# A Memetic Co-Clustering Algorithm for Gene Expression Profiles and Biological Annotation

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 microarrays, researchers are capable of measuring thousands of gene expression levels in parallel at various time points of the biological process. To investigate general regulatory mechanisms, biologists cluster genes based on their expression patterns. In this paper, we propose a new memetic co-clustering algorithm for expression profiles, which incorporates *a priori* knowledge in the form of Gene Ontology information. Ontologies offer a mechanism to capture knowledge in a shareable form that is also processable by computers. The use of this additional annotation information promises to improve biological data analysis and simplifies the identification of processes that are relevant under the measured conditions.**

## I. Introduction

In the past few years, DNA microarrays have become one of the major tools in the field of gene expression analysis. In contrast to traditional methods, this technology enables the monitoring of expression levels of thousands of genes in parallel [36]. Thus, microarrays are a powerful tool helping to understand the underlying regulatory mechanisms of a cell. A problem inherent in the use of DNA arrays is the tremendous amount of data produced, whose analysis itself constitutes a challenge. Several approaches have been applied to analyze microarray data including principal component analysis [35] as well as supervised [12] and unsupervised learning [10], [32], [33]. In unsupervised learning, clustering techniques are utilized to extract the gene expression patterns inherent in the data and thus find potentially co-regulated genes. Various methods have been applied, such as self-organizing-maps (SOMs) [32], k-means [33] and hierarchical clustering [10]. Evolutionary approaches have also been applied to gene expression data and were shown to be superior to classical clustering algorithms [23], [30].

Although the results of all these approaches are useful, one basic problem remains: none of these methods incorporates known biological information. Therefore, biologists are still forced to do a sequential analysis of their data by first clustering the expression data alone and afterwards annotating the genes of each cluster by hand and thus incorporating biological information into their models. Such an approach is slow and exhausting and may also result in a suboptimal clustering since information from other resources could often help in resolving ambiguities or avoiding errors caused by linkages based on noisy data or spurious similarities. One major problem of pure clustering methods is that cluster boundaries are often close and may also be arbitrary to some degree.

Our work is based on the expectation that the use of the available biological knowledge is essential for the development of powerful automatic methods for the analysis of gene expression data. To our knowledge there are only a few published attempts that make use of additional biological information for the interpretation of gene expression profiles. One of them is to map gene expression clusters determined by pure mathematical clustering onto metabolic networks in order to find pathways of interest [34]. Although this method at least incorporates additional biological knowledge, it still is a sequential data analysis. A sophisticated approach for an integrated non-sequential method is described in [37]. It generates biologically possible pathways and scores them with respect to gene expression measurements. Zien *et al.* also provide a significance measure that is calculated by the comparison to a number of scores for random pathways. Kurhekar *et al.* [18] also propose scoring functions to characterize known pathways at the transcriptional level based on gene expression, co-regulation and cascade effects. They also present an approach for the visualization of gene expression data in metabolic and regulatory pathways using multi-resolution animation.

So far only one attempt is known to us that directly integrates biological information to improve the result of a clustering [13]. Hanisch *et al.* [13] map genes to components of known biological networks and propose a combined distance function to calculate distances between these genes based on both: their position in the biological network and their gene expression profiles. Their results seem promising, but assume an exact knowledge of the relevant regulatory pathways. Because these are usually not easily available, the authors show their algorithm performance on data of metabolic pathways [13]. So far, no approach is known that uses information in a more general sense.

Researchers doing gene expression experiments often have access to genetic network annotation data for the probes on their arrays. This ranges from semi-structured data like keywords of a defined vocabulary to unstructured free text descriptions. Often there is even a large amount of annotation available. This served the community well in the past when the annotation was meant for humans to read. However, it causes difficulties when trying to analyze the annotation computationally since

computational interpretation of text data is hard. Partly because of that there has been growing interest in ontologies within the bioinformatics community. They provide a set of vocabulary terms that label domain concepts and at the same time terms are placed within a structure of relationship. This makes it easily processable by computers.

