

**Gene Expression Data Clustering with Minimum Spanning Trees**

Mazlum ERYILMAZ

*Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, Buca, 35160, İzmir*

*E-mail:* [*mazlum.eryilmaz@ceng.deu.edu.tr*](mailto:mazlum.eryilmaz@ceng.deu.edu.tr)

**ABSTRACT**

The rapid advance of biotechnology studies and as probably the most explosively expanding tool for genome analysis, microchips of gene expression have made it possible to simultaneously monitor the expression levels of tens of thousands of genes under diﬀerent experimental conditions. Analyzing the gene data, which is produced by gene expression profiling, offers potential insight into cellular functions of genes. One of the first steps in gene expression analysis is the detection of groups of genes that show similar expression patterns and are activated by similar agents. This is a algorithmic problem is known as clustering multi-conditional gene expression data.

A new strategy, which uses the minimum spanning tree (MST) representation of gene expression data, is suggested as an alternative to classical clustering algorithms (Hierarchical, K-Means, or SOMs). In this paper, we present a comparative study and performance evaluation for the new MST based clustering algorithms. We compare three different algorithms (Single linkage, Hierarchical, and Global Optimal) on EXCAVATOR framework by using microarray gene expression data of *Saccharomyces cerevisiae* and different similarity metrics (Euclidean and Pearson correlation).

***Keywords:*** Clustering, microarray gene expression, minimum spanning tree.

**ÖZET**

Biyoteknoloji çalışmalarının hızlı ilerlemesi ve muhtemelen genom analizi için en patlayıcı şekilde genişleyen araç olan gen ifadesinin mikroçipleri, farklı deneysel koşullar altında on binlerce genin ifadesi seviyelerinin aynı anda izlenmesini mümkün kılmıştır. Gen ifadesi profili ile üretilen gen verilerinin analizi, genlerin hücresel fonksiyonlarına potansiyel bir bakış sağlar. Gen ifadesi analizindeki ilk adımlardan biri, benzer ifade benzerliği gösteren ve benzer ajanlar tarafından aktive edilen gen gruplarının saptanmasıdır. Bu çok koşullu gen ifadesi verilerinin kümelenmesi olarak bilinen algoritmik bir sorundur.

Gen ifade verilerini asgari tarama ağaçları (ATA) şeklinde gösterimlenmesini benimseyen yeni bir strateji, klasik kümeleme algoritmalarına (Hierarchical, K-means veya SOMs gibi) alternatif olarak önerilmiştir. Bu makalede, ATA esaslı kümeleme algoritmaları için karşılaştırmalı performans değerlendirmeleri sunulmaktadır. Saccharomyces cerevisiae türüne ait mikro-dizi gen ifade verisi ve farklı gen benzerlik ölçütleri (Pearson korelasyonu ve Euclidean uzaklığı gibi) kullanılmıştır. EXCAVATOR yardımıyla, üç farklı ATA kümeleme algoritması (Single-linkage, Hierarchical, ve Global Optimal) kıyaslanmıştır.

***Anahtar kelimeler:*** Kümeleme, mikro-dizi gen ifadesi, asgari tarama ağaçları

**INTRODUCTION**

By the formal definition, *gene expression* is the process by which information from a gene is used in the synthesis of a functional gene product. In the other words, our DNA code or the genotype turns into encoding instructions. These instructions encode mRNAs and proteins which shape our characteristics or phenotype. This famous vital process is also named as *central dogma of life* in the biology literature.

Likewise, *gene expression level* represents the activity of a gene in any given time. Expression level of a gene is generally obtained by measuring the relative amount of mRNA expressed under different experimental conditions –like cell division cycle, sporulation, or temperature shock. Besides, measuring the activity of thousands of genes simultaneously, across different conditions and over time is named as *gene expression profiling*. We obtain a global picture of a cellular function through gene expression profiling.

In the last two decades, various high throughput techniques, like *DNA microarray* *hybridization* (microchips), *serial analysis of gene expression* (SAGE), and *RNA-Sequencing*,have been developed to perform gene expression profiling. However, DNA microarrays come to the forefront in the field by their unique approach to measure huge amount of gene expression in a relatively fast and cost-effective manner.

