Clustering Gene Expression Data with Memetic Algorithms based on Minimum Spanning Trees

Nora Speer, Peter Merz, Christian Spieth, Andreas Zell
University of Tübingen,
Center for Bioinformatics (ZBIT),
Sand 1, D-72076 Tübingen, Germany
nspeer@informatik.uni-tuebingen.de

Abstract- With the invention of microarray technology, researchers are capable of measuring the expression levels of ten thousands of genes in parallel at various time points of the biological process. During the investigation of gene regulatory networks and general cellular mechanisms, biologists are attempting to group genes based on the time-depending pattern of the obtained expression levels. In this paper, we propose a new Memetic Algorithm - a Genetic Algorithm combined with local search - based on a tree representation of the data - a Minimum Spanning Tree - for clustering gene expression data. The combination of both concepts is shown to find near-optimal solutions quickly. Due to the Minimum Spanning Tree representation of the data, our algorithm is capable of finding clusters of different shapes. We show, that our approach is superior in solution quality compared to classical clustering methods.

1 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 [27]. 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 [25] as well as supervised [11] and unsupervised learning [10, 22, 23]. In unsupervised learning, clustering techniques are utilized to extract the gene expression patterns inherent in the data and thus find potentially co-regulated genes. Hierarchical clustering [10] appears to be the most widely used method. It produces a representation of the data in the form of a binary tree, in which the most similar genes are clustered in a hierarchy of nested subsets. In [22] self-organizing-maps (SOM) were used to analyze human hematopoietic differentiation. Tavazoie et al. [23] applied the k-means algorithm to identify clusters in yeast data. Although the results of all these approaches are useful, some basic problems remain: (i) algorithms like k-means and SOM are only capable of detecting clusters of convex shape and generally fail if the clusters are of a more complex, non-convex shape and (ii) most algorithms are simple local search heuristics, which usually converge to the first local optimum. In practice this problem is unsatisfactorily solved by repeating the clustering and then comparing the solutions either by visual inspection or an external cluster index. In this paper, we present a new clustering algorithm that faces these two problems. First, due to a representation of the data as Minimum Spanning Tree, a concept from graph theory, our algorithm is capable of finding clusters of different and complex shapes. Second, since our algorithm is based on a memetic framework, it is able to overcome less promising local optima and find more optimal solutions. In addition we show that our framework is much more effective than classical clustering algorithms in finding near-optimum solutions quickly.

The paper is organized as follows: the MST data representation is described in section 2. A short introduction to Memetic Algorithms is given in section 3, and in section 4 the Memetic Algorithm (MA) proposed is described in detail. In section 5, we present the results of our Memetic Algorithm on gene expression datasets. Furthermore, the MA is compared to other tree based clustering algorithms such as Hierarchical Average Linkage clustering and another MST based method. Section 6 concludes the paper and outlines areas of future research.

2 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 Euclidean distance as dissimilarity measure, but theoretically any other distance measure (e.g. correlational distance, Manhattan

distance) could also be applied. 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 [15] or Prim's algorithm [20] 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 [7].

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 of course leads to a loss of information. But our results indicate 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. The advantage of the MST representation is that it preserves proximity which is the most basic principle of so called Gestalt clusters [26] (see Fig. 1 for an example of differently shaped Gestalt clusters and Fig. 2 for an example of an MST).

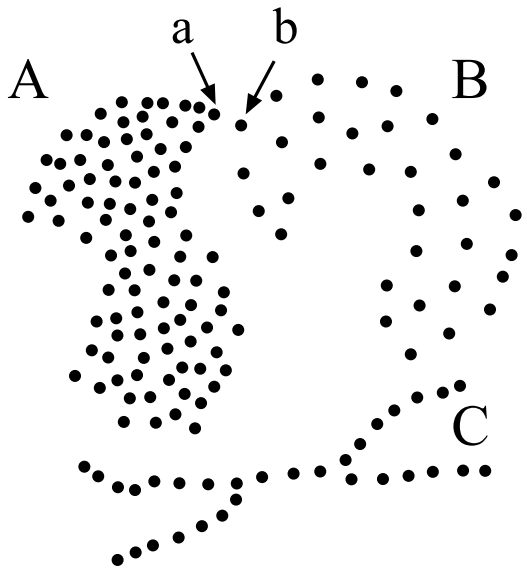

Figure 1: Gestalt clusters with different shapes.