In this paper we utilize biological knowledge in the form of ontological information and propose incorporating that into the clustering algorithm. The advantage of such a method is that combining pure clustering with biological information may lead to more meaningful clusters in the biological sense and that no prior knowledge about relevant pathways is necessary, except a mapping of the expression profiles to the ontological information. The latter is often available in public databases. At the same time we use a memetic clustering framework, which is generally able to overcome less promising local optima to find globally more optimal solutions. In practice our memetic framework has been shown to be superior in solution quality compared to classical clustering methods [30].

The paper is organized as follows: a brief introduction to the ontological information used, the Gene Ontology (GO), is given in section II. The semantic distance measure used within the ontology and the gene expression distance function are described in section III. In section IV the memetic co-clustering algorithm is described in detail. The performance of our co-clustering algorithm on a real world gene expression dataset is shown in section V. Section VI 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 [6], [7] and is developed from the Gene Ontology Consortium [2]. 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 16000 terms (as of December 2003) and additionally new ontologies covering other biological or medical aspects are being developed.

The GO represents terms within 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 (Fig. 1): 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 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

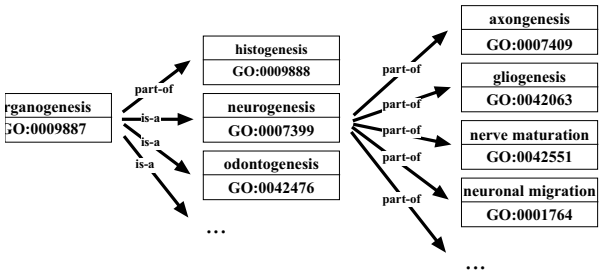


Fig. 1. Relations in the Gene Ontology. Each node is annotated with a unique accession number.

done by computers and not by humans. The GO is available as flat files and XML files at [2] and has also been ported to a MySQL database scheme [7], [3].

## III. Calculating Distances

In the following, we will first define the distance functions for the GO and for the gene expression measurements separately. After that we will explain how we combined these two distance functions.

### A. Distances within the Gene Ontology

There are a couple of semantic similarity measures of different complexity [16], [19], [27], [28], most of them were originally developed for taxonomies like WordNet [11]. In this paper we use an approach based on the information content [28] of each term, originally described in [16] and first adapted to the GO in [20], [21].

The information content of a term (terms are named classes in the following) is defined as the probability with which this term or any child term occurs in a large corpus. Following the notation in information theory, the information content ($IC$) of a term or class $c$ can be quantified as follows:

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

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

In the case of a hierarchical structure, such as the GO, where a class 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 classes occurring in the corpus and $\text{freq}(c)$ is the number of times class $c$ or any child class of $c$ occurs in the corpus.

The similarity of two classes $c_i, c_j$ can then be defined as followed:

$$\text{sim}(c_i, c_j) = -\ln \min_{c \in S(c_i, c_j)} P(c) = -\ln P_{ms}(c_i, c_j) \quad (3)$$

where $S(c_i, c_j)$ is the set of parental classes shared by both $c_i$ and $c_j$. As the GO allows multiple parents for each class, two classes 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* [21]:

$$P_{ms}(c_i, c_j) = \min_{c \in S(c_i, c_j)} P(c) \quad (4)$$

Given the similarity score $\text{sim}(c_i, c_j)$, Jiang *et al.* [16] developed a distance measure, which is the inverse of similarity. They defined the semantic distance of two classes $c_i, c_j$ as follows:

$$d_{sem}(c_i, c_j) = 2 \ln P_{ms}(c_i, c_j) - (\ln P(c_i) + \ln P(c_j)) \quad (5)$$

Since genes are often annotated with more than one GO term, we needed to combine the calculated distances. On previous work, based on WordNet [11], a similar problem was found, as individual words have more than one meaning [29]. 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.* [21] used the average similarity. They argued that in contrast to WordNet, a gene product will generally have all of the roles attributed to it. Since we are using distances not similarities, we will again take the minimum distance, since the average distance mathematically loses basic features of a metric, e.g. that $d(c_i, c_i) = 0$.

