AIMS ESSAY 2006

Overview of Tools for Microarray Data Analysis and Comparison Analysis

By Chodziwadziwa Whiteson Kabudula

Supervisors:

Bajic, Vladimir B. (Prof) and Hofmann, Oliver (Dr)

(South African National Bioinformatics Institute (SANBI),

University of the Western Cape)

May 25, 2006

Dedication

To my family (Uncle, Aunt, Mum, Sister and Cousins) for their patience, understanding, support, encouragement and most of all love.

Contents

List of Figures

Abstract

Progress in microarray gene expression technology has been complemented by advances in techniques and tools for microarray data analysis. There exist various types of analyses of microarray data and a variety of public tools are available for performing these analyses. Here, we present an overview of three publicly-accessible web-based tools for microarray data analysis; Gene Expression Pattern Analysis Suite (GEPAS), Expression Profiler: Next Generation (EP:NG), and Microarray Data Analysis Web Tool (MIDAW). The discussion particularly focuses on one of the most widely used microarray data analysis techniques known as clustering. Insights are provided on the properties and usefulness of each of the three tools with regard to clustering. For each of the tools, a thorough exploration of the possibilities provided for various clustering techniques is made. In addition, we present a comparison analysis of the performance of the three tools with emphasis on clustering.

Introduction

Microarray gene expression technology has emerged as a fundamental tool in biomedical research because of its ability to allow the study of gene expression profiles of thousands of genes simultaneously. Uses of this technology include studying disease patterns, gene expressions, gene regulations and interactions; and identifying potential therapeutic drug targets and diagnostic markers for diseases. The information that is generated through microarrays in turn serves to answer most of the questions currently asked by researchers.

The growth in usage of microarrays has also resulted in advancement of the physical microarray chip technology and a tremendous increase in the number of techniques and tools available for analysing microarray data. There are different variations on the microarray technology, and there also exist various types of analyses of microarray data, as well as a variety of data analysis tools [1].

Presented here is an overview of three publicly-accessible web-based tools for microarray data analysis; namely, Gene Expression Pattern Analysis Suite (GEPAS), Expression Profiler: Next Generation (EP:NG), and Microarray Data Analysis Web Tool (MIDAW). The focus is on the possibilities that each of these tools provide for clustering, which is one of the most widely used techniques for microarray data analysis.

The first chapter discusses microarray technology and the biology behind it. For this, a brief description is presented on flow of genetic information, the steps involved in microarray experiments and applications of microarrays.

The second chapter looks at techniques and tools for microarray data analysis. The data analysis technique presented is clustering. A number of existing statistical and computational approaches for clustering, which include agglomerative hierarchical clustering, k-means, k-mediods, self-organising maps, and self-organising tree algorithm as well as measures of (dis)similarity are discussed. Thereafter, we thoroughly explore the options which each of the three tools provide for

the various clustering techniques.

The third chapter deals with a comparison analysis of the performance of these three tools. The performance of the tools is assessed by analysing a dataset of 200 gene expression values measured on 28 samples, using the same measures of (dis)similarity and clustering techniques across all the tools.

Finally, the last chapter presents discussions and conclusion.

Chapter 1

Microarray Technology and the Biology Behind It

Microarray gene expression technology is a widely used technique in biomedical research to elucidate the relations between, the expression of, and the function of genes [2]. This chapter is going to describe microarray technology and its applications. To set the scene, a brief description of the biological background is first presented.

1.1 Biological Background

The basis for the development of microarrays is the genomic sequence of information [3]. The *genome*, which may be defined as the genetic information of an individual, is encoded in a double-stranded helical structure of molecules called *deoxyribonucleic acid* (**DNA**). The **DNA** molecules are packaged into structures called chromosomes found in the *nucleus* of the *cell* of *Eukaryotes* (see Figure 1.1), which is the fundamental unit of life.

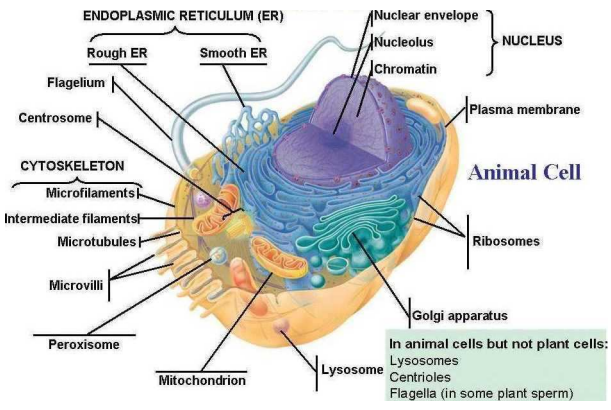


Figure 1.1:

Eukaryotic Cell, Picture from [4]

Each strand of the **DNA** is made up of a sugar-phosphate backbone and complex chemical

compounds called nucleotides, which are distinguished by four bases, *Adenine* (**A**), *Cytosine* (**C**), *Guanine* (**G**) and *Thymine* (**T**). These bases contain the elements *carbon*, *nitrogen*, *oxygen* and *hydrogen*. The two strands are held together by hydrogen bonds between the bases in a complementary form; **A** in one strand pairs with **T** in the other strand, and **C** pairs with **G** (see Figure 1.2 for the basic structure of the DNA). This is referred to as *the fundamental "base pair rule"of* **DNA** [6]. This base pair rule makes it possible to produce the reverse complementary strand based on knowledge of the sequence of bases on one strand.

The **DNA** molecule may be described by a linear sequence of the bases such as **GCATCAATGCGTCCGATGCATTACGGCGG...**. Substrings of the complete **DNA** sequence are called *genes*. Genes may be defined as instructions that code for information necessary to construct the chemicals (*proteins*, polypeptide chains of twenty different *amino acids* etc.) needed for an organism to function [2].

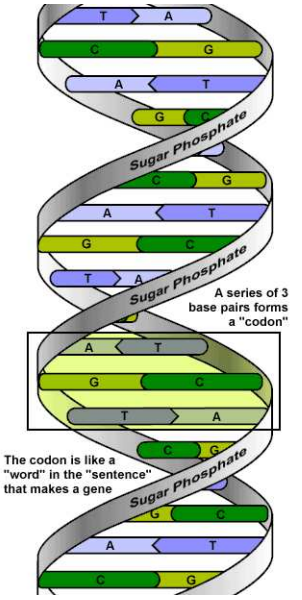


Figure 1.2: Basic Structure of DNA, Picture from [5]

The way genetic information flows from the **DNA** to proteins is that, first, the information in the **DNA** is copied into a *ribonucleic acid* (**RNA**) molecule through the process called transcription. Using the enzyme *RNA polymerase*, the bases in the **DNA** molecule are respectively transcribed into **RNA**. **RNA** is a nucleic acid that is very similar to **DNA**. However, it is less stable than **DNA** and is almost exclusively found in single-stranded form. It is also made up of four bases with a slight difference that the base *Thymine* (**T**) is replaced by *Uracil* (**U**).

In *Eukaryotes*, genes in a **DNA** sequence consist of coding regions (*exons*) and non-coding regions (*introns*). So, after **RNA** has been transcribed, it goes through a series of *post-transcription* modifications where the introns are removed and the remaining exons are joined. The final product is a molecule called *messenger RNA* (mRNA). Lastly, mRNA ferry the genetic information out of the *nucleus* to the *cytoplasm* where *ribosome*, using the *genetic code*, converts it to *amino acids* which form proteins. The *genetic code* is a triplet base code where successive *codons* (three adjacent letters in the mRNA) encode one of the twenty amino acids or the signal to stop *translation* (the process by which the mRNA bases are used to make amino acids). The whole process from transcription to translation is what is referred to as *gene expression* [3]. Figure 1.3 schematically depicts the flow of genetic information. A gene is said to be expressed if mRNA is transcribed from

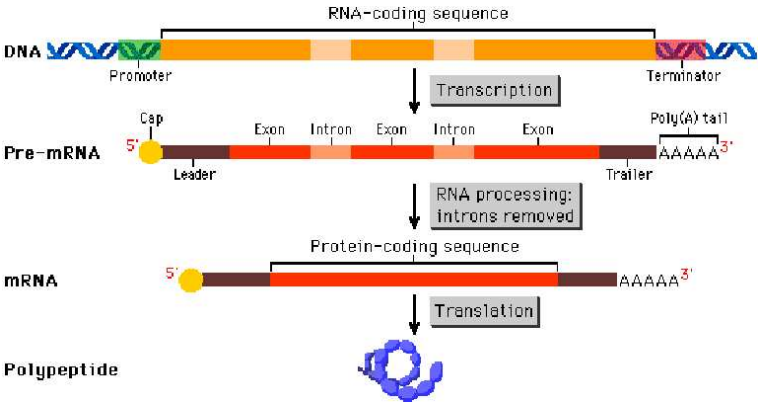


Figure 1.3:

Schematic Flow of Genetic Information, Picture from [7]

the gene's **DNA** sequence and is used as a template to guide the synthesis of a protein. However, sometimes the term *gene expression* is used only for the transcription part of this process. Even though all cells in the body possess the same **DNA**, *gene expression* is controlled by developmental stage, tissue, age and environmental conditions [2].

