A Memetic Clustering Algorithm for the Functional Partition of Genes Based on the Gene Ontology

Nora Speer, 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 of a functional grouping of genes arises. In this paper, we propose a new clustering algorithm for the partition of genes or gene products according to their known biological function based on Gene Ontology terms. Ontologies offer a mechanism to capture knowledge in a shareable form that is also processable by computers. Our functional cluster algorithm promises to automatize, speed up and therefore improve 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 the 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 their underlying regulatory mechanisms and are therefore quite popular among biologists. During the analysis of such data, researchers are forced to group genes according to their known biological function to build up their hypothesis about the cellular processes taking place in their systems.

In gene expression analysis researchers tend to cluster genes according to their expression profiles, in order to structure the huge amounts of data that DNA microarrays produce. To our experience the use of available biological knowledge is also essential for the analysis of high throughput data. Therefore, a second step is almost always applied: biologists categorize their long lists of co-expressed genes to known biological functions and thus try to combine a pure numerical analysis with biological information.

So far, many approaches are known that address this problem. Some methods score whole clusterings or each single cluster due to their biological relevance [9], [18], [11], [26]. Others evaluate all annotations in a group of genes and score each single annotation using sophisticated methods [1], [29], [30]. Approaches intending to find clusters of co-expressed genes that share a common function directly incorporate the biological knowledge into the clustering process [12], [33], [31]. All these methods either require a clustering based on or at least not independent of genes expression profiles or simply produce again lists of scored annotations. But in many cases, biologists just want to group lists of genes according to their function independent of any other data.

So far, no automatic method is known to us, that groups genes according to their function alone and thus, can be used as a second step analysis for gene lists obtained by any kind of prior analysis either clustering or statistical over- or under-expression. Therefore, in that case biologists are still forced to do a sequential analysis of their data. First they annotate their genes by hand, which sometimes can be automatized by scripting a database. But then, they go through each single annotation, in some cases also doing time consuming literature search to set the annotation found in a biological context, and try to group the genes in this manner. Such an approach is time consuming, exhausting and may take weeks depending on the size of the dataset. In this paper, we present a method that addresses this question. We use Gene Ontology terms as information about the gene function.

Ontologies offer a mechanism to capture knowledge in a shareable form that is also processable by computers. The advantage of such a method is that it can be applied to any kind of data that can be mapped to Gene Ontology terms. No prior knowledge about relevant pathways is necessary, except a mapping to the ontological information. The latter is often available in public databases. In this paper, we propose a new functional clustering method for genes and show its performance on real world datasets.

The paper is organized as follows: a brief introduction to the ontological information used, the Gene Ontology (GO), is given in section II. The biological distance measure used within the ontology is described in section III. In section IV the memetic clustering algorithm is described in detail. The cluster validation technique applied is described is section V. The performance of our functional clustering algorithm on real world gene expression datasets is shown in section VI. Section VII discusses the paper and outlines areas of future research.

II. 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 [34]. It is specifically intended for annotating gene products with a consistent, controlled and structured vocabulary. The GO is limited to the annotation of gene products and independent from any biological species. It is rapidly growing, having over 16,600 terms

(as of June 2004) and additionally new ontologies covering other biological or medical aspects are being developed.

The GO 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 a number of 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 (neurogenesis and odontogenesis are for example children of organogenesis) and the "part-of" relationship that describes, for instance, that histogenesis is part of organogenesis or axongenesis is part of neurogenesis. The GO terms are used to annotate gene products in the

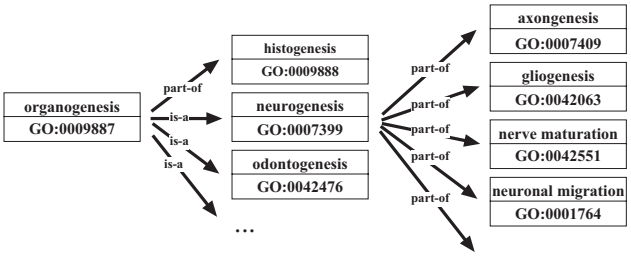

Fig. 1. Relations in the Gene Ontology. Each node is annotated with a unique accession number.

widest sense, e.g. sequences in databases as well as measured expression profiles. By 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. The GO is available as flat files and XML files and has also been ported to a MySQL database scheme [34].