Although gene expression profiles provide very important knowledge about vital processes of livings, computational analysis of this complex data is known as a challenging issue. At this point, clustering is a key step in gene expression analysis. We can categorize genes by cellular functions by clustering, which may help us to understand the functional properties of unknown genes. By assuming that the genes with the same cellular functions or in the same biological pathway show similar expression patterns and are activated by similar agents, we can infer functions of unknown genes based on the known functions of genes with similar expression patterns.

|  |  |
| --- | --- |
| M.ERYILMAZ | 2 |

**MATERIALS AND METHODS**

Several approaches and algorithms are developed for clustering gene expressions. The most common methods include (i) hierarchical clustering, (ii) K-means clustering, (iii) K-nearest neighbor clustering, and (iv) clustering by self-organizing maps (SOMs). Although these algorithms are useful in most applications, they have some problems like (1) not guaranteeing to produce a globally optimal clustering, (2) heavy dependence on the regularity of the geometric shape, and (3) noise sensitivity (Ben-Dor et al, 1999).

To overcome such problems, Xu et al. suggested a new approach which represents gene expression data as a minimum spanning tree (Xu et al, 2001) The basic idea of MST-based clustering includes two main steps, (1) construction of MST and (2) generating clusters by cutting edges on MST. To construct a MST, a complete graph should be constructed first. In this graph, each node represents a gene and each edge represents the distance between a pair of genes’ expression levels. The length of edges can be calculated by various similarity metrics like Euclidean distance or Pearson correlation. By use of Kruskal’s algorithm (Kruskal, 1956), a MST is obtained from the fully connected graph. Here, the MST represents a tree structure which connects all the nodes together with the minimum total distance. Through this transformation, the problem of clustering multi-dimensional dataset is reduced to a tree partitioning problem. At this point, the clusters are generated by various edge-cutting algorithms like (i) single linkage, (ii) hierarchical, (iii) iterative, or (iii) global optimal.

**MST clustering algorithms**

Apparently, different clustering problems may need different objective functions, in order to achieve the best clustering results. In this respect, there are four main objective functions and their corresponding clustering algorithms. The following algorithms objects to partition a tree into N subtrees (clusters) where N > 0.

***Single-linkage:*** Cluster by simply cutting the longest edges in the MST. It is the fastestmethod and it works well for obvious clusters (which mean the inter-cluster edge-distances are clearly larger than the intra-cluster edge-distances). However, the result may not be the desired one for complicated clusters.

***Hierarchical:*** Minimize hierarchically the sum of the distances between a gene and thecenter of its cluster. Produce a sequence of nested clusters, ranging from singleton clusters of individual points to an all-inclusive cluster. This hierarchy of clusters is often graphically represented by a dendrogram.

***Iterative:*** Minimize the sum of the distances between a gene and the center of itscluster, through an iterative procedure. The clustering result, while reaching a local minimum, may not reach the global optimal solution for the objective function.

***Global optimal:*** Minimize globally the sum of the distances between a gene and thebest representative gene from the cluster. The global optimal solution is guaranteed, but it takes much longer than the other methods.

(Xu et al, 2003)

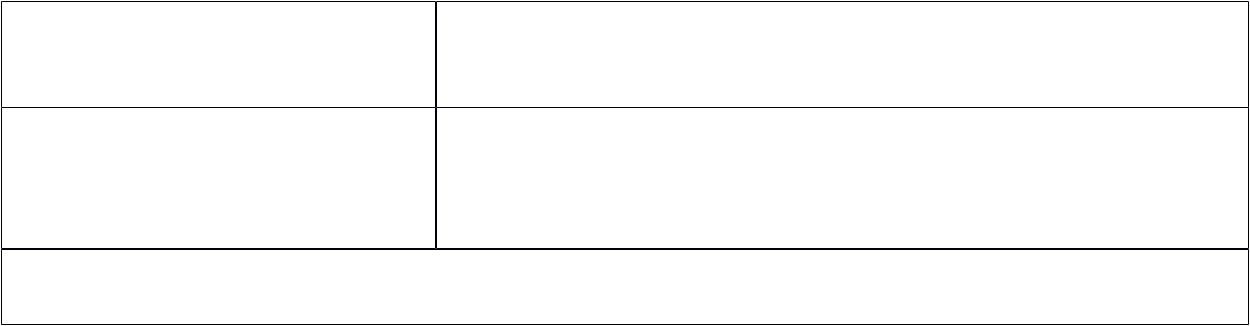
3 Gene Expression Data Clustering with Minimum Spanning Trees



**Similarity Metrics**

The similarity measure is used to calculate the distance between gene expression profiles. Two of the easiest and most commonly used similarity metrics for gene expression data are Euclidean distance and Pearson correlation coefficient. Note that Euclidean distance is sensitive to scaling and differences in average expression level, whereas correlation is not.