1.2 Microarray Technology

Microarray technology is a high-throughput technique that enables biologists to measure how much mRNA corresponding to a particular gene is present in the cell(s) or tissue of interest [3]. It allows the measure of the expression of thousands of genes simultaneously [1, 2, 3, 8, 9, 10]. In general, a microarray consists of a solid support such as glass, plastic or silicon on which thousands of **DNA** sequences, known as *probes* (corresponding to segments of genes) are arranged and fixed in a regular pattern [3, 12].

There are two major categories of microarray platforms that are widely used for measuring gene expression. One is that of complementary DNA (cDNA) microarrays, and the other is oligonucleotide microarrays [1, 3, 9, 10]. The difference between them is in their experimental protocols, lengths of probes and number of samples measured per array. Despite these variations, in both cases, the experiment begins with the collection of samples of a particular tissue or cell of interest and extraction of mRNA from the samples [3, 10]. Usually two (test and reference) groups of samples are collected. Next, complementary DNA (cDNA) is obtained from the mRNA by reverse transcriptase - polymerase chain reaction (RT-PCR). The cDNA samples obtained are called *targets*. The target samples are then labeled with fluorescent dyes. Thereafter, the labeled target samples are *hybridized*

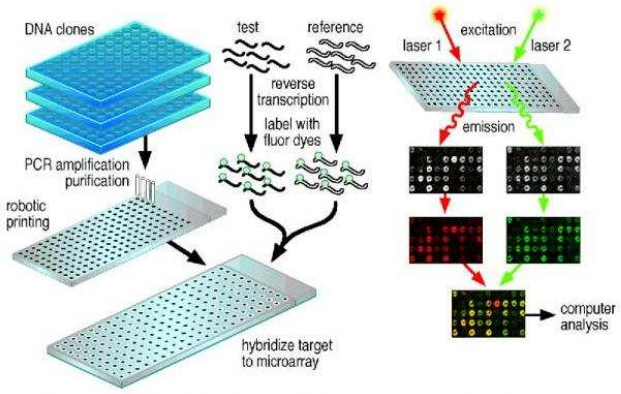


Figure 1.4: Steps Involved in a Microarray Experiment, Picture from [11]

onto the microarray. The reasoning behind this is that, under proper conditions, complementary sequences will bind to each other while non-complementary sequences will not bind. For example, if the **DNA** sequence on the array is fifteen nucleotides long, **GCATCAATGCGTCCA**, the sequence **CGTAGTTACGCAGGT** in the target sample will "hybridize" to that probe. Once the cDNA targets have been hybridized to the array, the array is then washed to remove any loose targets. Finally, the array is scanned by a laser scanner to determine the amount of the target that is bound to each spot. The resulting signal intensity, which correlates with the amount of captured probe, is measured, stored in a computer and later analyzed (see Figure 1.4 for a summary of the steps involved in a microarray experiment). The major assumption is that the amount of mRNA corresponding to a particular gene is positively correlated with the expression level of that particular gene.

1.3 Applications of Microarrays

Microarrays are used for a number of purposes. Peeters and Van der Spek (2005) [1] presented an overview of a number of fields which are currently applying microarray technology. These include such fields as:

- *Pharmacogenomics*, where microarrays are utilised for the purposes of drug discovery, drug target validation and prediction of undesirable side effects.
- *Forensics*, where microarrays serve the purpose of individual identification.

- *Epidemiological Research*, where gene expression profiles are used to monitor infectious outbreaks and determine genotypic varations that underlie disease outbreaks.

- *Cardiovascular Research*, where microarrays are used for chromosomal mapping and identification of genes involved in the primary etiology of cardiac diseases as well as identification of significant risk factors for the development and advancement of such diseases.

- *Oncology and Disease Classification Research*, where microarrays are used to study diagnostics and progression of tumours and variations in treatment responses as well as identifying particular pathological subgroups of a disease.

Chapter 2

Techniques and Tools for Microarray Data Analysis

The analysis of data generated through microarray experiments is done in several steps. After conducting all the biological and hybridization experiments, the very first step towards data analysis is the extraction of raw intensity data from the images obtained through scanning the slides. This is done in a number of steps which include (i) scanning, (ii) spot recognition, (iii) segmentation, (iv) intensity extraction and ratio calculation. Once, that is done, the data is then preprocessed to get rid of poor quality spots and normalised to remove as many systematic errors as possible [9, 13]. Thereafter, statistical and data mining techniques are applied to extract meaningful information concerning the relations between, the function of, and the expression of the genes [13, 14]. There is a variety of publicly available tools that have been developed to perform various analyses to yield information that can answer a number of biological questions. This chapter is going to provide an overview of three such tools, namely, Gene Expression Pattern Analysis Suite (GEPAS) [15, 16, 17], Expression Profiler: Next Generation (EP:NG) [18], and Microarray Data Analysis Web Tool (MIDAW) [19]. The discussion is particularly focused on one of the widely used microarray data analysis techniques known as clustering. The first section of the chapter is devoted to a description of clustering as a technique for analysing microarray data. The last section gives an account of each of the mentioned tools. Specifically, it explores the possibilities that each of the tools provides for clustering.

2.1 Clustering

Clustering belongs to a group of data analysis techniques known as Exploratory Data Analysis (EDA) or unsupervised learning, which are techniques that aim at detecting structures in the data without incorporation of any prior knowledge (e.g. gene or sample annotations) in the process [13, 14, 20, 21]. Clustering is a technique that groups together genes or samples that are similar to one another in a subset or "cluster" [22, 23]. Genes or samples assigned to a particular cluster are more similar to one another than those not assigned to that cluster. In microarrays, clustering is extensively used as a tool for:

a) Dimension reduction: Since microarrays generate datasets with thousands of gene expression values, the data must be reduced for meaningful exploration of the relationships between genes (or samples). So, clustering techniques help to reduce the size of the dataset by gathering genes (or samples) into a small number of groups where observations in a group can be represented by an *average* of the observations [20, 24].

b) Hypothesis generation: The reasoning behind clustering techniques is that genes in the same cluster may be functionally related and co-regulated. Hence, clustering may suggest possible roles for genes with unknown functions, which can then be formally validated through additional experiments, based on the known functions of some other genes that they share the same cluster with [2].

c) Hypothesis testing: Clustering techniques are also used in an attempt to determine if patterns defined by other procedures are indeed manifested in the dataset [8].

Clustering is mainly characterised by two fundamental steps; computation of similarities among the genes (or samples) to be clustered, and the use of a clustering technique (method) to create groups of relatively homogeneous genes (or samples).

2.1.1 Similarity Measures

Similarity measures are mathematical calculations of distances between gene expression vectors [21, 26, 27]. The commonly used similarity measures are the *Euclidean Distance, Euclidean Distance Squared, Manhattan or City Block Distance, Linear Correlation Distance, Uncentred Linear Correlation Distance, Spearman's Rank Correlation Distance, and Chord Distance* [10, 13, 14, 21, 28].

Considering two gene expression vectors, $\mathbf{x}$ and $\mathbf{y}$, involving measurements on p samples,

- *Euclidean Distance* is the sum of the squared distances between the two vectors and is given by:

$$d_{Ec}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2} \tag{2.1}$$

- *Manhattan or City Block Distance* is the sum of linear distances between the two vectors and is given by:

$$d_m(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{p}|x_i - y_i| \tag{2.2}$$

Both Euclidean and Manhattan distances are useful when one wants to look at absolute values and they reveal genes that have similar expression levels in clusters.