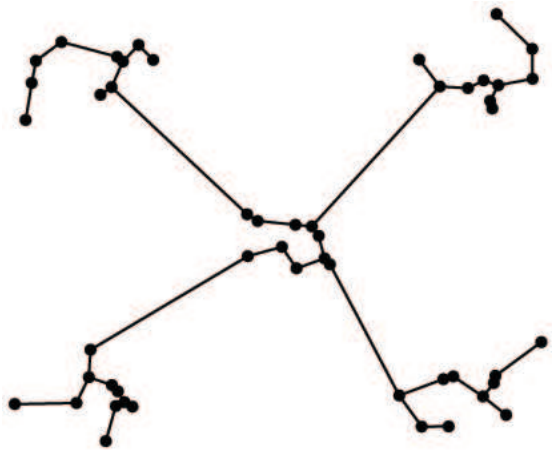

Figure 2: An example of a minimum spanning tree representation of the dataset.

3 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 [16]. The combination of an Evolutionary Algorithm with a local search heuristic is called Memetic Algorithm [18, 19]. MAs are known to exploit the correlation structure of the fitness landscape of combinatorial optimization problems [16, 17]. They differ from other hybrid evolutionary approaches, 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. The pseudocode of a Memetic Algorithm is given in Fig. 3.

Begin
  Initialize population
  Local search
  Evaluate fitness
  While (stopping criteria not met) **do**
    Select individuals for variation
    Crossover
    Mutation
    Local search
    Evaluate fitness
    Select new population
  od
end

Figure 3: Pseudocode of a standard Memetic Algorithm.

4 The clustering algorithm: MST-MA

The memetic clustering algorithm described in this paper is based on a conceptual framework also used in [17].

4.1 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 using Prim's [20] algorithm. Then, for each edge in the MST a mutual neighborhood value [12] (*mnv*) is calculated that will be used in the local search. The concept of the *mnv* will be explained later. For a given dataset, both the MST and the *mnv*s are only computed once 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.

4.2 Fitness Function

The objective function to optimize has to satisfy several criteria: (i) Since we are not only looking for spherical cluster shapes, it should not be based on calculating the distance to a centroid, because that would inevitably lead to convex clusters regardless of the fact, that with the MST representation it is possible to find more complex shaped groups. However, (ii) it should prefer compact clusters to less compact clusters, since in clusters shaped too strangely the individual gene expression profiles might differ too much to be similar in the biological and mathematical sense (see cluster C in Fig. 1 for an example of a cluster that is highly diverse and therefore not interesting for biologists). Two well-known objective functions used for clustering purpose, the sum-of-squared-error criterion [17],

$$\min \sum_{i=1}^{k} \sum_{x_j \in C_i} d^2(x_j, \hat{x}_i), \text{ with } \hat{x}_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j \quad (1)$$

and the Davies-Bouldin-Index [8], which minimizes intra-cluster and maximizes inter-cluster distances, are both based on calculating distances to a cluster center. Hence, we decided to use another function that satisfies the two criteria mentioned above:

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

where in both equations $d(\cdot,\cdot)$ is the Euclidean distance, $|C_i|$ the number of cluster members in cluster C_i, k the number of clusters and p a term to penalize results including clusters with less than a defined number C_i of members. Eq. (2) (with $\forall C_i, p = 0$) is also known as total squared distance measure [13].

4.3 Local Search

The local search applies the concept of the mutual neighborhood value (*mnv*) [12]: Let x_i and x_j be two points (genes) of the given dataset. If x_i is the mth nearest neighbor of x_j and x_j is the nth nearest neighbor of x_i, then the mutual neighborhood value is $m+n$. The main motivation for this definition is that two points x_i and x_j have a higher tendency to group together if not only x_i is close to x_j, but also x_j is close to x_i. Referring to the MST, the idea is, that edges with a higher *mnv* might be those that separate two clusters. For example consider the two points a and b in Fig. 1. A human eye would clearly recognize that a belongs to cluster A and b to cluster B. The nearest neighbor of b is a, but the first two nearest neighbors of a are two points out of A, so that the *mnv* of the edge (a,b) would be high, although the weight of the edge might be low. The *mnv* is a semi-metric and does not satisfy the triangle inequality.