**Table 1.** Gene expression similarity measures (D’haeseleer, 2005).



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Euclidean distance | *dfg*=∑( *efc – egc* )2 |  |  |  |  |  |  |
|  | *c* |  |  |  |  |  |  |
|  |  |  |  | ∑ ( *efc* *−e*´*f* )( *egc* *−e*´*g* ) | | |  |
| Pearson Correlation | *dfg*=1 *−r fg* with | *rfg*= |  |  | *c* | |  |
|  |  |  |  |  |
| √ *c* | | *c* | |  |
|  |  |  |  |
|  |  |  |  | ∑ (*efc* *−* ´*ef* )2 ∑ (*egc* *−* ´*eg* )2 | | |  |
| *dfg* , distance between expression patters for genes f and g. *egc* | | | | | , expression level of g under | |  |
| condition c. |  |  |  |  |  |  |  |

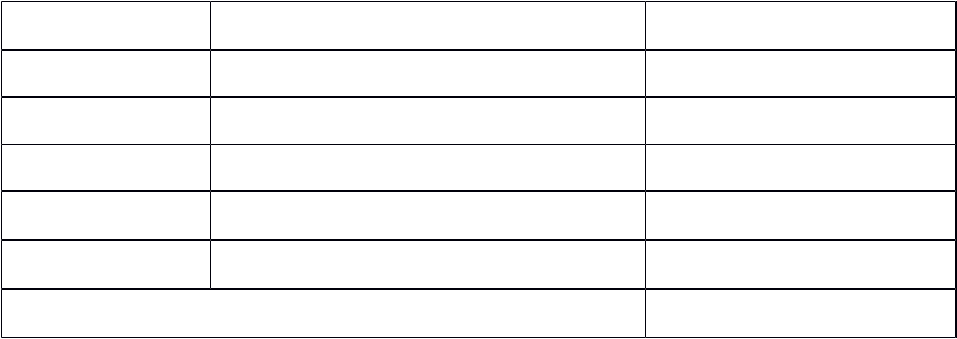
**Yeast Data**

We obtain the combined gene expression dataset for budding yeast (*Saccharomyces* *cerevisiae*) from *Stanford Genomic Resources Database*. The Yeast dataset is commonly usedin this kind of analytical studies since it is the first fully-sequenced eukaryote genome whose sequencing is completed in 1996 and expressions published in 1998. Yeast has 6,178 known genes and functional annotation of 2,467 of them is available. The expression dataset is a combination of different profiling studies on 79 data point (time course) sampled in 8 different conditions. Previously, Eisen et al. studied on 2,476 genes of Yeast dataset and presented 10 clusters according to cellular functions (Eisen et al, 1998).

We select 5 relatively obvious clusters from the dataset as: (1) Protein degradation, (2) Glycolysis, (3) Protein synthesis, (4) Chromatin structure, and (5) Cell cycle. The raw dataset consist of 2467 x 79 expression values which is an excessive amount of data. We removed insignificant genes by applying fold change into each separate cluster with an experimental significance threshold value σ = 1. The threshold value represents σ = log2|Ma-Mc| where Ma is the mean of expression levels of all genes and M c is the mean of expression levels of cluster C. Fold change is often used in this type of large datasets to remove insignificant genes after measuring the amount of expression change of a gene. After fold change, the remaining dataset contains 106 genes whose distribution is shown in Table 2. We compared the clustering results with clusters represented in the work of Eisen et al (Eisen et al, 1998).