- *Linear Correlation Distance* is a measure of association between the two vectors and is given by:

$$d_r(\mathbf{x}, \mathbf{y}) = 1 - r(\mathbf{x}, \mathbf{y}) \quad \text{or} \quad 1 - |r(\mathbf{x}, \mathbf{y})| \tag{2.3}$$

where,

$$r(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{p}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{p}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{p}(y_i - \bar{y})^2}} \tag{2.4}$$

is the Pearson correlation coefficient and $\bar{x}$ and $\bar{y}$ are the means of vectors $\mathbf{x}$ and $\mathbf{y}$ respectively. The Linear Correlation Distance measure of similarity is useful when one wants to look at the shapes of the gene expression patterns regardless of the expression levels.

- *Uncentred Linear Correlation Distance* has the same formula as *Linear Correlation Distance* except that the means of the expression vectors are offset to zero. Thus, $r(\mathbf{x}, \mathbf{y})$ is replaced by the uncentred Pearson correlation coefficient

$$r_{uc}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{p} x_i y_i}{\sqrt{\sum_{i=1}^{p} x_i^2}\sqrt{\sum_{i=1}^{p} y_i^2}} \tag{2.5}$$

- *Spearman's Rank Correlation Distance* also has the same formula as *Linear Correlation Distance* except that $r(\mathbf{x}, \mathbf{y})$ is replaced by the Spearman's rank correlation coefficient, which is a non-parametric measure of association between the two vectors,

$$r_s(\mathbf{x}, \mathbf{y}) = 1 - \frac{6\sum_{i=1}^{p} d_i^2}{p(p^2 - 1)} \tag{2.6}$$

where d_i =rank of x_i−rank of y_i.

This measure of similarity is used when one wants to look at shapes in a non-parametric way.

- *The Chord Distance* is given by:

$$Chord(\mathbf{x}, \mathbf{y}) = \sqrt{2 - \frac{2\sum_{i=1}^{p} x_i y_i}{\sqrt{\sum_{i=1}^{p} x_i^2 \sum_{i=1}^{p} y_i^2}}} \tag{2.7}$$

The similarity measures are classified as either metric distances or semi-metric distances [21, 25]. They fall under metric distances if they satisfy the following axioms:

- *Indistinguishability of identicals*. A gene expression vector, $\mathbf{x}$, is at zero distance from itself, $d(\mathbf{x}, \mathbf{x}) = 0$.
- *Positive definite*. Given two gene expression vectors, $\mathbf{x}$ and $\mathbf{y}$, the distance, d, between them satisfies, $d(\mathbf{x}, \mathbf{y}) \geq 0$.
- *Symmetric*. Given two gene expression vectors, $\mathbf{x}$ and $\mathbf{y}$, the distance, d, between them satisfies, $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$.
- *Triangular rule*. Given three gene expression vectors, $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$, the distance, d, between them satisfies, $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{y}, \mathbf{z})$.

Examples of metric distances are Euclidean and Manhattan distances.

On the other hand, semi-metric distances satisfy the first three axioms of metric distances, but fail to obey the triangular rule. Correlation coefficient distances belong to this category of distance measures.

2.1.2 Clustering Methods

Once a measure of similarity has been chosen, clustering can be performed using two analytic techniques, *hierarchical clustering* and *non-hierarchical clustering* [9, 21, 27]. Here we are going to describe some of the commonly used clustering methods in microarray data analysis, namely, agglomerative hierarchical clustering, k-means clustering, k-mediods clustering, self-organising maps clustering, and self-organising tree algorithm clustering.

Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering is a bottom-up technique that iteratively builds clusters of genes with similar expression patterns [10]. The result is a nested sequence of partitions that is represented with a tree diagram called a *dendrogram*, which is a graphical display of the hierarchical

structure implied by the similarity (distance) matrix and clustered by a linkage rule (described below) [27].

The approach begins with calculation of a pair-wise distance or similarity matrix, as illustrated below

$$D = \begin{bmatrix} d(1,1) & d(1,2) & \dots & d(1,n) \\ d(2,1) & d(2,2) & \dots & d(2,n) \\ \vdots & & & \vdots \\ d(n,1) & d(n,2) & \dots & d(n,n) \end{bmatrix} \tag{2.8}$$

where $d(i,j)$ denotes the distance between gene i and gene j calculated from the dataset containing n gene expression vectors using one of the similarity measures described above. Initially, each gene is considered to belong to a different cluster. Thereafter, the pair of genes with the smallest distance is identified and the two closest genes forming that pair are merged into one cluster. The method proceeds by defining a *linkage rule*; a rule for calculating the distance between the new cluster and remaining clusters. Using the appropriate chosen linkage rule, the distances between the new cluster and all the other clusters are recalculated and the distance matrix is updated accordingly. This step is repeated until all the genes are merged into a single cluster [9, 21].

There are several possible linkage rules, each producing a distinct agglomerative hierarchical clustering method. Some of the most widely used agglomerative hierarchical clustering methods in microarrays are Single-linkage clustering, Complete-linkage clustering, and Average-linkage clustering [21]. According to [25], Lance and Williams(1967), developed a general formula that describes a linkage rule for any agglomerative hierarchical clustering method. The formula is,

$$d((\mathbf{j},\mathbf{k}),\mathbf{l}) = A(\mathbf{j})d(\mathbf{j},\mathbf{l}) + A(\mathbf{k})d(\mathbf{k},\mathbf{l}) + Bd(\mathbf{j},\mathbf{k}) + C|d(\mathbf{j},\mathbf{l}) - d(\mathbf{k},\mathbf{l})| \tag{2.9}$$

where $d((\mathbf{j},\mathbf{k}),\mathbf{l})$ is the distance between the fused cluster $(\mathbf{j},\mathbf{k})$ and further candidates $\mathbf{l}$ to be merged to it, $d(\mathbf{j},\mathbf{l})$ is the distance between clusters $\mathbf{j}$ and $\mathbf{l}$, $d(\mathbf{k},\mathbf{l})$ is the distance between clusters $\mathbf{k}$ and $\mathbf{l}$, $d(\mathbf{j},\mathbf{k})$ is the distance between clusters $\mathbf{j}$ and $\mathbf{k}$, and the capital letters, $A(\mathbf{j})$, $A(\mathbf{k})$, B, and C refer to parameters that further define the linkage form. Below are details about the various agglomerative hierarchical clustering methods and the values of the parameters in each method as implemented in the GEPAS tool.

a) Single-linkage clustering: This is also known as *nearest neighbour clustering*. The distance between any cluster, **A**, to the new cluster, **B**, is the minimum of all distances between members of cluster **A** and members of cluster **B**. The values of the parameters are: $A(\mathbf{j}) = A(\mathbf{k}) = \frac{1}{2}$, $B = 0$, and $C = -\frac{1}{2}$.

b) Complete-linkage clustering: This is also known as *furthest neighbour clustering.* The distance between any cluster, **A**, and the new cluster, **B**, is calculated as the maximum of all the distances between objects in cluster **A** and those in cluster **B**. Here, the parameters take the following values: $A(\mathbf{j}) = A(\mathbf{k}) = C = \frac{1}{2}$, and $B = 0$.

c) Average-linkage Clustering: The distance of any cluster, **A**, to the new cluster, **B**, is the average of the distances between items in cluster **A** and items in cluster **B** . There are several algorithms for average-linkage clustering. Only two, which are most common are described here. These are arithmetic average-linkage clustering and centroid clustering.

i. *Arithmetic Average-linkage clustering:* These algorithms calculate the average similarity of a new cluster to an already existing cluster. There are two of these algorithms:

- Unweighted Pair-Group Method Average (**UPGMA**), which computes distance between two clusters as the average distance between all pairs of items in the two clusters. The values of the parameters are: $A(\mathbf{j}) = \frac{N_{\mathbf{j}}}{N_{(\mathbf{j},\mathbf{k})}}$, $A(\mathbf{k}) = \frac{N_{\mathbf{k}}}{N_{(\mathbf{j},\mathbf{k})}}$, $B = C = 0$, where $N_{\mathbf{j}}$ is the size of cluster $\mathbf{j}$, $N_{\mathbf{k}}$ is the size of cluster $\mathbf{k}$, and $N_{(\mathbf{j},\mathbf{k})}$ is the size of the fused cluster $(\mathbf{j}, \mathbf{k})$.
- Weighted Pair-Group Method Average (**WPGMA**), which is almost the same as **UPGMA** except that in the calculations, the sizes of the respective clusters are used as weights. For this method, the values of the parameters are $A(\mathbf{j}) = \frac{N_{\mathbf{j}}}{N_{(\mathbf{j},\mathbf{k})}}$, $A(\mathbf{k}) = \frac{N_{\mathbf{k}}}{N_{(\mathbf{j},\mathbf{k})}}$, $B = -\frac{N_{\mathbf{j}} N_{\mathbf{k}}}{{N_{(\mathbf{j},\mathbf{k})}}^2}$, and $C = 0$.

ii. *Centroid clustering:* These algorithms calculate the centroid (centre of mass) of the objects that merge to form clusters. There are also two of these algorithms:

- Unweighted Pair-Group Method Centroid (**UPGMC**), which calculates the distance between two clusters as the distance between their centroids. The parameter values are $A(\mathbf{j}) = A(\mathbf{k}) = \frac{1}{2}$, and $B = C = 0$.
- Weighted Pair-Group Method Centroid (**WPGMC**), which is identical to **UPGMC**, except that it introduces weighting in the computations. The parameter values are: $A(\mathbf{j}) = A(\mathbf{k}) = \frac{1}{2}$, $B = -\frac{1}{4}$, and $C = 0$.

In the agglomerative hierarchical clustering technique, the final number of clusters is determined by the level at which the *dendrogram* is cut [20]. For an illustration on how agglomerative hierarchical clustering technique works see Example 1 in Appendix A.

K-means Clustering

k-means clustering is a non-hierarchical clustering technique, which simply partition the genes into a pre-determined number of clusters [9, 21]. The method starts with the selection of a pre-specified number, k, of clusters and initial gene expression vectors from the dataset that serve as the starting positions of the centres of the chosen k clusters. Thereafter, distances are calculated from each gene expression vector in the dataset to each of the k centres. Genes are then assigned to the cluster whose centre is closest to them. Next, each cluster centre is replaced by the average of the expression vectors of the genes belonging to it. The procedure is repeated by recalculating the distance from each gene expression vector in the dataset to the updated cluster centres and reassigning genes to the closest cluster until no genes are reassigned [20, 21]. The goal of the k-means algorithm is to partition the set of gene expression vectors, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ represented by the whole dataset, into k disjoint subsets S_i with N_i elements, i.e. $\mathbf{X} = \{S_1, S_2, \ldots, S_k\}$, where $S_i = \{\mathbf{x}_{1S_i}, \mathbf{x}_{2S_i}, \ldots, \mathbf{x}_{N_iS_i}\}$, in a way that minimises the sum of squared distances from each gene expression vector to the centre of its set, μ_{S_i}. Formally, it seeks to minimise:

$$I = \sum_{i=1}^{k} \sum_{j=1}^{N_i} ||\mathbf{x}_{jS_i} - \mu_{S_i}||^2, \tag{2.10}$$

where μ_{S_i} is the mean gene expression vector of the genes in the set S_i and is given by:

$$\mu_{S_i} = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{x}_{jS_i}. \tag{2.11}$$

There is no *dendrogram* produced; however, hierarchical clustering techniques can be used on each data partition after they are constructed. For an illustration on how k-means clustering technique works see Example 2 in Appendix B.

k-mediods Clustering

k-mediods is another non-hierarchical clustering technique that is similar to the k-means technique except that, in the iterations, centres for each cluster are restricted to be one of the gene vectors belonging to the cluster [23]. Thus, $\mu_{S_i} = \mathbf{x}_{jS_i}$, for some j. The k-mediods algorithm, therefore, consists of iterating the following steps until there is no more change in cluster assignments.

- For each gene expression vector, $\mathbf{x}$, the closest cluster centre, $\mu_{S_i} = \mathbf{x}_{jS_i}$, is identified, and the gene vector is assigned to that cluster.

- For the i^{th} cluster, $(i = 1, 2, \ldots, k)$, the gene expression vector (from cluster i) that minimises the sum of dissimilarities to the other gene expression vectors is identified and becomes the new centre of the cluster.

Self-Organising Maps Clustering

Self-Organising Maps (SOM) is a neural-network-based non-hierarchical clustering technique developed by Professor Kohonen at the University of Helsinki [21, 10, 22, 28, 29]. First, the user defines a topology (*geometric configuration*) among the clusters. Usually, this is a two-dimensional hexagonal or rectangular grid of neurons also called nodes. The difference between the two topologies lies in how the neurons of the map are connected to adjacent neurons by a neighbourhood relation which determines the structure of the map. In the hexagonal grid, neurons have six nearest neighbours while in a rectangular one they only have four (see Figure 2.1). The objective of the SOM technique is to define which clusters should be neighbours and keep the information throughout the clustering process so that genes that are more similar across clusters can be assigned to neighbouring clusters.

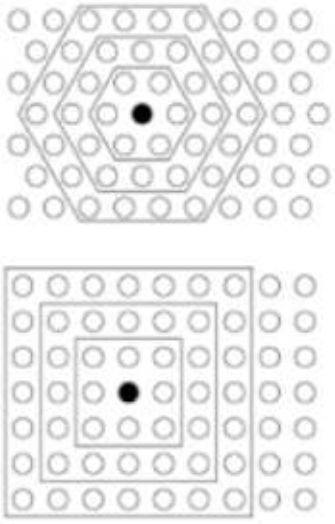


Figure 2.1:
SOM Grids in the
2-dimensional case.
Picture from [28]

Suppose that p-dimensional gene expression vectors $\mathbf{x_i} = [x_{i1}, \ldots, x_{ip}]$, forming an $n \times p$ gene expression data matrix $\mathbf{X}$, are to be clustered; then, each node, k, of the SOM is associated with a p-dimensional weight vector (cluster centre) $\mathbf{c_k} = [c_{k1}, \ldots, c_{kp}]$. The number of nodes is equal to the number of expected clusters. In the case of a 2-dimensional topology, the total number of nodes on the map is equal to $q_1 \times q_2$, where q_1 is the number of nodes (neurons) in the horizontal dimension and q_2 is the number of nodes (neurons) in the vertical dimension. The SOM clustering technique proceeds from "training" the network to the actual clustering of the genes. Before training starts, the weight vectors are initialised using one of the following two procedures:

- Random initialisation, where small random values are used to initialise the weight vectors, with the only restriction that all the weight vectors should be different.
- Sample initialisation, where the weight vectors are initialised with sample gene expression

vectors drawn randomly from the dataset.

Training is done using an iterative process that continues until convergence (or some other termination condition). At every training step, one gene expression vector, $\mathbf{x_i}$, chosen in some order (possibly random) from the input dataset is compared with all the weight vectors (nodes) of the network by means of a similarity measure, typically the Euclidean distance, to identify the node that is most similar to it. This node is called the winning node or Best-Matching Unit (BMU). After finding the winning node, l, whose weight vector, $\mathbf{c_l}$, has the greatest similarity with the input gene expression vector, $\mathbf{x_i}$, the weight vectors of the winning node and its neighbourhood are updated (changed) to make them closer to the input gene expression vector. A trial or cycle is said to have been completed when this process has been performed for all the gene expression vectors in the input dataset. The training ends when a predefined number of trials has been carried out.