III. Calculating Distances within the Gene Ontology

There are a couple of semantic similarity and distances measures of different complexity [16], [19], [25], [27], most of them were originally developed for taxonomies like WordNet [8]. In this paper we use a similarity measure based on the information content [27] of each GO term developed by Lin in [19] and show how it can easily be transformed into a distance.

The information content of a term is defined as the probability with which this term or any child term occurs in a dataset. Following the notation in information theory, the information content (IC) of a term c can be quantified as follows:

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

where $P(c)$ is the probability of encountering an instance of term c.

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 $P(c)$ 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", GO:0003673) and take, for example, "cellular component" (GO:0005575) as our root node instead. $P(c)$ is simply computed using maximum likelihood estimation:

$$P(c) = \frac{\text{freq}(c)}{N} \tag{2}$$

where N is the total number of terms occurring in the dataset and freq(c) is the number of times term c or any child term of c occurs in the dataset.

As the GO allows multiple parents for each term, two terms can share parents by multiple paths. We take the minimum $P(c)$, if there is more than one parent. This is called P_{ms}, for *probability of the minimum subsumer* [20]:

$$P_{\text{ms}}(c_i, c_j) = \min_{c \in S(c_i, c_j)} P(c) \tag{3}$$

where $S(c_i, c_j)$ is the set of parental terms shared by both c_i and c_j. Given these probabilities, Lin [19] developed a similarity measure. It defines the similarity of two terms c_i, c_j as follows:

$$\text{sim}_{\text{Lin}}(c_i, c_j) = \frac{2 \ln P_{ms}(c_i, c_j)}{\ln P(c_i) + \ln P(c_j)} \tag{4}$$

It is obvious that $P_{ms}(c_i, c_j) \geq P(c_i)$ and $P_{ms}(c_i, c_j) \geq P(c_j)$. Thus, values for $\text{sim}_{\text{Lin}}(c_i, c_j)$ vary between 1 (for similar terms) and 0 (for unsimilar terms).

Given the similarity score $\text{sim}_{\text{Lin}}(c_i, c_j)$, one can easily transform the similarity into a distance, such that the distance of two classes c_i, c_j is defined as follows:

$$d_{\text{Lin}}(c_i, c_j) = \text{sim}_{\text{Lin}}(c_i, c_i) + \text{sim}_{\text{Lin}}(c_j, c_j) - 2(\text{sim}_{\text{Lin}}(c_i, c_j)) \tag{5}$$

It is obvious that $d_{\text{Lin}}(c_i, c_j)$ varies between 0 and 2. Since genes are often annotated with more than one GO term, we needed to combine the calculated similarities or distances. On previous work, based on WordNet [8], a similar problem was found, as individual words have more than one meaning [28]. In this case the maximum similarity, corresponding to the minimum distance was taken, as generally only a single word meaning is used at a time. In contrast, Lord *et al.* [20] used average values. They argued that in contrast to WordNet, a gene product will generally have all of the roles attributed to it. Although this is a good argument we use the best distance d_{Lin} calculated on maximum similarities, because in previous experiments, we got much better results with these distances.

IV. The Clustering algorithm: MST-MA

Many popular clustering algorithms are based on calculating cluster means (e.g. SOMs and k-means). In our case, we cannot calculate means and also want to avoid it, since it might become difficult and computationally very expensive in directed graphs. Therefore, the clustering algorithm has to satisfy a major criterion: no mean calculation should be used. The most popular type of clustering algorithms, which do not need means, are hierarchical methods, especially Average

Linkage clustering. In [32] we presented a Memetic Algorithm (MA) based on Minimum Spanning Trees (MST) that highly outperformed this method and also does not use means. Therefore, we use this algorithm called MST-MA. The basic idea of the MST-MA is to build an MST from the dataset and find so called inconsistent edges in the tree to cut and thus build the resulting clustering. In the next section we will review the MST-MA briefly.

A. Memetic Algorithms

Memetic Algorithms, and Genetic Algorithms in general, are population-based heuristic search approaches and have been applied in a number of different areas and problem domains, mostly combinatorial optimization problems. It is known that it is hard for a 'pure' Genetic Algorithm to 'fine tune' the search in complex spaces [7]. It has been shown that a combination of global and local search is almost always beneficial [21]. The combination of an Evolutionary Algorithm with a local search heuristic is called Memetic Algorithm [23]. MAs are known to exploit the correlation structure of the fitness landscape of combinatorial optimization problems [21], [22]. They differ from non-hybrid evolutionary approaches in that all individuals in the population are locally optimized, since after each variation step, a local refinement is applied.