|  |  |
| --- | --- |
| M.ERYILMAZ | 4 |

**Table 2.** Class distribution of the dataset after preprocessing.



|  |  |  |
| --- | --- | --- |
| **Cluster No** | **Functional Annotation** | **Number of Genes** |
| 1 | Cell Cycle (CC) | 14 |
| 2 | Protein Degradation (PD) | 34 |
| 3 | Protein Synthesis (PS) | 24 |
| 4 | Glycolysis (GL) | 25 |
| 5 | Chromatin Structure (CH) | 9 |
| **TOTAL** |  | **106** |

**Key Features of Excavator**

EXCAVATOR (Expression data Clustering Analysis and VisualizATiOn Resource) framework has a number of unique features like (1) data-constrained clustering, (2) identification of genes with similar expression patterns, (3) cluster identification from a noisy data, (4) comparison between two clustering results of the same dataset (Xu et al, 2003). EXCAVATOR supports data files with or without annotation column, and can handle with missing data in various ways. It provides wide range of options for distance functions (Pearson correlation, absolute correlation, squared correlation, Euclidean distance, and squared Euclidean) and MST-based clustering algorithms (hierarchical, iterative, global optimal, single linkage). The number of clusters may be pre-defined or selected by computer. EXCAVATOR is also able to visualize the clustering results as heat-maps and binary tree plotting.

**EXPERIMENT RESULTS**

We have tested three different MST-based clustering algorithms on a microarray gene expression data of *Saccharomyces cerevisiae* by sampling 106 most significant genes which belong to 5 relatively obvious clusters (CC, PD, PS, GL, and CH). Note that, the complete graph has 8,376 nodes since there are 79 data points for each gene. We used 2 different distance metrics: Pearson correlation and Euclidean. We compared each algorithm in terms of execution time and accuracy in terms of correctly clustered genes. We set N=5 for clustering with pre-defined number of clusters and N*max*=10 for clustering with undefined number of clusters. The performance evaluation obtained on Intel Core i5-4200U 3.1Ghz PC with 8 Gb of RAM.

5 Gene Expression Data Clustering with Minimum Spanning Trees

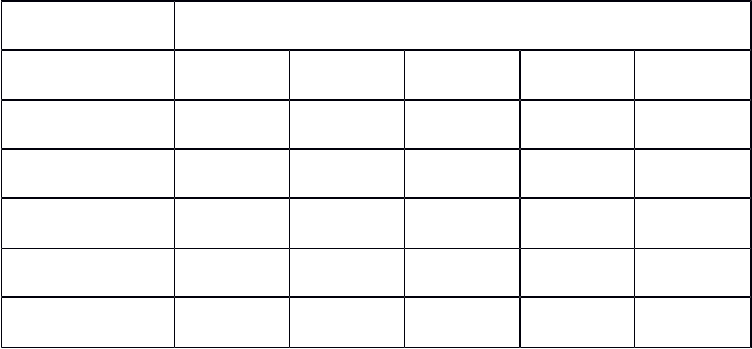


**Single Linkage (Cutting Long Edges)**

First we apply single linkage algorithm (Gower & Ross, 1969) which simply cuts the longest edges until construct the desired number of clusters. When we set the number of clusters pre-defined as 5, 47/106 genes are categorized correctly with Pearson correlation, which means 44.3% of success. However, only 23/106 genes are clustered correctly when we use Euclidean distance, which corresponds to 21.6% of success.

As it is seen in the confusion matrix in Table 2, Pearson correlation is failed to split the genes which are responsible by CC and PD. Also, it is failed to split PS and GL. Worse, single linkage with Euclidean distance (Table 4) placed almost all genes in GL, which shows that it is an inapplicable similarity metric for this algorithm.