At each trial time-step, t, the weight vectors of the nodes of the network are updated using the following rule:

$$\mathbf{c_k}(t+1) = \mathbf{c_k}(t) + h_{j,l}(t)(\mathbf{x_i}(t) - \mathbf{c_k}(t)) \tag{2.12}$$

where $h_{j,l}(t)$, is the neighbourhood kernel around the winner node, l, during trial t. The neighbourhood kernel is a non-increasing function of time and distance from the location of node j to that of the winner node, l, on the map grid. It consists of a neighbourhood function $D(t)(j,l)$ and a learning rate function $\alpha(t)$. Thus,

$$h_{j,l}(t) = \alpha(t)D(t)(j,l) \tag{2.13}$$

In the simplest form, the neighbourhood function is defined as a bubble, constant over the whole neighbourhood of the winner node and zero elsewhere. Alternatively, it may take the form of a Gaussian neighbourhood function:

$$D(t)(j,l) = exp\left(-\left[\frac{d(j,l)}{\sigma(t)}\right]^2\right), \tag{2.14}$$

where $d(j,l)$ denotes the distance of the location of node j from that of the winning node, l, on the map grid, and $\sigma(t)$ is a monotonically decreasing positive function of t that defines the width of the kernel (i.e radius of the neighbourhood), and decreases linearly from a starting value, r, to 1 during the training process. The learning rate $\alpha(t)$ is also a decreasing function of t which is either a linear function or a function inversely proportional to t: $\alpha(t) = \frac{A}{t+B}$, where A and B are suitably selected constants. Typically, $\alpha(t)$ linearly decreases from 0.9 to zero during training.

The training is usually done in two phases. In the initial phase, which is also called the ordering phase, relatively large values of $\alpha(t)$ and $\sigma(t)$ are used to achieve global order; while in the second,

which is the fine-tuning phase, small values are used. Typically, the neighbourhood radius for the ordering phase starts from r equal to the diameter of the map (half map dimension) and decreases to 3 while a typical starting value of $r = 3$ is used for the fine-tuning phase. On the other hand, the learning rate $\alpha(t)$, is usually decreased form 0.9 to 0.1 during the ordering phase, and from 0.1 to zero in the fine-tuning phase. A predefined training length (number of steps) is always set for each training phase and this number is usually greater in the fine-tuning phase.

Finally, after the network has been properly trained, the genes are then mapped to the relevant clusters; this depends on which weight vector they are most similar to.

Self-Organising Tree Algorithm Clustering

Self-Organising Tree Algorithm (SOTA) is a divisive (top-down) hierarchical neural network clustering technique based on both the SOM and growing cell structures. The algorithm was developed by Herrero, Valencia and Dopazo [30]. It implements a binary tree topology, instead of the classical rectangular or hexagonal one, and a different strategy of training.

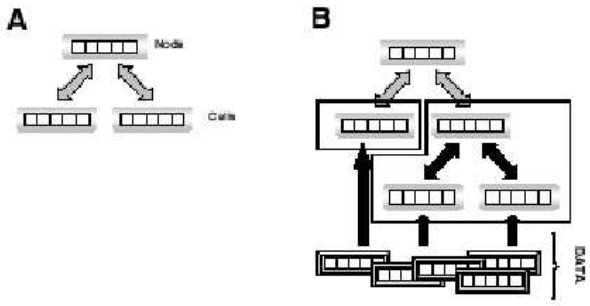


Figure 2.2: Topology of the SOTA Network and the Growing Algorithm

In SOTA, a series of nodes, arranged in a binary tree, are adapted to the intrinsic characteristics of the input dataset. The technique proceeds in the following manner. Initially, the method starts with two external vectors called cells connected by an internal vector called a node (see Figure 2.2 A). The vectors are of the same size as the input gene expression vectors. At the very beginning, the two cells and the node are initialised with either the mean values of the corresponding columns of the input dataset or with random values. Next, the gene expression profile vectors in the dataset are presented to the network and compared with the cells. Then, the output topology is expanded by generating two new descendant cells from the cell having the most heterogeneous population of associated input gene expression profile vectors (see Figure 2.2 B). The cell from which the two new cells are generated changes its status and becomes a node.

There are a series of operations that are performed before a cell changes its status to become a node. The whole process is called a cycle. During a cycle, cells and nodes are repeatedly adapted

by the input gene expression profile vectors. The adaptation process in each cycle is performed during a series of epochs. In each epoch, all the gene expression profile vectors are presented to the network. Each gene expression profile vector is compared to the cells and the best matching cell, also referred to as the winning cell (the cell with the smallest distance from the input gene expression vector), is identified and the gene expression vector is assigned to that cell. Thereafter, that winning cell and its neighbourhood are updated using the following formula

$$\mathbf{c}_i(\tau + 1) = \mathbf{c}_i(\tau) + \eta(\mathbf{x}_j - \mathbf{c}_i(\tau)) \tag{2.15}$$

where η is a factor that accounts for the magnitude of the updating on a cell depending on its distance from the winning cell, $\mathbf{c}_i(\tau)$ is the i^{th} cell vector at the presentation τ, and $\mathbf{x}_j$ is the j^{th} input gene expression profile vector. If the sister cell of the best matching cell has no descendants, the neighbourhood includes the winning cell, the parent node, and the sister cell, otherwise it is formed by the winning cell itself (see Figure 2.2 B). Different decreasing values, $\eta_w = 0.01$, $\eta_p = 0.005$, and $\eta_s = 0.001$ are typically used for the winning cell, the parent node, and the sister cell respectively. In the case where both sister cells are equal (during the initial stage of the network and just after cell duplication resulting in two new sister cells), by default the winner is taken to be the first cell to which the input gene expression profile vector is compared.

The growth of the network at the end of each cycle is determined by the heterogeneity under each cell, which is computed by its resource R. By definition, the resource is the mean value of the distances among a cell and the gene expression profile vectors associated to it and is given by:

$$R_i = \frac{\sum_{k=1}^{K} d_{\mathbf{x}_k \mathbf{c}_i}}{K} \tag{2.16}$$

where $d_{\mathbf{x}_k \mathbf{c}_i}$ is the distance between cell i and gene expression profile vector k and the summation is done over K gene expression profile vectors asssociated with cell i.

The convergence of the network is controlled by the total error, ϵ_t . This is a measure which determines the proximity of the gene expression profiles to their corresponding best matching cells after an epoch and is given by:

$$\epsilon_t = \sum_i R_i \tag{2.17}$$

A cycle is terminated when the relative increase of the error falls below a given threshold

$$\left| \frac{\epsilon_t - \epsilon_{t-1}}{\epsilon_{t-1}} \right| < E$$

The growing process of the network proceeds by replicating the cell with the largest resource value and is terminated when the heterogeneity of the system falls below a threshold. The heterogeneity

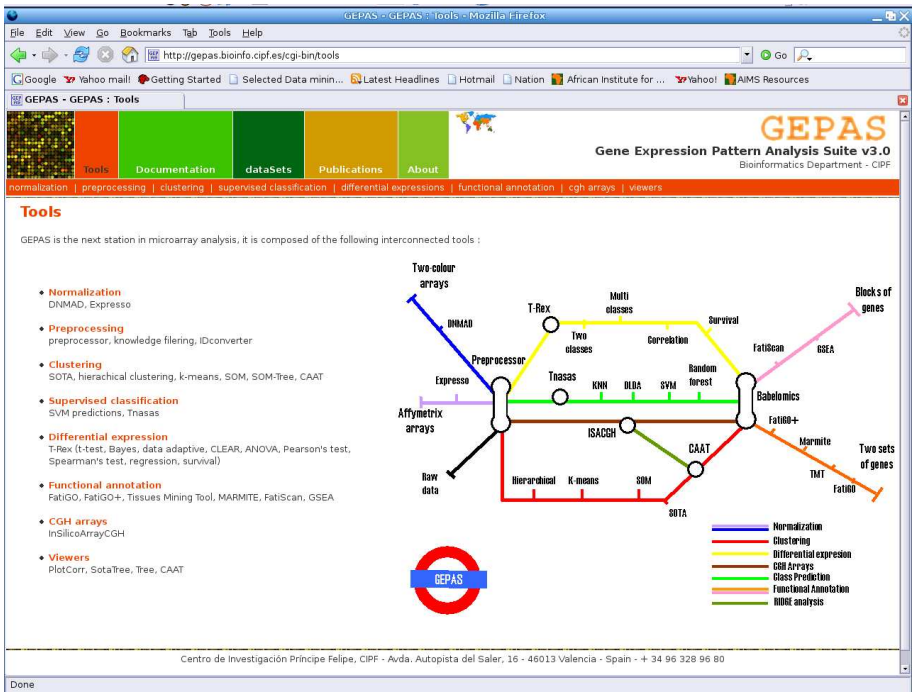


Figure 2.3: GEPAS: Data Analysis Tools

of the system is measured in two ways. One of them is the resource value, R, of the network, which is the maximum resource value among all the cells and the other is variability, V, of the network. Variability is defined as the maximum value among the maximum values, D_i, among all the possible profile-profile distances between all pairs of genes belonging to cell i. Thus, $V = max_i\{D_i\}$, where $D_i = max_{jk}\{d_{\mathbf{x}_j,\mathbf{x}_k}\}$. If the threshold is chosen to be zero, then the network grows until every cell is either associated with one, distinct gene expression vector or several identical gene expression vectors. Otherwise, different threshold values make the network to stop at higher hierarchical levels, clustering in single cells those groups of gene expression vectors whose heterogeneity falls below the threshold.