### B. Distances between Expression Profiles

For gene expression profiles, several different distance functions have been proposed: Euclidean distance, Manhattan distance as well as Pearson Correlation Coefficient, suggested in [10]. The last is a measure for the degree of linear dependence between two time-courses of gene expression levels and has widely been used for gene expression data. The Correlation Coefficient $\rho$ is defined as follows:

$$\rho(x_i, x_j) = \frac{1}{N} \sum \left( \frac{x_{ik} - \mu_i}{\sigma_i} \right) \left( \frac{x_{j_k} - \mu_j}{\sigma_j} \right) \quad (6)$$

where $x_i$ and $x_j$ are the expression vectors of gene $i$ and $j$, $x_{ik}$ is the expression value of gene $i$ at time point $k$ and $\mu_i$ and $\sigma_j$ denote mean and standard deviation of the measured time series data of gene $i$. $\rho(x_i, x_j)$ takes a value of 1, if gene $i$ and $j$ are totally correlated, 0, if they are not correlated, and -1, if they are anti-correlated. The correlation coefficient can easily be converted to a distance measure $d_{expr}$ in the range $[0, 2]$

$$d_{expr}(x_i, x_j) = 1 - \rho(x_i, x_j) \quad (7)$$

This distance function quantifies the degree of dissimilarity of two genes. We consider anti-correlated genes as most distant. In our purpose the use of correlation as a distance function seems reasonable, since we are looking for genes that participate in the same process. Because of that their expression patterns may depend on each other and thus should be correlated. However, we can not expect to see perfect correlation because of noise and the fact that we do not measure functional gene products, such as proteins, but only mRNA levels. Nevertheless, we could expect correlation of genes belonging to the same pathway.

### C. Combining Distances

The semantic distance function $d_{sem}$ operates on pairs of GO nodes in a DAG, whereas the other distance function $d_{expr}$ operates on RNA expression measurements, often seen as "gene activity" of an organism. To construct a combined distance function, both types of information must be available, thus a mapping $M$ that relates genes of a microarray experiment to nodes in the GO graph is required. For human genes many databases (e.g. SwissProt, TrEMBL and NCBI) provide GO annotation for their entries and also Affymetrix makes flat files accessible to their customers containing GO annotation for their probeset ids. In our case, we used Gene Lynx [1] to map NCBI accession numbers to GO terms. Such a mapping is not one-to-one, a fact that leads to problems: not every expression probe (or gene) can be annotated with at least one GO term. Enforcedly, for single expression measurements a combined distance calculation is not possible, reducing the number of genes that can take part in such a combined analysis.

The combined distance function should assign a small distance to genes that are close in the GO and have similar expression profiles. Genes that are far apart in the GO, but still show similar expression, should be assigned higher distances. The same holds true for genes with highly distinct expression patterns, but that are close in the GO. The largest distances should be assigned to genes, which are far apart according to both measures: expression and GO distance. One function that fulfills these criteria is a simple linear combination of the two distance functions using scaled distances in the range $[0, 1]$. Scaling is essential, to make both distance measures comparable:

$$\begin{aligned} d_{comb}(x_i, x_j) = {} & w_{expr} d_{expr}(x_i, x_j)_{scaled} \\ & + w_{sem} d_{sem}(x_i, x_j)_{scaled} \end{aligned} \quad (8)$$

with $w_{expr} + w_{sem} = 1$, where $w_{expr}$ is a weight defining the amount the distance calculation should be influenced by the pure expression measurement and $w_{sem}$ being the influence of the GO annotation. We use equal amount of both, gene expression and GO distance, thus $w_{expr} = w_{sem} = 0.5$.