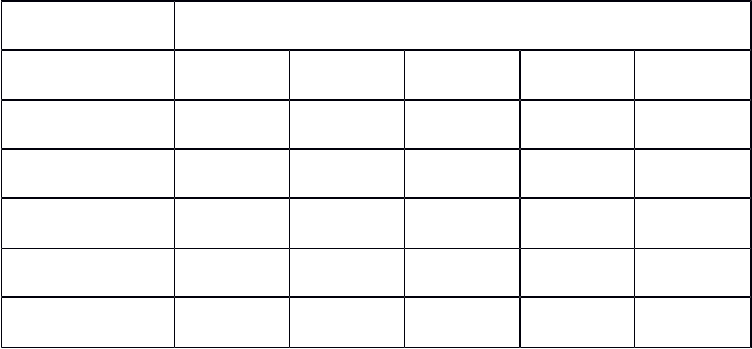
**Table 3.** Confusion matrix for Single Linkage with Pearson correlation.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Expected (True) Cluster | | |  |  |
| Prediction | CC | PD | PS | GL | CH |
|  |  |  |  |  |  |
| CC | 14 | 0 | 0 | 0 | 0 |
| PD | 33 | 1 | 0 | 0 | 0 |
| PS | 0 | 0 | 23 | 1 | 0 |
| GL | 0 | 0 | 25 | 0 | 0 |
| CH | 0 | 0 | 0 | 0 | 9 |
|  |  |  |  |  |  |

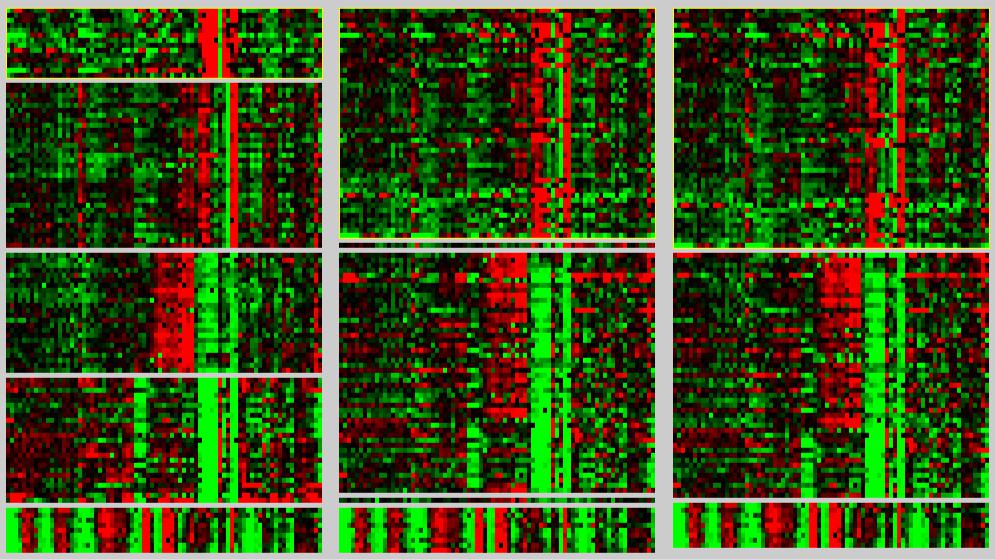
On the other hand, when we do not pre-define the number of clusters and set the maximum number of clusters as 10, the dataset is separated into 3 clusters with use of Pearson coefficient. CC merged with PD and PS merged with GL. Meanwhile, Euclidean distance split the dataset to 3 clusters that contains irrelevant genes. For all variations of single linkage algorithm, the clustering took less than 1 second. The heat-maps in Figure 1 represent the expected clustering versus the single linkage distribution of the genes.

**Table 4.** Confusion matrix for Single Linkage with Euclidean distance.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Expected (True) Cluster | | |  |  |
| Prediction | CC | PD | PS | GL | CH |
|  |  |  |  |  |  |
| CC | 0 | 0 | 1 | 12 | 1 |
| PD | 0 | 0 | 0 | 34 | 0 |
| PS | 0 | 0 | 0 | 24 | 0 |
| GL | 1 | 1 | 0 | 23 | 0 |
| CH | 0 | 0 | 0 | 9 | 0 |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| M.ERYILMAZ | 6 |



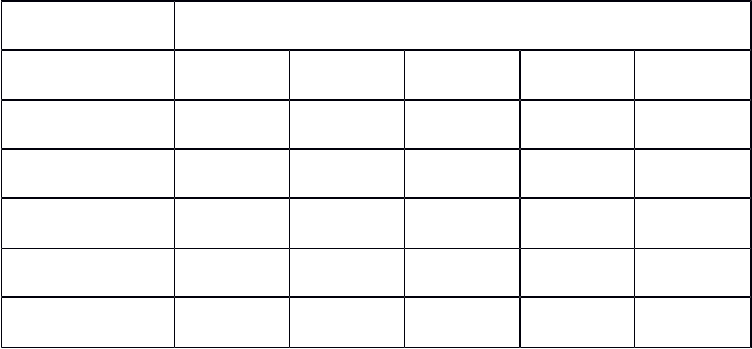
**Figure 1.** Heat-map visualization for single linkage algorithm. Expected clustering (left),single linkage with pre-defined number of clusters (middle); single linkage with undefined number of clusters (right).