MAs are inspired by Dawkin's notion of a *meme* [7]. A *meme* is a "cultural gene" and in contrast to genes, *memes* are usually adapted by the people who transmit them before they are passed to the next generation. From the optimization point of view, it is argued that the success of an MA is due to the tradeoff between the exploration abilities of the underlying EA and the exploitation abilities of the local searchers used. This means that during variation, the balance between disruption and information preservation is very important: on the one hand, the escape of local optima must be guaranteed, but on the other hand, disrupting too much may cause the loss of important information gained in the previous generation. The pseudocode of a Memetic Algorithm is given in Fig. 2.

Algorithm MA:
begin
  $t := 0$;
  $P(t) :=$ initPop();
  $P(t) :=$ localSearch($P(t)$);
  evaluateFitness($P(t)$);
  while *(stopping criteria not met)* **do**
    $P'(t) :=$ selectForVariation($P(t)$);
    $P'(t) :=$ recombine($P'(t)$);
    $P'(t) :=$ mutate($P'(t)$);
    $P'(t) :=$ localSearch($P'(t)$);
    evaluateFitness($P'(t)$);
    $P(t+1) :=$ selectNewPop($P(t)$,$P'(t)$);
    $t := t + 1$;
  end
end

Fig. 2: Pseudocode of a standard Memetic Algorithm.

B. Minimum Spanning Trees

As described earlier we use a Minimum Spanning Tree (MST) to represent the dataset. Let $X = \{x_1, \ldots, x_n\}$ be a set of genes. Let $G(X) = (V, E)$ be an undirected weighted and complete graph, with $V = \{x_i | x_i \in X\}$ being a set of vertices (in our case genes) and $E = \{x_i, x_j | x_i, x_j \in X \wedge i \neq j\}$ a set of edges connecting the genes. Each edge $(u, v) \in E$ has been assigned with a weight $w(u, v)$ that represents the dissimilarity between u and v. We use the functional distance measure based on the Gene Ontology as dissimilarity (distance) measure. A tree is a connected graph with no circuits and a spanning tree T of a connected weighted graph $G(X)$ is a weighted tree of $G(X)$ that contains every vertex of $G(X)$. If we define the weight of a tree to be the sum of its edge weights, an MST is a spanning tree with minimum total weight. An MST can be computed using either Kruskal's [17] or Prim's algorithm [24] in $O(|E| \log |E|)$ and $O(|E| \log |V|)$ time, respectively, $|\cdot|$ denoting the number of elements in the set. We decided to use Prim's algorithm, since it is faster for fully connected graphs. For details on the algorithm and its implementation see [4].

By utilizing this MST representation we transform the multi-dimensional clustering problem (that is usually defined as finding the best partition $P(X)$ according to an objective function) into a tree partitioning problem: finding a set of tree edges and deleting them, so that the resulting unconnected components determine the clustering. Representing a multi-dimensional dataset as a relatively simple tree structure leads to a loss of information. In [32] our results indicated that no indispensable information is lost that is needed to solve the clustering problem. Instead, the MST representation of the dataset allows us to deal with clusters of complex shapes, with which classical algorithms, which are based on the idea of grouping the data around a center, have problems.

C. Representation of an individual and Initialization

The representation used in the MA resembles the one in Genetic Algorithms, since we reduced the multi-dimensional clustering problem to a binary tree partitioning problem: First, the MST is computed once using Prim's [24] algorithm and then copied to each individual. The individual itself is represented as a bit vector of length $n - 1$, with n denoting the number of genes. Each bit corresponds to an edge of the MST indicating whether the edge is deleted (0) or not (1). The resulting cluster memberships can then be calculated from the MST partition.

To initialize the population, $k-1$ edges are randomly chosen according to a uniform distribution and deleted from the MST, with k denoting the number of clusters.

D. Fitness Function

A common fitness function for clustering is the minimum sum of squared error (SSE) [15], the sum over all the squared distances to the respective cluster mean. Since we cannot

calculate means, we use the total distance between all items in the cluster. Therefore, our fitness function is defined as follows:

$$\min \sum_{i=1}^{k} \sum_{x,y \in C_i, x \neq y} d(x,y) \tag{6}$$

where $d(\cdot,\cdot)$ denotes the functional distance between gene x and gene y, and k is the number of clusters. In contrast to the SSE function, we do not use squared distances, because in previous experiments, we did not receive significantly different clustering results by using squared distances.