## IV. The Co-Clustering algorithm: MST-MA

Many popular clustering algorithms for gene expression data 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 [30] 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 [9]. It has been shown that a combination of global and local search is almost always beneficial [22]. The combination of an Evolutionary Algorithm with a local search heuristic is called Memetic Algorithm [24], [25]. MAs are known to exploit the correlation structure of the fitness landscape of combinatorial optimization problems [22], [23]. They differ from other hybrid evolutionary approaches in that all individuals in the population are local optima, since after each variation step, a local search is applied.
MAs are inspired by Dawkin's [9] notion of a *meme*. 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.

### 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 gene expression data with each $x_i = (x_{i_1}, \ldots, x_{i_m}) \in \Re^m$ denoting the $m$-dimensional data vector of gene $i$ with its expression levels at time $1, 2, \ldots, m$. Let $G(X) = (V, E)$ be an undirected weighted acyclic and complete graph, where $V = \{x_i | x_i \in X\}$ being a set of vertices (in our case genes) and $E = \{x_i, x_j | x_i, x_j \in X \vee i \neq j\}$ 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 combined distance $d_{comb}$ as dissimilarity measure. A tree is a connected weighted graph with no circuits and a spanning tree $T$ of a connected weighted graph $G(X)$ is a 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 [26] 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 [8].
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 [30] 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 [26] 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

One of the two fitness functions used in [30] does not use centroids and therefore is useful for our special clustering purpose. It is defined as follows:

$$\min \sum_{i=1}^{k} \left( \sum_{x_i, x_j \in C_i, i \neq j} \frac{d^2(x_i, x_j)}{|C_i|} \right) \tag{9}$$

where $d(\cdot, \cdot)$ is the combined distance ($d_{comb}$), $|C_i|$ the number of cluster members in cluster $C_i$, $k$ the number of clusters. Eq. (9) is also known as total squared distance measure [14].

### 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. (9). This procedure is repeated until no enhancement could be made or one of

the two lists is empty. Since for each deleted 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 [5]. To guarantee that the population contains each solution only once, duplicates are eliminated.

The recombination operator is a modified uniform crossover, similar to the uniform crossover for binary strings [31]. To preserve the number of clusters, for both parents, lists of their deleted edges are created. Each bit of the child's bit vector is set to 1. Then, a pair of deleted edges (one from each parent) is randomly chosen and deleted from the lists. With a probability of 0.5 either the deleted edge of parent $a$ or the one of parent $b$ is copied to the child. 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. Results

The system was implemented in Java 1.4. For the GO graph the MySQL database implementation [3], release December 2003, was used. The performance of our combined clustering algorithm is discussed on a real world dataset.

### A. Dataset

The dataset used is publicly available at [4]. The authors [15] 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 [15]. We used these 517 genes for which the authors provide NCBI accession numbers. The GO mapping was done via Gene Lynx [1] ids. After mapping to the GO 288 genes remained. The other 229 genes unfortunately had no GO annotation. Since we are interested in knowledge incorporation of ongoing processes, 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 clustering. Their expression vectors were normalized to have a mean of 0 and a variance of 1 as described in [33]. We selected 10 clusters as described in the original paper [15].

### B. Computational Results

In the experiments, the MA was run with a population size of $P = 40$. The MA was terminated upon convergence or before the 400th 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. 9 is shown.

The results of the clustering are shown in Fig. 2 and 3. The 19-dimensional gene expression vectors are visualized as heatmap, red indicating up-regulated genes, black no change and green standing for down-regulation. The expression measures are in the following order: 0 min, 15 min and 30 min, 1 h, 2 h, 4 h, 6 h, 8 h, 12 h, 16 h, 20 h, 24 h after serum stimulation, unsynchronized followed by 30 min, 1 h, 2 h, 4 h and 0 h and after serum and cyclohexamide (a protein synthesis inhibitor) stimulation, unsynchronized with cyclohexamide. Additionally, for each gene the NCBI accession number and a description are provided.