The local search works as follows: for each individual a list of deleted and non-deleted edges is created. During each step, a pair of a deleted and a non-deleted edge is chosen randomly. For the non-deleted edges, edges are favored with a higher *mnv* and for the deleted edges those with a lower *mnv* are favored. 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. (2). This procedure is repeated until no enhancement could be made or the two lists are empty. Since for each flipped deleted edge a non-deleted edge is flipped as well, the number of clusters is preserved during local search.

4.4 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 [4]. 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 [21]. 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 deleted and non-deleted edges are created. A pair of a deleted and a non-deleted edge is randomly chosen and both are flipped.

5 Results

We compare our memetic clustering algorithm to two other clustering methods. The first seems to be the most widely used method to cluster gene expression data: the average linkage algorithm (for details see [10]). The algorithm is denoted as AvgLink. The second is also based on an MST-representation of the dataset, has recently been published in [24] and is denoted as Best2Partition. The latter algorithm works as follows: first, delete randomly $k-1$ edges from the MST. Then for each pair of adjacent clusters that are connected by an edge, go through all the edges within the two merged clusters and cut that one, that globally optimizes the 2-partitioning of the merged cluster, measured by an objective function. Xu et al. applied the squared error function defined in Eq. (1). All algorithms were implemented in Java 1.4. The performance of our MA is shown on four publicly available datasets often used as sample datasets for gene expression clustering purpose.

5.1 Datasets

The first dataset denoted as Y-SP is available at [3] and has been produced by Chu et al. [5]. They used DNA microarrays to analyze the transcriptional program of sporulation in budding yeast. The chip contains 6118 genes and the mRNA levels were measured at seven time points during the sporulation process. Chu et al. found about 1143 genes with significant changes in their expression. Significant meant that the root mean square of the $\log_2(R)$ was greater than 1.13, where R is the measured ratio of each gene's mRNA level to its mRNA level in vegetative cells just before transfer to sporulation medium. For further details see [5]. We applied a variation filter, which discarded all genes with an absolute change less or equal to 3 of the $\log_2$-transformed expression values, and used the resulting 375 genes for clustering. After filtering the vectors were normalized to have a mean of 0 and a variance of 1 as described in [23]. We chose 15 clusters, which appeared to be a reasonable number for 375 genes.

The second dataset is denoted as H-HD and is also publicly available at [2]. The authors [22] used Affymetrix chips and examined the hematopoietic differentiation across 4 different cell lines (HL-60, U937, NB4 and Jurkat). After removing the spiked control genes, the data consists of 7225 genes measured at 17 different time points and cell lines, respectively. We applied an absolute variation filter that discarded all genes with an absolute change in expression level less or equal to 30 and an expression level of max/min < 3. After filtering 773 genes remained and their expression vectors were normalized to have a mean of 0 and a variance of 1 as described in [23]. We selected 30 clusters as described in [22].

The third dataset is denoted as H-FB and is publicly available at [1]. The authors [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 clustering, their expression vectors where normalized as described above (mean of 0; variance of 1). We selected 10 clusters as described in [14].

The fourth dataset is denoted as H-MYC and is described in [6]. The authors [6] used Affymetrix chips and examined the effects of c-myc activation in human fibroblasts. After removing the spiked control genes, the dataset contains expression levels of 7236 human genes and ESTs measured under 4 different conditions repeated in 3 identical experiments leading to a 12 dimensional space. We used a variation filter which discarded the genes with an absolute change in expression level of less than 50 and an expression level of max/min < 2. The resulting number of genes was 498. Again the vectors were normalized as described above.

Dataset	Algorithm	Best Obj.	Avg. Obj.	Best SSE	Avg. SSE	Avg. Time [s]
Y-SP	MST-MA	264.6	269.2	264.6	269.2	61.3
	Best2Partition	271.2	301.4	261.2	280.0	5.2
	AvgLink	391.9	391.9	321.9	321.9	2.0
H-HD	MST-MA	4392.0	4421.7	4382.0	4418.7	216.8
	Best2Partition	4404.5	4503.7	4397.4	4490.2	45.6
	AvgLink	4984.5	4984.5	4884.5	4884.5	5.8
H-FB	MST-MA	3510.5	3538.0	3510.5	3538.0	96.1
	Best2Partition	3512.2	3626.9	3521.2	3626.2	8.1
	AvgLink	3928.9	3928.9	3888.9	3888.9	4.5
H-MYC	MST-MA	1347.5	1369.5	1347.5	1369.5	100.2
	Best2Partition	1349.5	1398.3	1349.5	1389.7	10.8
	AvgLink	1808.9	1808.9	1748.9	1748.9	2.0

Table 1: Comparison of MST-MA and two other clustering algorithms on four datasets.