E. Local Search

The local search works as follows: for each individual a list of deleted and non-deleted edges is created. During each step, a deleted and a non-deleted edge is chosen randomly. Then both states of the edges are reversed, the deleted becomes undeleted and vice versa, if the resulting clustering has a smaller objective value according to Eq. (6). This procedure is repeated until no enhancement could be made or one of the two lists is empty. Since for each deleted edge a non-deleted edge is reversed as well, the number of clusters is preserved during local search.

F. Selection, Recombination and Mutation

Selection is applied twice during the main loop of the algorithm: selection for variation and selection for survival. For variation (recombination and mutation) individuals are randomly selected without favoring better individuals. To determine the parents of the next generation, selection for survival is performed on a pool consisting of all parents of the current generation and the offspring. The new population is derived from the best individuals of that pool. Hence, the selection strategy is similar to the selection in a $(\mu + \lambda)$-ES [2]. To guarantee that the population contains each solution only once, duplicates are eliminated.

As recombination operator we use Allelic Recombination [5]. In this case, it works as follows: First the edge-bit vector of parent a is copied to the child. Thus, it is guaranteed that all alleles are there at least once. Then, for both parents, lists of their deleted edges are created. For each pair of deleted edges (one from each parent), with equal probability either the deleted edge of parent a or the one of parent b is chosen to be inherited to the child. If the edge of parent b has been chosen and if it isn't already deleted in the child, it is now deleted. At the same time, the corresponding edge of parent a, that has already been copied to the child, is undeleted. Otherwise, nothing is done, because the deleted edge (of parent a) has already been inherited to the child in the beginning. This is repeated until both lists are empty. Thus, it is guaranteed that the number of clusters is preserved.

As mutation operator a simple modified point mutation is applied. Since each individual contains much more non-deleted than deleted edges a normal point mutation (just flipping a randomly chosen bit) would lead to more and more clusters. To preserve the number of clusters, again the two lists with either deleted and non-deleted edges are created. A pair of a deleted and a non-deleted edge is randomly chosen and both are reversed.

V. Calculating Cluster Validities

Beside the biological validation, we want to somehow measure the result of our clustering, thus we need a cluster validity index that can be applied and that does not utilize means. A good cluster validity index should be independent of the number of clusters, thus allowing to compare two clusterings with different number of clusters. At the same time it is desirable that items in one cluster have the minimum possible distance to each other and maximum distance to the genes in other clusters, in other words, we seek clusters that are compact and well separated.

One well known cluster validity index is the Davies-Bouldin (DB) index, which has been defined in [6]. Given a clustering $C = \{C_1, C_2, \ldots, C_k\}$, it is defined as:

$$DB(C) = \frac{1}{k} \sum_{i=1}^{k} \max \left\{ \frac{\Delta(C_i) + \Delta(C_j)}{\delta(C_i, C_j)} \right\} \tag{7}$$

where $\Delta(C_i)$ represents the inner cluster distance of cluster C_i and $\delta(C_i, C_j)$ denotes the inter cluster distance between cluster C_i and C_j. k is the number of clusters. It is clear from the above definition, that $DB(C)$ is the average similarity between each cluster $C_i, i = 1, 2, \ldots, k$, and its most similar one. It is desirable for the clusters to have minimum possible similarity to each other. Therefore, we seek clusterings that minimize $DB(C)$.

Usually, $\Delta(C_i)$ and $\delta(C_i, C_j)$ are calculated as the sum of distances to the respective cluster mean and the distance between the centers of two clusters, respectively. Since mean calculation in a DAG is difficult and computationally expensive, we use the average diameter of a cluster as inner cluster distance and the average linkage between two clusters as inter cluster distance. Thus $\Delta(C_i)$ and $\delta(C_i, C_j)$ are defined as follows:

$$\Delta(C_i) = \frac{1}{|C_i|(|C_i - 1|)} \sum_{x,y \in C_i, x \neq y} d(x,y) \tag{8}$$

$$\delta(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i, y \in C_j} d(x,y) \tag{9}$$

where $d(x,y)$ defines the functional distance between any of the two genes x and y belonging to cluster C_i and C_j, respectively. $|C_i|$ and $|C_j|$ denotes the number of genes included in clusters C_i and C_j, respectively. This validity index has the advantage that it also provides a value $DB(C_i)$ for each cluster. Therefore, one cannot only use it to compare whole clusterings, but also to distinguish more compact clusters from less compact ones in the same clustering.

