and thus much faster as indicated by our results. This aspect especially carries weight with large datasets, where the distance matrix is too large to be stored in memory and distance calculation has to be repeated several times. Indeed, it is true, that one could argue, that with our method, one has additional cost, because at first one has to cluster the GO terms to determine the prototypes. But to our experience, the number of different GO terms in a dataset, especially in large datasets, is usually much smaller than the number of genes such that the additional cluster step costs much less effort than calculating the feature vector using all genes as prototypes as in our previous publication [21]. To summarize our results, we showed that using a smaller prototype set than in [21], leads to comparable results in cluster quality, but is much faster than the original method.

Additionally, our experiments revealed that the Spectral Clustering algorithm using our feature vector representation leads to significantly better results than K-means and Single Linkage clustering. This result was expected since there is a theoretical connection between Spectral Clustering and the *empirical feature map* representation as described earlier in this paper.

ACKNOWLEDGMENT

This work was supported by the National Genome Research Network (NGFN) of the Federal Ministry of Education and Research in Germany under contract number 0313323.

REFERENCES

[1] B. Adryan and R. Schuh. Gene Ontology-based clustering of gene expression data. *Bioinformatics*, 20(16):2851–2852, 2004.

[2] T. Beißbarth and T. Speed. GOstat: find statistically overexpressed Gene Ontologies within groups of genes. *Bioinformatics*, 20(9):1464–1465, 2004.

[3] R.J. Cho, M. Huang, M.J. Campbell, H. Dong, L. Steinmetz, L. Sapinoso, G. Hampton, S.J. Elledge, R.W. Davis, and D.J. Lockhart. Transcriptional regulation and function during the human cell cycle. *Nature Genetics*, 27(1):48–54, 2001.

[4] S.W. Doniger, N.Salomonis, K.D. Dahlqusi, K. Vranizan, S.C. Lawlor, and B.R. Conklin. MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. *Genome Biology*, 4(1), 2003.

[5] I. Gat-Viks, R. Sharan, and R. Shamir. Scoring clustering solutions by their biological relevance. *Bioinformatics*, 19(18):2381–2389, 2003.

[6] T.R. Hvidsten, A. Laegreid, and J. Komorowski. Learning rule-based models of biological process from gene expression time profiles using Gene Ontology. *Bioinformatics*, 19(9):1116–1123, 2003.

[7] V.R. Iyer, M.B. Eisen, D.T. Ross, G. Schuler, T. Moore, J.C.F. Lee, J.M. Trent, L.M. Staudt, J Hudson Jr, M.S. Boguski, D. Lashkari, D. Shalon, D. Botstein, and P.O. Brown. The transcriptional program in response of human fibroblasts to serum. *Science*, 283:83–87, 1999.

[8] J.J. Jiang and D.W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, Taiwan, 1998. ROCLING X.

[9] C.A. Joslyn, S.M. Mniszewski, A. Fulmer, and G. Heaton. The gene ontology categorizer. *Bioinformatics*, 20(Suppl. 1):i169–i177, 2004.

[10] L. Kaufmann and P.J. Rousseeu. *Fining Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, New York, 1990.

[11] S.G. Lee, J.U. Hur., and Y.S. Kim. A graph-theoretic modeling on go space for biological interpretation on gene clusters. *Bioinformatics*, 20(3):381–388, 2004.

[12] B. Lenhard, W.S. Hayes, and W.W. Wassermann. GeneLynx: A gene-centric portal to the human genome. *Genome Research*, 11(12):2151–2157, December 2001.

[13] P.W. Lord, R.D. Stevens, A. Brass, and C.A. Goble. Semantic similarity measures as tools for exploring the gene ontology. In *Proceedings of the Pacific Symposium on Biocomputing*, number 8, pages 601–612, 2003.

[14] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, pages 849–856, Cambridge, MA, 2002. MIT Press.

[15] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, volume 1, pages 448–453, Montreal, 1995.

[16] P.N Robinson, A. Wollstein, U. Böhme, and B. Beattie. Ontologizing gene-expression microarray data: characterizing clusters with gene ontology. *Bioinformatics*, 20(6):979–981, 2003.

[17] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational Applications in Math*, 20:53–65, 1987.

[18] B. Schölkopf, S. Mika, C.J.C Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A.J. Smola. Input space versus feature space in kernel based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.

[19] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[20] N.H. Shah and N.V. Fedoroff. CLENCH: a program for calculating Cluster ENriCHment using Gene Ontology. *Bioinformatics*, 20(7):1196–1197, 2004.

[21] N. Speer, H. Fröhlich, C. Spieth, and A. Zell. Functional grouping of genes using spectral clustering and gene ontology. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 298–303. IEEE Press, 2005.

[22] N. Speer, C. Spieth, and A. Zell. A memetic clustering algorithm for the functional partition of genes based on the Gene Ontology. In *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2004)*, pages 252–259. IEEE Press, 2004.

[23] N. Speer, C. Spieth, and A. Zell. Spectral clustering gene ontology terms to group genes by function. In *Proceedings of the 5th Workshop on Algorithms in Bioinformatics (WABI 2005)*, volume 3692 of *Lecture Notes in Bioinformatics (LNBI)*, pages 001–012. Springer, 2005.

[24] The Gene Ontology Consortium. The gene ontology (GO) database and informatics resource. *Nucleic Acids Research*, 32:D258–D261, 2004.

[25] B.R. Zeeberg, W. Feng, G. Wang, A.T. Fojo, M. Sunshine, S. Narashimham D.W. Kane, W.C. Reinhold, and S. Labadridi *et al.* GOminer: a resource for biological interpretation of genomic and proteomic data. *Genome Biology*, 4(R28), 2003.

[26] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. In *Neural Information Processing Systems (NIPS 2001)*, volume 14, pages 1057 – 1064, Dec. 2001.