We selected 15 clusters which appeared to be a reasonable number.

5.2 Computational Results

In the experiments, the Memetic Algorithm 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 penalty p was set to 10 if $|C_i| < 5$, which turned out to be a reasonable number. To compare the methods, their computational time should be the same. For example, on dataset Y-SP, MST-MA needed 61.3 s, which is about 12 times as long as Best2Partition. Therefore, comparisons were made by performing Best2Partition 12 times and selecting the best solution to be compared with the solution by MST-MA. Analogously, Best2Partition was performed 5 times on H-HD, 12 times on H-FB and 10 times on H-MYC. The experiments were repeated 20 times on each dataset.

The results of the tests are shown in Tab. 1. For each algorithm the average (Avg. Obj.) and the best objective function value (Best Obj.) according to Eq. (2) and the average computation time for a single run are given (measured on a PC with an AMD Athlon (1.2GHz) and 512 MB RAM compiled with the j2sdk 1.4 compiler from Sun Microsystems). Additionally, for all tests the best and the average sum-of-squared-error (SSE) defined in Eq. (1) are provided, although MST-MA optimizes Eq. (2). This is done because Best2Partition optimizes Eq. (1) and in addition the SSE is a widely used objective function for clustering purpose.

It is evident that the MA outperforms the other two algorithms according to both functions. Although on the first dataset (Y-SP) Best2Partition is slightly better than the MA regarding the best solution found according to Eq. (1), we can show that the MA is superior concerning the other objective function (Eq. (2)) and in average solution quality on both objective functions. We show that for larger datasets (H-FB, H-HD and H-MYC) the MA is more efficient, especially in average solution quality. This is not surprising since MAs are combinatorial optimization methods and with a growing number of genes, the clustering problem becomes more complex. The MA outperforms both other methods according to both objective functions in best and average solution quality. Furthermore, the average linkage algorithm shows minor performance on all datasets compared to the MST based methods.

From the biologists point of view, the distribution of genes to the clusters is also important. As an example, the number of genes belonging to each cluster for the best solution found on the first dataset (Y-SP), is displayed in Fig. 4-6. In the clustering presented by AvgLink, the distribution of genes per cluster is quite asymmetric: two clusters contain only one gene, two clusters contain two, two clusters three, one four and on the other hand there is one cluster with 83 and one with 91 genes (Fig. 6). This distribution explains the high objective function values, especially if the larger clusters are not compact.

Both MST based methods produce compact clusters as shown in Tab. 1, but the MA still better than Best2Partition. For clustering gene expression data, it is important that the clusters can be differently sized, meaning that the genes should not be totally equally distributed, but on the other hand the clusters should be reasonable, not only containing one or two genes. Fig. 4-6 show that the MST based methods are able to find such clusters. Furthermore, MST-based methods are in general capable of finding differently shaped clusters.

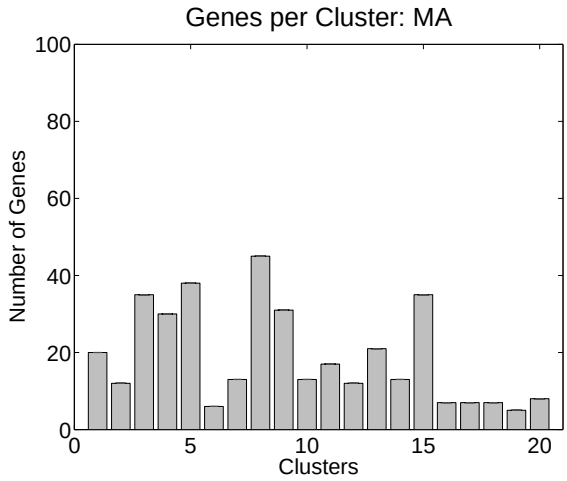

Figure 4: Best clustering solution found by the MA: the number of genes per cluster is displayed.