VI. Results

The system was implemented in Java 1.4. For the GO graph, the MySQL database implementation, release December 2003, was used. The performance of our functional MST-MA clustering algorithm is discussed on two real world datasets.

A. Datasets

One possible scenario where researchers would like to group a list of genes according to their function is when they examine gene expression with DNA microarray technology, afterwards do some filtering or statistical analysis and end up with a list of genes that show a significant change in their expression according to a control experiment. Because of that it is likely that these genes play an important role during the ongoing examined biological processes. Therefore, we chose two publicly available microarray datasets, annotated the genes with GO information and used them for functional clustering.

The authors of the first dataset [14] 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 [14]. We used these 517 genes for which the authors provide NCBI accession numbers. The GO mapping was done via GeneLynx [10] ids. After mapping to the GO 288 genes remained. The other 229 genes unfortunately had no GO annotation. Since we are interested in gene function, we only use the taxonomy *biological process* of the GO. Out of the 288 genes, 238 genes showed one or more GO mappings to *biological process* or a child term of *biological process*. These 238 genes were used for the functional clustering. We selected 14 clusters, because we received the best results with that number according to Eq. 7.

In order to study gene regulation during eukaryotic mitosis, the authors of the second dataset [3] examined the transcriptional profiling of human fibroblasts during cell cycle using microarrays. Duplicate experiments were carried out at 13 different time points ranging from 0 to 24 hours. Cho *et al.* [3] found 388 genes whose expression levels varied significantly. Hvidsten *et al.* [13] provide a mapping of the dataset to GO. 233 of the 388 genes showed at least one mapping to the GO *biological process* taxonomie and were thus used for clustering. We selected 10 clusters for the same reason as above.

B. Computational Results

In the experiments, the MST-MA was run with a population size of $P = 40$. The MA was terminated upon convergence or before the 200th generation. The recombination and mutation rate was set to 40% and a single point-mutation per mutation step was applied. The experiments were repeated 50 times and the best solution according to Eq. 6 is shown. Additionally, we did random partitions, took the best out of 400.000 runs (50x40x200) and evaluated them in the same manner.

The Davies-Bouldin (DB) indices and the number of genes per cluster for the functional MST-MA clusterings and the random partitions are shown in Tab. I and Tab. II. For both datasets, the DB indices of the MST-MA clustering are much lower than for the random partition indicating good clusters. Nevertheless, in both cases the MST-MA clustering produces both good clusters with very low DB indices and a little less compact ones with a higher validity index. Nevertheless, all clusters are much better than those of the random partition.

TABLE I

DATASET 1: CLUSTER VALIDITY VALUES FOR THE FUNCTIONAL MST-MA CLUSTERING AND A RANDOM PARTITION.

Cluster	MST-MA		random partition	
	DB(C)	# genes	DB(C)	# genes
1	1.440	20	2.055	17
2	1.692	40	2.056	16
3	1.240	9	1.892	20
4	1.255	9	2.076	13
5	1.500	22	2.084	19
6	1.585	25	2.071	18
7	1.303	21	2.044	18
8	1.692	18	2.084	18
9	1.529	25	2.055	14
10	1.227	12	2.045	17
11	1.233	9	2.005	16
12	1.240	12	2.076	16
13	1.062	8	2.053	18
14	1.254	8	2.071	18
total	1.375	238	2.047	238

TABLE II

DATASET 2: CLUSTER VALIDITY VALUES FOR THE FUNCTIONAL MST-MA CLUSTERING AND A RANDOM PARTITION.

Cluster	MST-MA		random partition	
	DB(C)	# genes	DB(C)	# genes
1	1.713	28	2.040	23
2	1.772	48	2.053	22
3	1.772	14	2.053	21
4	1.729	24	2.029	25
5	1.730	41	2.026	21
6	1.513	9	2.040	25
7	1.307	14	2.025	27
8	1.758	30	2.048	22
9	1.530	10	2.026	20
10	1.523	15	2.025	27
total	1.635	233	2.037	233

TABLE III

DATASET 1: GO ANNOTATION OF THE GENES OF CLUSTER 11.