**Hierarchical MST**

Next, we apply hierarchical MST clustering that minimizes hierarchically the sum of the distances between a gene and the center of its cluster. By setting the number of clusters N=5, 106/106 genes are categorized correctly with Pearson correlation, which means 100% of success. In the other part, 97/106 genes are clustered correctly with Euclidean distance, which corresponds to 91.5% of success.

As it is seen in the confusion matrices in Table 5 and Table 6, Pearson correlation is perfectly separated all of the genes which belong to different clusters, while Euclidean distance only failed in placing 9 genes with GL annotation.

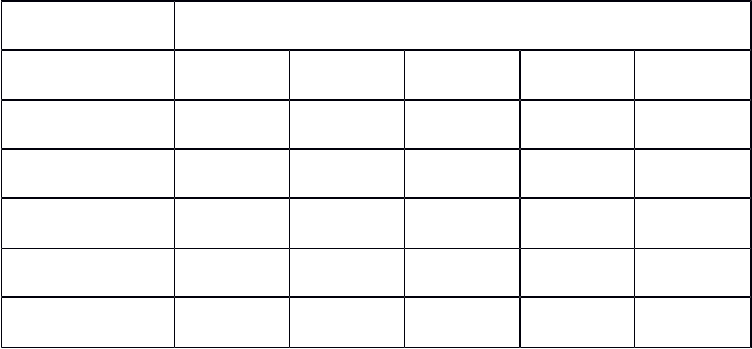
**Table 5.** Confusion matrix for Hierarchical MST with Pearson correlation.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Expected (True) Cluster | | |  |  |
| Prediction | CC | PD | PS | GL | CH |
|  |  |  |  |  |  |
| CC | 14 | 0 | 0 | 0 | 0 |
| PD | 0 | 34 | 0 | 0 | 0 |
| PS | 0 | 0 | 24 | 0 | 0 |
| GL | 0 | 0 | 0 | 25 | 0 |
| CH | 0 | 0 | 0 | 0 | 9 |
|  |  |  |  |  |  |

7 Gene Expression Data Clustering with Minimum Spanning Trees

**Table 6.** Confusion matrix for Hierarchical MST with Euclidean distance.

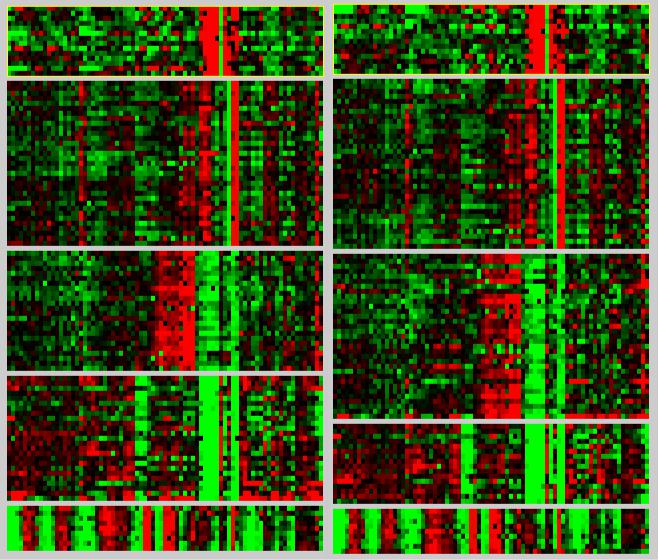


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Expected (True) Cluster | | |  |  |
| Prediction | CC | PD | PS | GL | CH |
|  |  |  |  |  |  |
| CC | 14 | 0 | 0 | 0 | 0 |
| PD | 0 | 34 | 0 | 0 | 0 |
| PS | 0 | 0 | 24 | 0 | 0 |
| GL | 0 | 0 | 9 | 16 | 0 |
| CH | 0 | 0 | 0 | 0 | 9 |
|  |  |  |  |  |  |

On the other hand, when we let the computer to predict the number of clusters, we obtained exactly the same categorization. For all variations of Hierarchical MST algorithm, the clustering took less than 1 second, and total distance on the MST is calculated as 382.3. The heat-maps in Figure 2 represent the expected clustering versus the Hierarchical MST distribution of the genes.