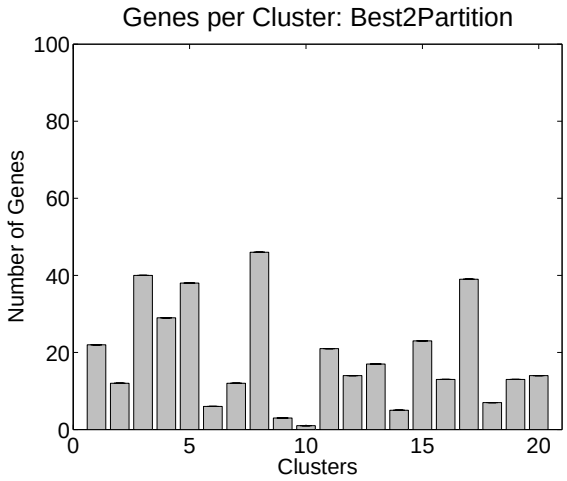

Figure 5: Best clustering solution found by Best2Partition: the number of genes per cluster is displayed.

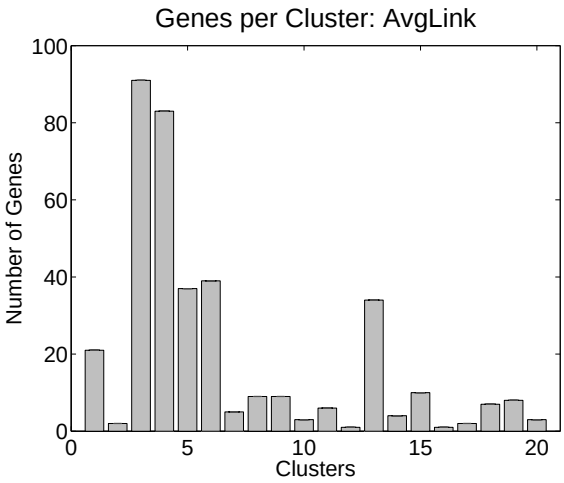

Figure 6: Best clustering solution found by AvgLink: the number of genes per cluster is displayed.

6 Conclusions and future research

In this paper, a Memetic Algorithm in combination with a data representation as a minimum spanning tree was proposed. Thereby, we reduced the multi-dimensional clustering problem to a tree partitioning problem, which is solved on the basis of a memetic framework using a sophisticated local search approach. Our method is not restricted to finding clusters of a spherical shape, which is important if one wants to explore the clusters inherent in the data itself. On the other hand, we could outline that although our method is not mean based, compact clusters can be found, an important task, especially for clustering gene expression profiles. In particular, we demonstrated that our MA outperforms two other clustering methods such as the widely used average linkage clustering and another MST-based method. We show that the clusters found by the MA are more compact and reasonably sized. Hence, our proposed MST based MA is shown to be highly valuable for clustering gene expression profiles and therefore constitutes a good alternative to classical clustering methods.

For future research it would be very interesting to explore the clusters found by the MA in more detail. Furthermore, the determination of the number of clusters, could also be solved by the MA itself, having individuals representing clustering solutions with different numbers of clusters. Alternatively, one could think of sub-populations, each

representing several individuals with a defined number of clusters.

Bibliography

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

[2] Human hematopoietic cell differentiation dataset. http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi.

[3] Yeast sporulation dataset. http://cmgm.stanford.edu/pbrown/sporulation.

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

[5] S. Chu, DeRisi J., M. Eisen, J. Mullholland, D. Botstein, P.O. Brown, and I. Herskowitz. The transcriptional program of sporulation in budding yeast. *Science*, 282:699–705, 1998.

[6] H.A. Coller, C. Grandori, P. Tamayo, T. Colbert, E.S. Lande, R.N. Eisenman, and Golub T.R. Expression analysis with oligonucleotide microarrays reveals that myc regulates genes involved in growth, cell cycle, signaling, and adhesion. *PNAS*, 97:3260–3265, 2000.

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

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

[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] 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.

[12] K.C. Gowda and G. Krishna. Agglomertative clustering using the concept of mutual nearest neighborhood. *Pattern Recognition*, 10:105–112, 1978.

[13] 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.

[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] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

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

[22] 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.

[23] 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.

[24] Y. Xu, V. Olman, and Xu D. Clustering gene expression data using a graph theoretic approach: an application of minimum spanning trees. *Bioinformatics*, 18:536–545, 2001.

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

[26] C.T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, C-20:68–86, 1971.

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