Generally, it is visible that in some clusters the expression profiles are not as similar as they might be with pure mathematical clustering. But still all clusters contain similar expression profiles and there are also clusters with very strong corresponding expression vectors, e.g. cluster 1 and 6. Having a closer look to the genes and their GO annotation of each cluster, some clusters contain very clearly only genes belonging to the same biological process. All genes in cluster 1 except one are belonging to immune processes or stress response. On the expression side, all are slightly up-regulated when serum is added and down-regulated again when the transition from G0 to G1 phase of the cell cycle takes place (after 1 or 2 h). The same holds true for the genes cluster 3, but they differ on the GO side, being involved in the biosynthesis of lipids, amino acids and/or cholesterol. These two clusters are good examples for clusters that do not differ much in expression, but in the function of the genes. Another example of such a case are clusters 5, 6, 7, 8 and 9. They all share a more or less similar expression, being down-regulated or neutral at the beginning and showing a strong up-regulation starting at 8h, but mostly at 16 h after serum stimulation, the startpoint of mitosis [15]. Genes of cluster 5 could be found to have a role in apoptosis whereas those in cluster 6 mostly perform different tasks like initiating mitosis, playing a role in DNA replication and repair as well as chromosome condensation. All genes of cluster 7 belong to the glycolyse pathway, and those of cluster 8 are known to have a role in protein modification and folding. Cluster 9 genes are annotated to play a role in cell adhesion. Thus, we can show that our co-clustering MA is able to separate genes based on their expression profile and their role in a biological process. It is evident, that from the biologists point of view, such a cluster distribution helps a lot to realize the ongoing processes in a cell and simplifies gene expression data analysis.