**Global Optimal**

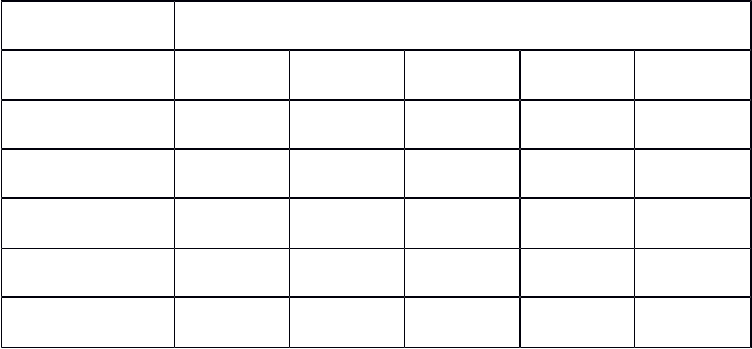
Last, we apply Optimal MST algorithm which minimizes globally the sum of the distances between a gene and the best representative gene from the cluster. In both cases of pre-defined and undefined number of clusters, and for both Pearson and Euclidean metrics, the algorithm correctly clustered 106/106 genes, which corresponds to 100% success.



**Figure 2.** Heat-map visualization for Hierarchical MST algorithm. Expected clustering (left),Hierarchical MST with both pre-defined and undefined number of clusters (right).

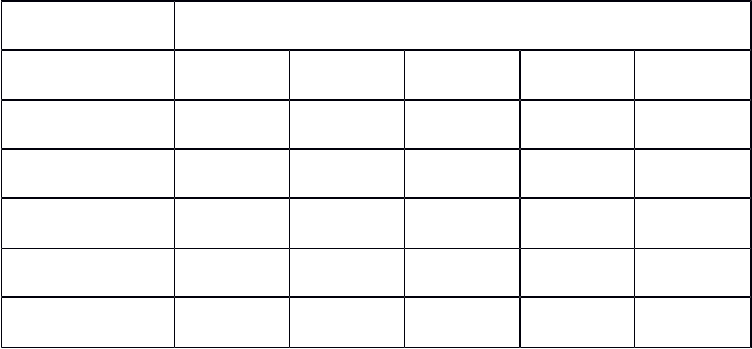
|  |  |
| --- | --- |
| M.ERYILMAZ | 8 |

**Table 7.** Confusion matrix for Global Optimal MST with Pearson correlation.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Expected (True) Cluster | | |  |  |
| Prediction | CC | PD | PS | GL | CH |
|  |  |  |  |  |  |
| CC | 14 | 0 | 0 | 0 | 0 |
| PD | 0 | 34 | 0 | 0 | 0 |
| PS | 0 | 0 | 24 | 0 | 0 |
| GL | 0 | 0 | 0 | 25 | 0 |
| CH | 0 | 0 | 0 | 0 | 9 |
|  |  |  |  |  |  |

**Table 8.** Confusion matrix for Global Optimal MST with Euclidean distance.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Expected (True) Cluster | | |  |  |
| Prediction | CC | PD | PS | GL | CH |
|  |  |  |  |  |  |
| CC | 14 | 0 | 0 | 0 | 0 |
| PD | 0 | 34 | 0 | 0 | 0 |
| PS | 0 | 0 | 24 | 0 | 0 |
| GL | 0 | 0 | 0 | 25 | 0 |
| CH | 0 | 0 | 0 | 0 | 9 |
|  |  |  |  |  |  |