Cluster 11	
Acc. number	Gene Ontology terms
AA053461	**asparagine biosynthesis** **glutamine metabolism**
R00824	**L-serine biosynthesis** **L-serine metabolism**
AA026314	**tetrahydrobiopterin biosynthesis**
AA025800	**L-serine biosynthesis**
AA043796	**lactose biosynthesis**
N32784	neurotransmitter biosynthesis and storage **nitric oxide biosynthesis** **phenylalanine catabolism**
W44416	drug resistance **glutamine metabolism** **nucleobase, nucleoside, nucleotide and nucleic acid metabolism** **'de novo' pyrimidine base biosynthesis**
N35315	**amino acid metabolism**
AA040861	**UDP-N-acetylglucosamine biosynthesis**

The number of genes per cluster indicate that the low DB indices for the MST-MA are not only due to clusters containing one or two genes, where the DB index would be low per

TABLE IV

DATASET 1: GO ANNOTATION OF THE GENES OF CLUSTER 5.

Cluster 5			
Acc. number	Gene Ontology terms	Acc. number	Gene Ontology terms
R45687	**cell cycle** **mitosis regulation of CDK activity**	AA039640	**regulation of cell cycle** mitosis
N21470	cell adhesion muscle contraction oncogenesis	R15989	**cell growth and/or maintenance** oncogenesis
		AA016305	**cell cycle**
AA019203	**chromosome organization and biogenesis (sensu Eukarya)**	N55327	**cell cycle arrest** negative regulation of DNA replication
AA001025	**cell cycle arrest** **regulation of cell cycle** **cell growth and/or maintenance** response to DNA damage stimulus regulation of transcription, DNA-dependent	W90493	**cell cycle** **DNA replication and chromosome cycle** **mitosis** **mitotic chromosome movement** **mitotic metaphase mitotic metaphase plate congression**
W46792	**cell cycle** **regulation of cell cycle** DNA metabolism oncogenesis regulation of transcription, DNA-dependent regulation of transcription from Pol II promoter	R20750	**cell growth and/or maintenance** DNA methylation inflammatory response oncogenesis transcription from Pol II promoter regulation of transcription from Pol II promoter
T91871	anterior compartment specification oncogenesis posterior compartment specification regulation of transcription, DNA-dependent	R43551	**cell growth and/or maintenance** DNA repair mismatch repair oncogenesis
R40626	**regulation of exit from mitosis** septin assembly and septum formation	N90191	**cell cycle** **mitosis**
R10992	**cell cycle** **mitosis** **mitotic checkpoint**	W74500	**cell cycle** **regulation of cell cycle** **mitosis start control point of mitotic cell cycle**
T48153	**cell cycle** **chromosome organization and biogenesis (sensu Eukarya)** **regulation of mitosis** DNA replication and chromosome cycle **mitosis**	N23941	**cell cycle** **cell cycle arrest** **regulation of cell cycle** induction of apoptosis by intracellular signals negative regulation of cell proliferation oncogenesis **regulation of CDK activity**
AA001916	cell cycle **mitosis**	N80129	heavy metal sensitivity/resistance
	mitotic G2 checkpoint oncogenesis **regulation of CDK activity**	T89175	**cell cycle** **cell growth and/or maintenance**

definition, but can be seen as real good clusters.

Beside the mathematical evaluation of the clusters, we also evaluate them biologically by having a closer look at the actual GO annotation of the genes. Tables III - V show the GO annotation of selected clusters of the first dataset (due to space limitation, we can only show selected clusters of one dataset), including examples of compact and well separated clusters as well as a cluster with an inferior validity value (Cluster 5, see Tab. IV). In all tables, GO terms belonging to the same biological process are printed in bold.

It is clearly visible that genes in clusters with good validity indices are also annotated with the same or similar GO terms. Tab. III shows the GO annotations of cluster 11. It is obvious that every gene is annotated with at least one function involved in amino and nucleic acid metabolism. Another example is cluster 13 (see Tab. V): all genes in that cluster participate in DNA repair and replication. The same holds true for cluster 14 (see Tab. V) where all genes are annotated with a role in protein folding and modification. In cluster 12 (see Tab. V), 9 out of 12 genes are involved in lipid metabolism. Due to space limitation, clusters 1-4, 6-9 and 10 are not shown, but in most of the cases the results are similar: e.g. cluster 3 contains genes that have to do with fatty acid metabolism, cluster 4 genes are involved in protein biosynthesis. Genes of cluster 7 regulate transcription and most genes in cluster 8 have something to do with cell adhesion (all data not shown).