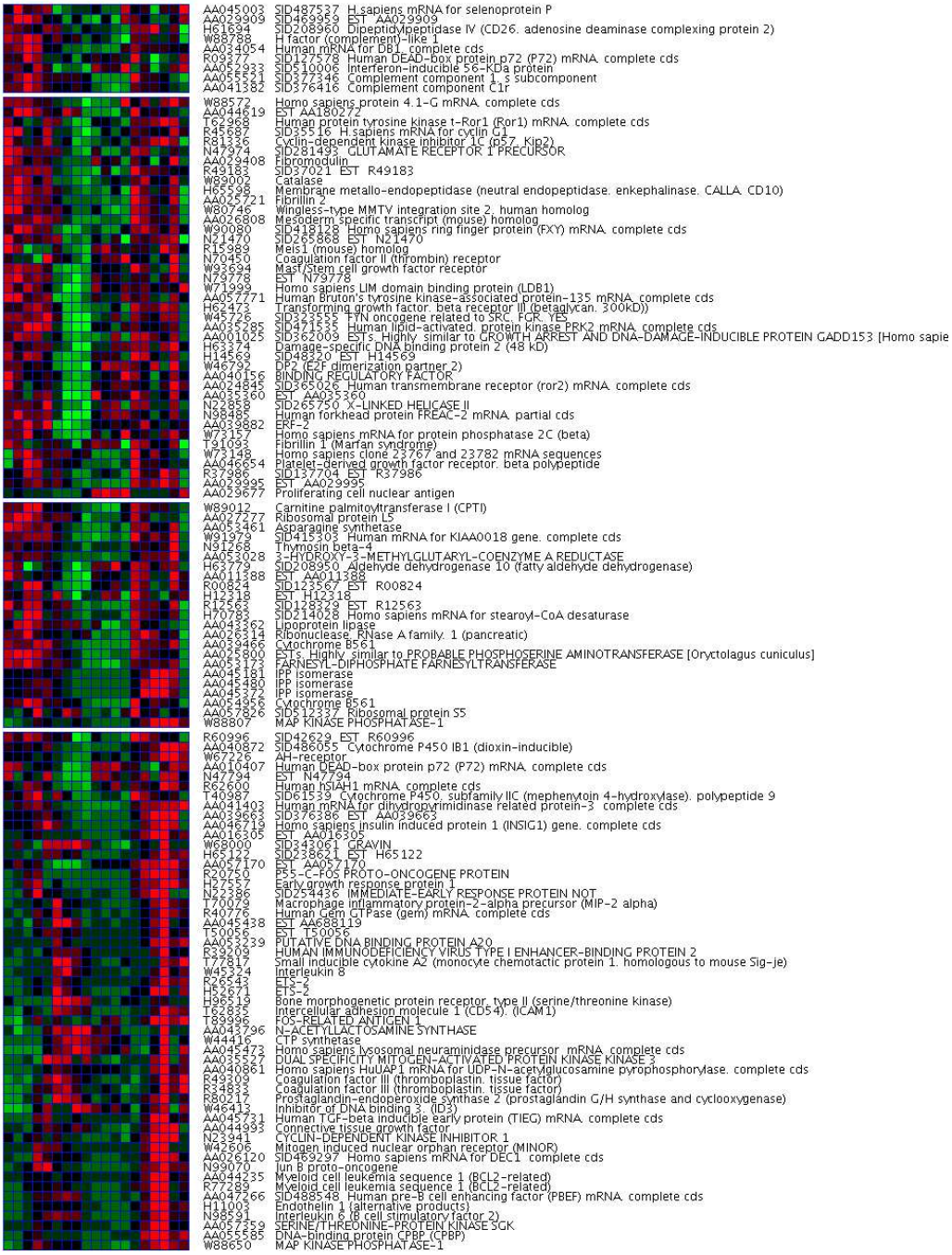


Fig. 2. Clusters 1-4: The squares contain the 19 dimensional expression vector visualized heatmap (red (up-regulated) via black to green (down-regulated)), followed by the NCBI accession number and a gene description. The expression measures are in the following order: 0, 15 and 30 min, 1, 2, 4, 6, 8, 12, 16, 20, 24 h after serum stimulation, unsynchronized followed by 30 min, 1, 2, 4 and 0h and after serum and cyclohexamide (a protein synthesis inhibitor) stimulation, unsynchronized with cyclohexamide.

## VI. Discussion and Future Research

In this paper, we presented a new co-clustering algorithm for gene expression data and biological annotation. The biological annotation is based on the GO, a tool that is available in most public databases. Our algorithm is based on a memetic framework that is generally able to overcome less promising local optima and find more global optimal solutions and has been shown to be superior to classical clustering algorithms [30]. Although our results are very promising and an auspicious attempt to bring more biological knowledge into the field of gene expression analysis, we recognized a couple of problems