2.2 Overview of Tools for Microarray Data Analysis

2.2.1 GEPAS

GEPAS, which stand for Gene Expression Pattern Analysis Suite is a publicly-accessible web-based tool for microarray data analysis [15, 16, 17]. It is available at *http://www.gepas.org* [15]. It contains several interconnected data analysis tools implemented as individual modules (see Figure 2.3).

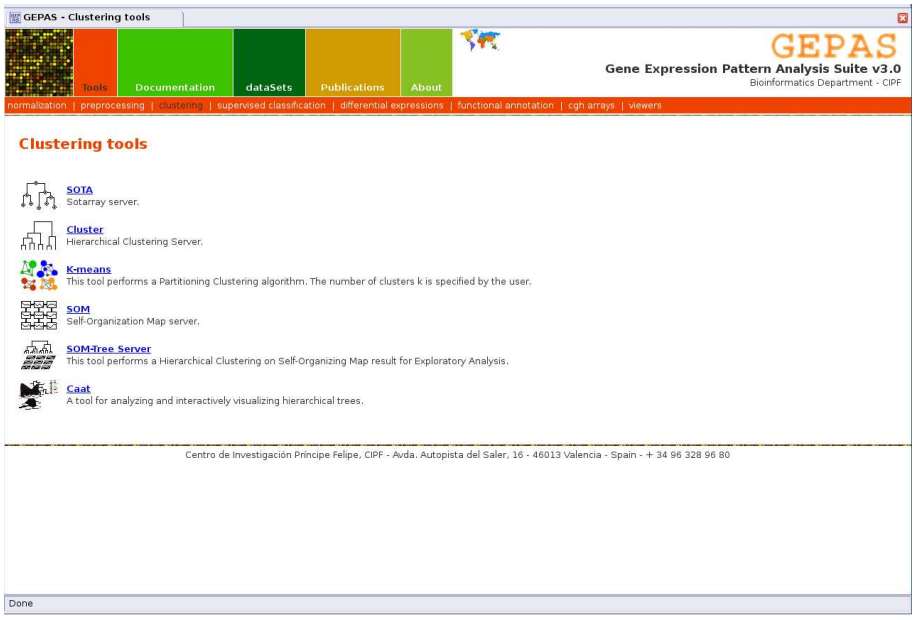


Figure 2.4: GEPAS: Clustering Tools

The modules can be used independently or within a pipeline. Through its various modules, GEPAS allows a variety of analyses to be performed on the microarray data.

When it comes to clustering, with an input raw data file of the following format

	Sample1	Sample2	Sample3	Sample4	Sample5	...	SampleP
Gene 1	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	...	x_{1p}
Gene 2	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	...	x_{2p}
Gene 3	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	...	x_{3p}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Gene n	x_{n1}	x_{n2}	x_{n3}	x_{n4}	x_{n5}	...	x_{np}

where each row represents a gene expression vector, each column a sample or chip, and the entries correspond to the expression levels of a gene (row) in a sample (column). For example, x_{ij} is the expression value of gene i in sample j; GEPAS offers options for agglomerative hierarchical clustering, k-means clustering, self-organising maps clustering and self-organising tree algorithm clustering (see Figure 2.4).

Agglomerative Hierarchical Clustering

GEPAS provides options for performing agglomerative hierarchical clustering on either conditions (samples) or genes or both. For clustering of conditions (upper tree), the options available are

single linkage using Euclidean distance, single linkage using correlation distance, UPGMA using Euclidean distance, UPGMA using correlation distance, complete linkage using Euclidean distance and complete linkage using correlation distance.

Figure 2.5: GEPAS: Agglomerative Hierarchical Clustering Results Output

For clustering of genes, options available for distance (similarity) measures are Euclidean, Euclidean Squared, Linear Correlation Coefficient, Uncentred Linear Correlation Coefficient, Spearman's Rank Correlation Coefficient, and Jackknifed Correlation Coefficient. As far as methods for agglomerative hierarchical clustering of genes are concerned, the options available are single linkage, complete linkage, UPGMA, WPGMA, UPGMC, and WPGMC.

Once the analysis is completed, the results are returned in a new output window (see Figure 2.5), and there are two options available for visualisation of the results.

One option is to send the results to a viewer called TreeView. This option is used to display the results as a tree diagram and a heatmap (grid of coloured points where each colour represents a gene expression value in a sample) (see Figure 2.6).

The tree and heatmap are constructed according to the specifications provided by the user in the TreeView Form, which among other things include the color scheme and appearance.

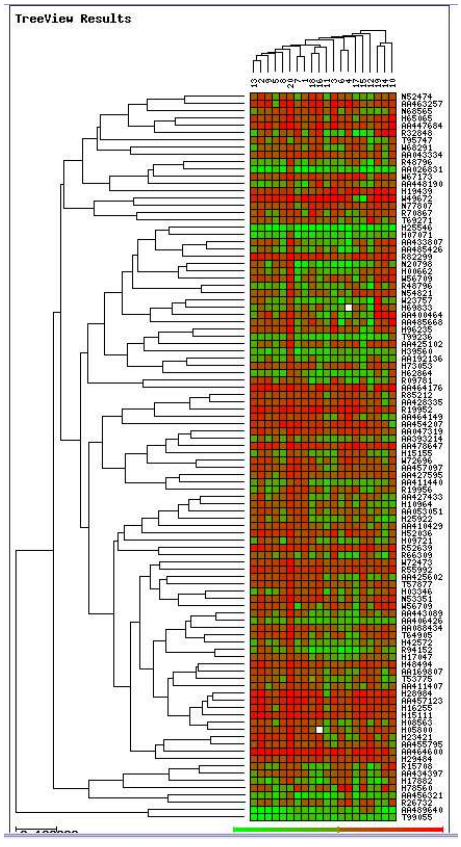


Figure 2.6: GEPAS: TreeView Results

Alternatively, the results can be sent to another tool called
Caat. This tool is used to interactively draw, browse, analyse,
and validate the results of hierarchical clustering. Caat offers three options for drawing trees. The first option is for drawing summary trees; with this option, the user can draw a tree starting from a chosen root node (see Figure 2.7).

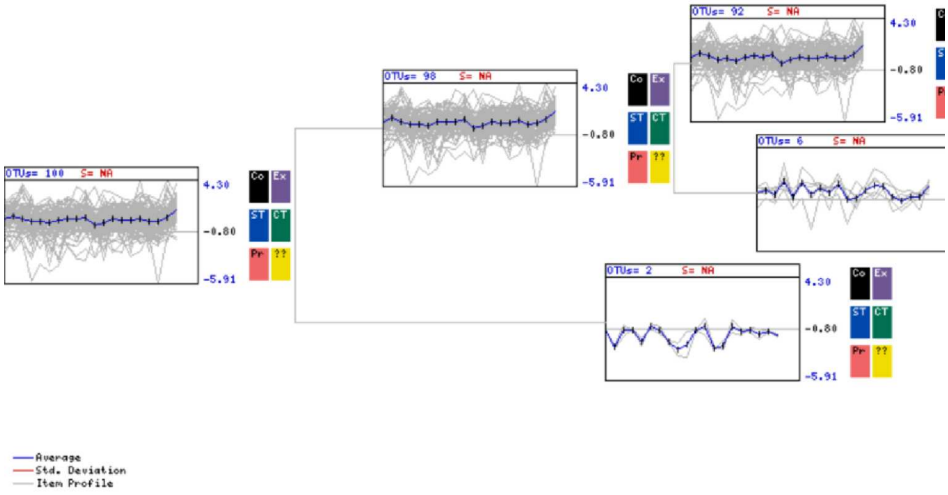


Figure 2.7: GEPAS: Summary Tree with Three Partitions from Caat

At every node of the tree, the user is provided with possibilities to collapse (close the node and hide its children), or expand (open the node and reveal its children), or draw a summary tree starting at that node, or draw a complete tree starting at that node. Furthermore, a click on the node opens a new window containing information for that node. The opened window also provides further options for manipulating the results, which include viewing of a list of genes in the cluster associated with a chosen node and a list of genes in a contrary cluster, as well as sending the list of genes to external tools, for example **FatiGO**, for further analyses (see Figure 2.8).