Nevertheless, some clusters, especially those with higher DB indices, contain genes of two different functions (data not shown). Cluster 1 genes are involved in RNA metabolism and / or response to stress. Those of cluster 2 are annotated with at least one of the following three functions: cell proliferation, cell growth and cell-cell signaling. However, cell growth and cell proliferation are not too far away, since a cell first has to grow before it proliferates. Additionally, genes of cluster 6 belong either to signal transduction or to cell adhesion. Cluster 9 contains genes that are mostly annotated with apoptosis, but some are also involved in development. Again, these two functions are not too far apart, since apoptosis often occurs during development. The same holds true for cluster 10 whose genes are involved in immune response or blood coagulation.

TABLE V

DATASET 1: GO ANNOTATION OF THE GENES OF CLUSTER 12, 13 AND 14.

Cluster 12	
Acc. number	Gene Ontology terms
W91979	**cholesterol biosynthesis**
N91268	**lipid metabolism** **steroid biosynthesis**
AA053028	**cholesterol biosynthesis** **cholesterol metabolism** germ-cell migration gonad development
R38619	fucose metabolism
AA053173	**cholesterol biosynthesis** **steroid biosynthesis**
AA045181	**C21-steroid hormone biosynthesis** **cholesterol metabolism** **lipid metabolism** mitochondrial transport **steroid metabolism**
AA045372	**cholesterol biosynthesis** **isoprenoid biosynthesis** **steroid biosynthesis**
AA045283	cell growth and/or maintenance germ-cell migration **lipid metabolism**
AA053331	**cholesterol biosynthesis**
AA044444	glycolysis
AA057761	glycolysis
AA001722	ATP catabolism citrate metabolism coenzyme A metabolism **lipid metabolism**
Cluster 13	
H6337	**DNA repair** **pyrimidine-dimer repair, DNA damage excision**
N22858	chromosome organization and biogenesis (sensu Eukarya) **DNA methylation DNA recombination** **DNA repair**
N68268	**DNA replication** **DNA replication, priming**
W93122	**DNA dependent DNA replication** **DNA replication**
N93479	**DNA replication**
H29274	**DNA repair** **DNA replication double-strand break repair** UV protection
AA053076	**DNA replication**
AA031961	cell cycle regulation of cell cycle cell proliferation **DNA repair** regulation of CDK activity
Cluster 14	
AA043103	**protein modification**
AA004517	**protein modification**
H94471	**protein complex assembly**
AA056621	**protein folding**
N49296	**protein folding**
AA045437	**protein modification**
N98463	**protein modification**
AA026120	**protein modification** regulation of transcription, DNA-dependent

Furthermore, we also want to evaluate biologically clusters with higher DB indices. An example for such a cluster is cluster 5 (see Tab. IV). It contains 22 genes and 19 of them play a role during the cell cycle. This indicates that despite the higher DB index, our MST-MA still finds good functional clusters. Similar results were optained with dataset 2.

In general, one can state that although some clusters are less homogeneous than others, most of the clusters found clearly contain genes that belong to defined biological processes. Additionally, the DB indices indicate a real clustering according to gene functions and no random partition. Thus, our results show that the proposed functional clustering algorithm is able to detect clusters of genes that share similar functions and thus belong to similar biological processes defined by Gene Ontology annotation. So far, no automatic method is known to us that groups genes according to their function. This task is especially important during the analysis of high throughput data, where often huge lists of genes have to be biologically sorted, a work that has been done by hand before. Thus, our method is highly valuable for the analysis of large amounts of genomic data.

VII. DISCUSSION AND FUTURE RESEARCH

In this paper, we presented a new functional clustering algorithm for gene expression data and biological annotation. The biological annotation is based on the Gene Ontology, a tool that is available in most public databases. We showed that our method is able to detect clusters of genes with similar functions. To our knowledge, this is the first automated method that produces a functional clustering of genes. Therefore, we provide a useful tool to replace exhausting and time consuming work that so far is done by hand. Additionally, our algorithm is based on a memetic framework that is generally able to overcome less promising local optima and find more global optimal solutions. Furthermore, in previous work, it has been shown to be superior to other classical non-mean based clustering algorithms [32].

Nevertheless, we recognized some problems that should also be discussed here: for each gene a mapping to the Gene Ontology annotation is needed. In most of the cases the GO annotation is available in public databases, especially when dealing with genes from standard microarrays of large companies that usually provide that kind of annotation to their customers. Nevertheless, there are still some genes that do not have that kind of annotation and that could therefore not participate in such an analysis.