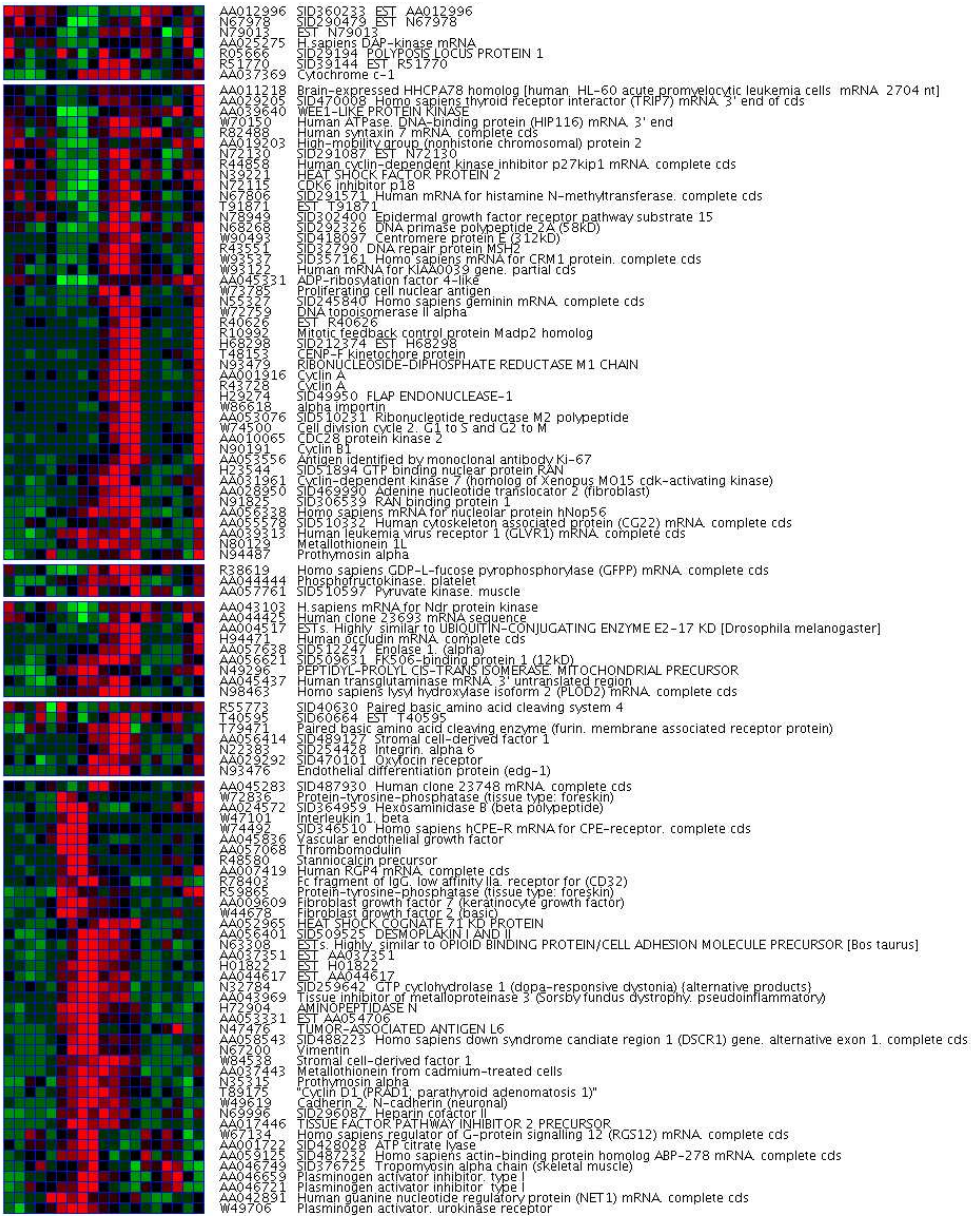


Fig. 3. Clusters 5-10: The squares contain the 19 dimensional expression vector visualized heatmap (red (up-regulated) via black to green (down-regulated)), followed by the NCBI accession number and a gene description. The expression measures are in the following order: 0, 15 and 30 min, 1, 2, 4, 6, 8, 12, 16, 20, 24 h after serum stimulation, unsynchronized followed by 30 min, 1, 2, 4 and 0h and after serum and cyclohexamide (a protein synthesis inhibitor) stimulation, unsynchronized with cyclohexamide.

that should also be discussed here.