The other two options are to draw a chosen node as a full expanded tree (see Figure 2.9) or to draw complete trees for the terminal nodes of a summary tree (see Figure 2.10).

Caat also provides a *silhouette* width for each cluster. The *silhouette* is based on the comparison of a cluster's tightness and separation from a neighbouring cluster. Hence, it is used for evaluation of clustering validity and might also be useful in selecting the optimum number of clusters. By definition, the silhouette width of a gene expression vector is

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{2.18}$$

where $a(i)$ is the average dissimilarity of gene i to all other genes within its cluster and $b(i)$ is the

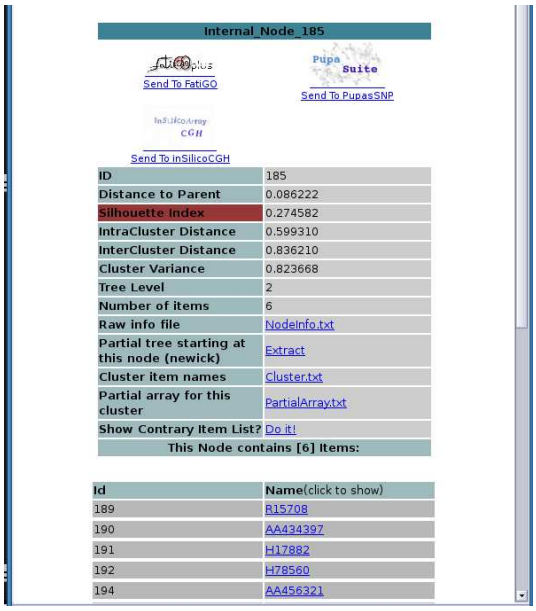


Figure 2.8: GEPAS: Cluster Information Page from Caat

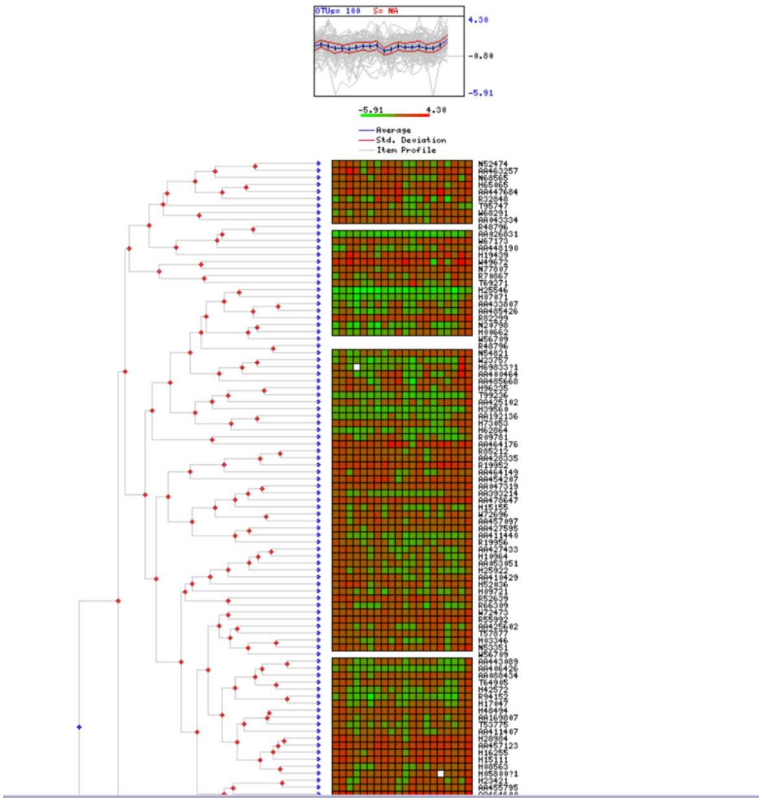


Figure 2.9: GEPAS: Complete Tree of one chosen node from Caat

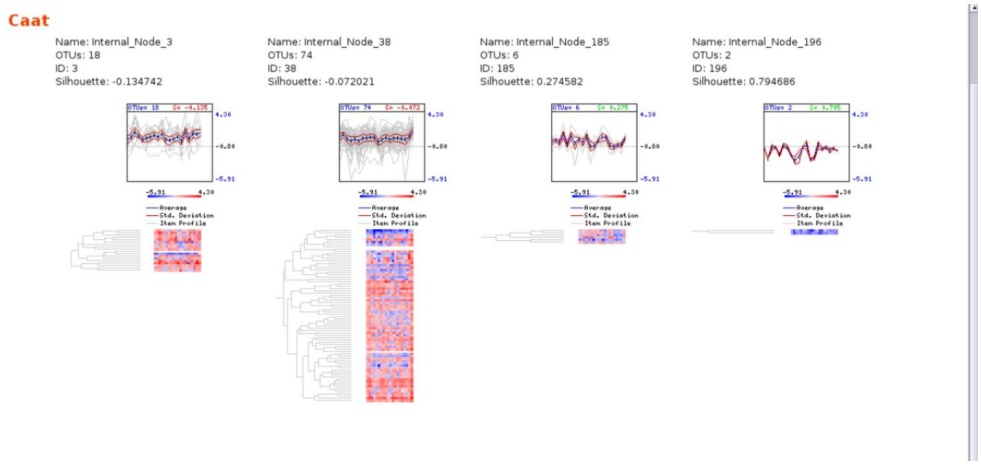


Figure 2.10: GEPAS: Complete Trees of nodes of a Summary Tree from Caat

average dissimilarity of gene i to all genes in its nearest neighbour cluster.

The silhouette width obeys the rule $-1 \leq s(i) \leq 1$. A gene is deemed to be well-clustered if $s(i)$ is close to 1. $s(i)$ close to zero means that the gene could be allocated to a neighbouring cluster equally well since the gene lies equally far away from both clusters. $s(i)$ value close to -1 means that the gene is assigned to a wrong cluster.

The silhouette width of a cluster is calculated by finding the average of the silhouette widths of all the genes in that cluster.

k-means Clustering

For k-means clustering, the measures of distance available in GEPAS are the Euclidean distance, the Pearson correlation coefficient distance and, the Spearman's rank correlation coefficient distance.

Once the analysis is completed, a plot of each of the clusters formed is displayed as part of the output for the results (see Figure 2.11).

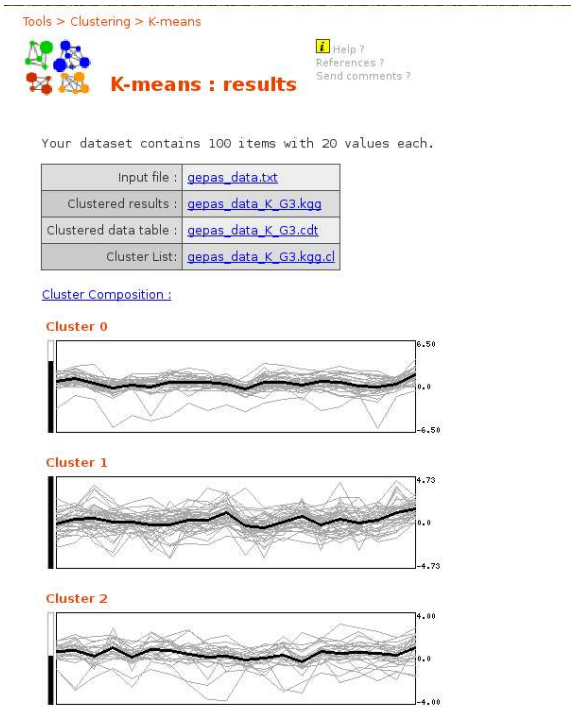


Figure 2.11: GEPAS: k-means Clustering Results Output

In addition, the user is provided with the option of
sending the results to Caat for further exploration and manipulation, which include viewing cluster information and cluster trees, and validating the clustering technique (see Figure 2.12)

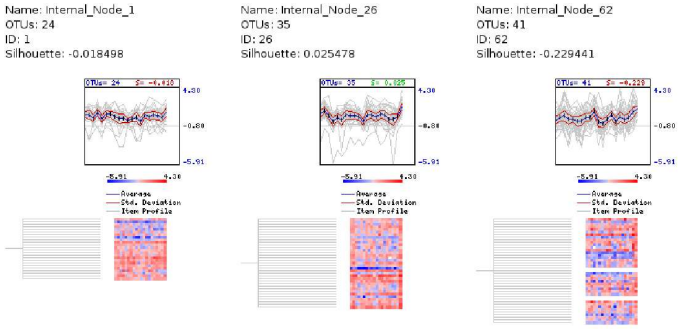


Figure 2.12: GEPAS: k-means Clustering Trees drawn using Caat

Self-Organising Maps Clustering

For self-organising maps clustering, GEPAS offers possibilities for both hexagonal lattice and rectangular lattice topologies. There are also two options for the neighbourhood function: step(bubble) and Gaussian. Figure 2.13 shows the SOM data uploading form.