Furthermore, we use best distances for our clustering, which of course produces a loss of available information, since a gene might have more than one function, but only one is used for the clustering. Additionally, the fact that our algorithm sometimes produces clusters that contain genes of two different biological processes is probably also caused by the usage of best distances. Since we build an MST in the beginning, two genes can be linked via a third gene that shares one function with the first and one with the second gene, although both functions may be quite different. One might think that using average functional distances instead, may solve that problem, but previous experiments with that

did not lead to good results. We think, that a fuzzy approach might be an appropriate solution for that problem and we are currently working on that.

Additionally, we want to examine more different biological distance measures than the proposed one in more detail as well as functional similarity measures. One might also think of a similarity based clustering, which is easy to implement with our MST-MA. Also other distance and similarity measures that are not Gene Ontology based could be developed.

In summary, we showed that most clusters found by our MST-MA contain genes annotated with the same or similar functions. This fact enormously facilitates the analysis of high throughput data during which researchers are often forced to simply group a list of genes according to their function. Hence, our proposed method is shown to be highly valuable for clustering genes according to their function and therefore constitutes a good alternative to classical non-automatized procedures.

REFERENCES

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

[2] H.G. Beyer. Toward a theory of evolution strategies: On the benefits of sex - the $\mu/\mu, \lambda$ theory. *Evolutionary Computation*, 1:81–111, 1995.

[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] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 2nd edition, 2001.

[5] C Cotta. A study of allelic recombination. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, volume 2, pages 1406–1413. IEEE Press, 2003.

[6] J.L. Davies and D.W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:224–227, 1979.

[7] R. Dawkins. *The selfish Gene*. Oxford University Press, 1976.

[8] C. Fellbaum. *WordNet. An electronic lexical database*. MIT Press, Massachusetts, Cambidge, 1998.

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

[10] Gene Lynx. http://www.genelynx.org, 2004.

[11] J.J. Goeman, S.A. van de Geer, F. de Kort, and H.C. van Houwelingen. A global test for groups of genes: testing association with a clinical outcome. *Bioinformatics*, 20(1):93–99, 2004.

[12] D. Hanisch, A. Zien, R. Zimmer, and T. Lengauer. Co-clustering of biological networks and gene expression data. *Bioinformatics*, 18 (Supplement):S145–S154, 2002.

[13] 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.

[14] 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.

[15] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1988.

[16] 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.

[17] J.B Kruskal. On the shortest spanning subtree of a graph and the travelling salesman problem. In *Proc. Amer. Math. Soc.*, volume 7, pages 48–50, 1956.

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

[19] D. Lin. An information-theoretic definition of similarity. In Morgan Kaufmann, editor, *Proceedings of the 15th International Conference on Machine Learning*, volume 1, pages 296–304, San Francisco, CA, 1998.

[20] 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*, pages 601–612, 2003.

[21] P. Merz. *Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies*. PhD thesis, Department of Electrical Engineering and Computer Science, University of Siegen, Germany, 2000.

[22] P. Merz. Clustering gene expression profiles with memetic algorithms. In *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature, PPSN VII*, pages 811–820. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2002.

[23] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical report, Caltech Concurrent Computation Program, California Institute of Technology, Technical Report C3P Report 826, 1989.

[24] R.C. Prim. Shortest connection networks and some generalizations. *Bell Sys. Tech. Journal*, pages 1389–1401, 1957.

[25] R. Rada, H. Mili, E. Bicknell, and M. Bletner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, 1989.

[26] S. Raychaudhuri and R.B. Altman. A literature-based method for assessing the functional coherence of a gene group. *Bioinformatics*, 19(3):396–401, 2003.

[27] 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.

[28] P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambigiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

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

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

[31] F. Sohler, D. Hanisch, and R. Zimmer. New methods for joint analysis of biological networks and expression data. *Bioinformatics*, 20(10):1517–1521, 2004.

[32] N. Speer, P. Merz, C. Spieth, and A. Zell. Clustering gene expression data with memetic algorithms based on minimum spanning trees. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, volume 3, pages 1848–1855. IEEE Press, 2003.

[33] N. Speer, C. Spieth, and A. Zell. A memetic co-clustering algorithm for gene exression profiles and biological annotation. In *Proceedings of the IEEE 2004 Congress on Evolutionary Computation, CEC 2004*, volume 2, pages 1631–1638. IEEE Press, 2004.

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