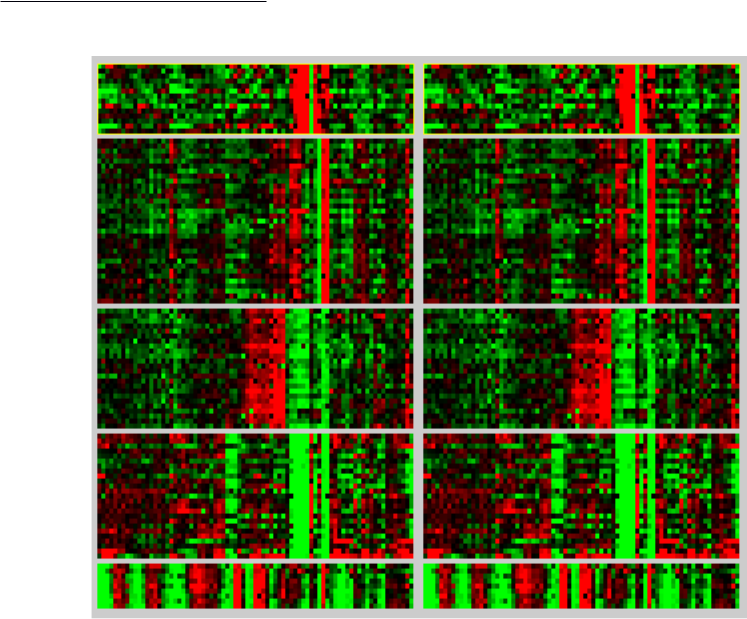
As it is seen in the confusion matrices in Table 7 and Table 8, both similarity metrics are perfectly split all of the genes which belong to different clusters. With pre-defined number of clusters Global Optimal MST algorithm took approximately 4 seconds, while it took about 10 seconds with undefined number of clusters. However, total distance on the MST is calculated as 20.63 which is significantly better than Hierarchical MST. The heat-maps in Figure 3 shows that the expected clustering is exactly the same with Global Optimal MST.

**CONCLUSIONS**

Although all of the three algorithms adopts similar MST-based clustering approaches, the variation in edge-cutting strategy causes a significant difference in their performance on clustering gene expression data. Additionally, each algorithm generates different values of total distance for the constructed MST. As it is expected by definition, single-linkage and hierarchical algorithms are much faster than global optimal algorithm while global optimal algorithm is perfectly accurate in the most cases. As it is suggested in the previous studies, Pearson correlation performs better as a gene similarity metric in comparison to Euclidean distance.

Based on our study, hierarchical MST is a preferable algorithm with its very fast execution (1 sec) and perfect accuracy (100%) on clustering with pre-defined number of clusters. When the number of clusters is undefined, although global optimal MST provides 10% boost in accuracy, it is 10 times slower than Hierarchical algorithm. In any conditions, with 45% of maximum accuracy, single linkage algorithm seems to be inapplicable on microarray gene expression data clustering. As a conclusion, in spite of the fact that MST-based clustering is promising for gene expression analysis; its performance is highly dependent on data and edge-cutting strategy.

9 Gene Expression Data Clustering with Minimum Spanning Trees



**Figure 3.** Heat-map visualization for Global Optimal MST algorithm. Expected clustering(left), Global Optimal MST with both pre-defined and undefined number of clusters (right).

**REFERENCES**

Ben-Dor, A., Shamir, R., Yahini, Z. (1999). “Clustering Gene Expression Patterns”. *Journal of Computational* *Biology*,**6**(3-4), 281–297.

D’haeseleer, P. (2005). “How does gene expression clustering work?”. *Nature Biotechnology*, **23**(12), 1499-1501

Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D. (1998). “Cluster analysis and display of genome-wide expression patterns”. *Proc. Natl. Acad. Sci. USA*, **95**, 14863–14868.

Gower, J.C., Ross, G.J.S. (1969). “Minimum spanning trees and single linkage cluster analysis”. *Applied* *Statistics*,

**18**, 54–64.

Kruskal, J. (1956). “On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem”.

*Proceedings of the American Mathematical Society*,**7**(1), 48-50.

Xu, D., Olman, V., Wang, L., Xu, T. (2003). “EXCAVATOR: a computer program for efficiently mining gene expression data”, *Nucleic Acids Research*, **32**(19), 5582-5589.

Xu, Y., Olman, V., Xu, D. (2001). “Minimum spanning trees for gene expression data clustering”. *Genome* *Informatics*,**12**, 24–33.