One problem is, that many genes, e.g. some of cluster 2 and 10 are only annotated with quite general GO terms like ”cell cycle”. This is of course not detailed enough for biologists examining cell cycle processes. But this deficit is not a general problem of the GO, that provides terms being detailed enough (e.g. ”G2 phase of mitotic cell cycle”), but a problem of the people annotating the genes. Probably with more and more annotation this problem might disappear over the years or may be coped with using hand curated databases like SwissProt. But still we could show that a discrimination between different more general aspects of the GO is possible. A second point is that using only the best and not the average GO distance due to mathematical properties might of course disregard other relations that may also be important. To overcome this problem a similarity and not a distance based clustering algorithm might be helpful, because with similarities one can easily use averages. Another point for future research constitutes the combined distance function. The linear combination of distances proposed in this paper shows

good performance that could very likely be improved using a more sophisticated distance function that separates much more the low distances from the middle and higher ones. Exploring a sigmoidal distance function might be interesting in this context. Furthermore, another field of research would be to develop a centroid calculation method for the GO. It should be tested if the higher computational complexity of centroid calculation in graphs is worth accepting due to a better clustering. At the same time one would be able to expand other standard clustering algorithms with co-clustering features and compare their performance to our memetic framework.

In summary, we showed that the clusters found by our co-clustering MA are separated mathematically as well as biologically. This fact enormously facilitates the gene expression data analysis, since it brings the view to the ongoing biological processes in a cell. Hence, our proposed method is shown to be highly valuable for clustering gene expression profiles and therefore constitutes a good alternative to classical clustering methods.

## REFERENCES


[1] Gene lynx. http://www.genelynx.org.

[2] Gene ontology consortium. http://www.geneontology.org.

[3] Gene ontology database. http://www.godatabase.org/dev/database.

[4] Human fibroblast serum response dataset. http://genome-www.stanford.edu/serum/data.html.

[5] 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.

[6] The Gene Ontology Consortium. Creating the gene ontology resource: design and implementation. *Genome Research*, 11(8):1425–1433, 2001.

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

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

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

[10] M. Eisen, P. Spellman, D. Botstein, and P. Brown. Cluster analysis and display of genome-wide expression analysis. In *Proceedings of the National Academy of Sciences, USA*, volume 95, pages 14863–14867, 1998.

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

[12] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caliguri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer: Class discovery by gene expression monitoring. *Science*, 286:531–537, 1999.

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

[14] D. Harel and Y. Koren. Clustering spatial data using random walks. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 281–286. ACM Press, New York, NY, USA, 2001.

[15] 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.

[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] M.P. Kurhekar, S. Adak, S. Jhunjhunwala, and K. Raghupathy. Genome-wide pathway analysis and visulization using gene expression data. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 462–473, 2002.

[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 across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, 19:1275–1283, 2002.

[21] 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*, 2003.

[22] 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.

[23] 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.

[24] 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.

[25] P. Moscato and M.G. Norman. A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems. In M. Valero, E. Onate, M. Jane, J. L. Larriba, and B. Suarez, editors, *Parallel Computing and Transputer Applications*, pages 177–186, Amsterdam, 1992. IOS Press.

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

[27] 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.

[28] 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.

[29] 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.

[30] 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*, pages 1848–1855. IEEE Press, 2003.

[31] G. Syswerda. Uniform crossover in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 2–9, 1989.

[32] P. Tamayo, D. Slonim, Q. Mesirov, J.AND Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander, and T.R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. In *Proceedings of the National Academy of Sciences, USA*, volume 96, pages 2907–2912, 1999.

[33] S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho, and G.M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.

[34] J. van Helden, D. Gilbert, L. Wernisch, K. Schroeder, and S. Wodak. Applications of regulatory sequence analysis and metabolic network analysis to the interpretation of gene expression data. *LNCS*, 2066:155–172, 2000.

[35] K.Y. Yeung and W.L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17:763–774, 2001.

[36] M. Zhang. Large-scale gene expression data analysis: A new challenge to computational biologists. *Genome Research*, 9:681–688, 1999.

[37] A. Zien, R. Küffner, R. Zimmer, and T. Lengauer. Analysis of gene expression data with pathway scores. In R. Altman *et al.*, editor, *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, pages 407–417, La Jolla, CA, 2000.