The results are displayed as a series of interconnected rectangles (see Figure 2.14), each corresponding to a node of the map. A click on a node opens a separate window showing the list of genes in the cluster represented by the node. Also displayed is a plot with the profile of the cluster and profiles of genes in that cluster (see Figure 2.15).

A further option is provided to send the results to Caat. Once sent to Caat, the plots of the clusters can either be displayed in a form of a list (see Figure 2.16), or as a tree (see Figure 2.17).

Self-organising Tree Algorithm Clustering

The Self-organising Tree Algorithm tool in GEPAS offers Euclidean distance, Euclidean distance squared, Uncentred correlation coefficient-based distance, Spearman's correlation cofficient-based distance and Jackkniffed correlation distance as options for distance measures between genes. The tool also allows clustering of conditions, cf. agglomerative hierarchical clustering. Figure 2.18 shows the Self-organising Tree Algorithm Data uploading form.

Once the analysis is done, the results are returned on a page with links to a number of outputs.One

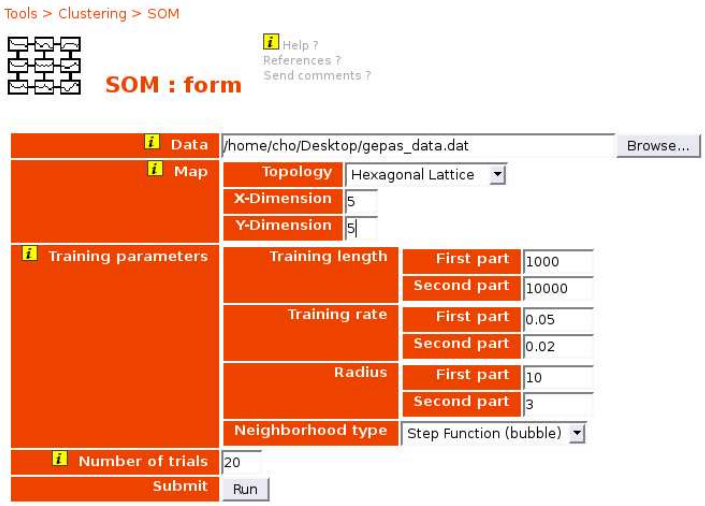


Figure 2.13: GEPAS: Self-organising Map Data Upload Form

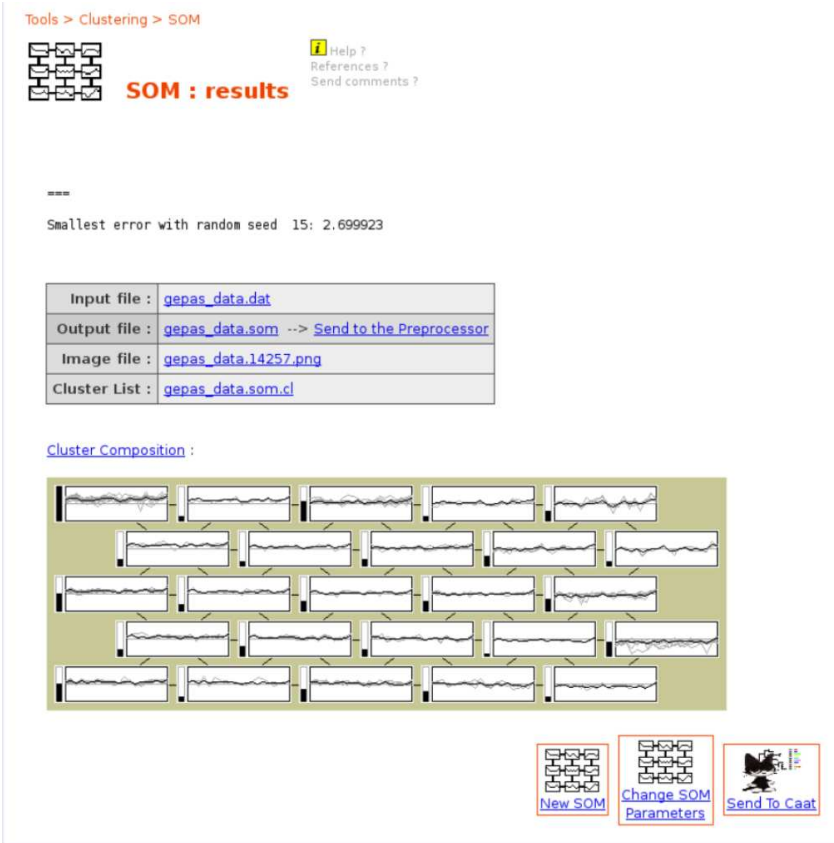


Figure 2.14: GEPAS: Self-organising Map Results Output

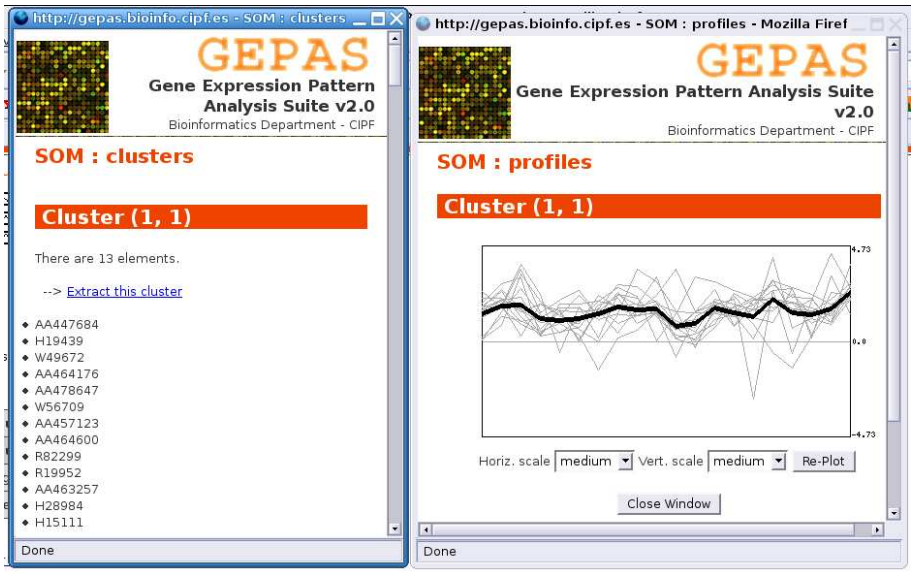


Figure 2.15: GEPAS: Self-organising Map Results Output for a selected node

of the links is the Cluster HTML File, which leads to another page showing the various clusters and their respective member genes (see Figure 2.19). One can extract a cluster from this page and send it to other tools, within GEPAS or external to it for further analysis (see Figure 2.20).

Also included on the results-output links page are options to send the results to SotaTree, TreeView and Caat. The SotaTree option allows the user to view the results as a tree (see Figure 2.21). It also allows the user to view a plot of the profile of a selected cluster (see Figure 2.22) and to extract members of a cluster for further analysis, as shown in Figure 2.20. The TreeView option shows a tree and heatmaps of the clusters (see Figure 2.23). The manipulations that can be done in Caat are the same as discussed above under agglomerative hierarchical clustering.

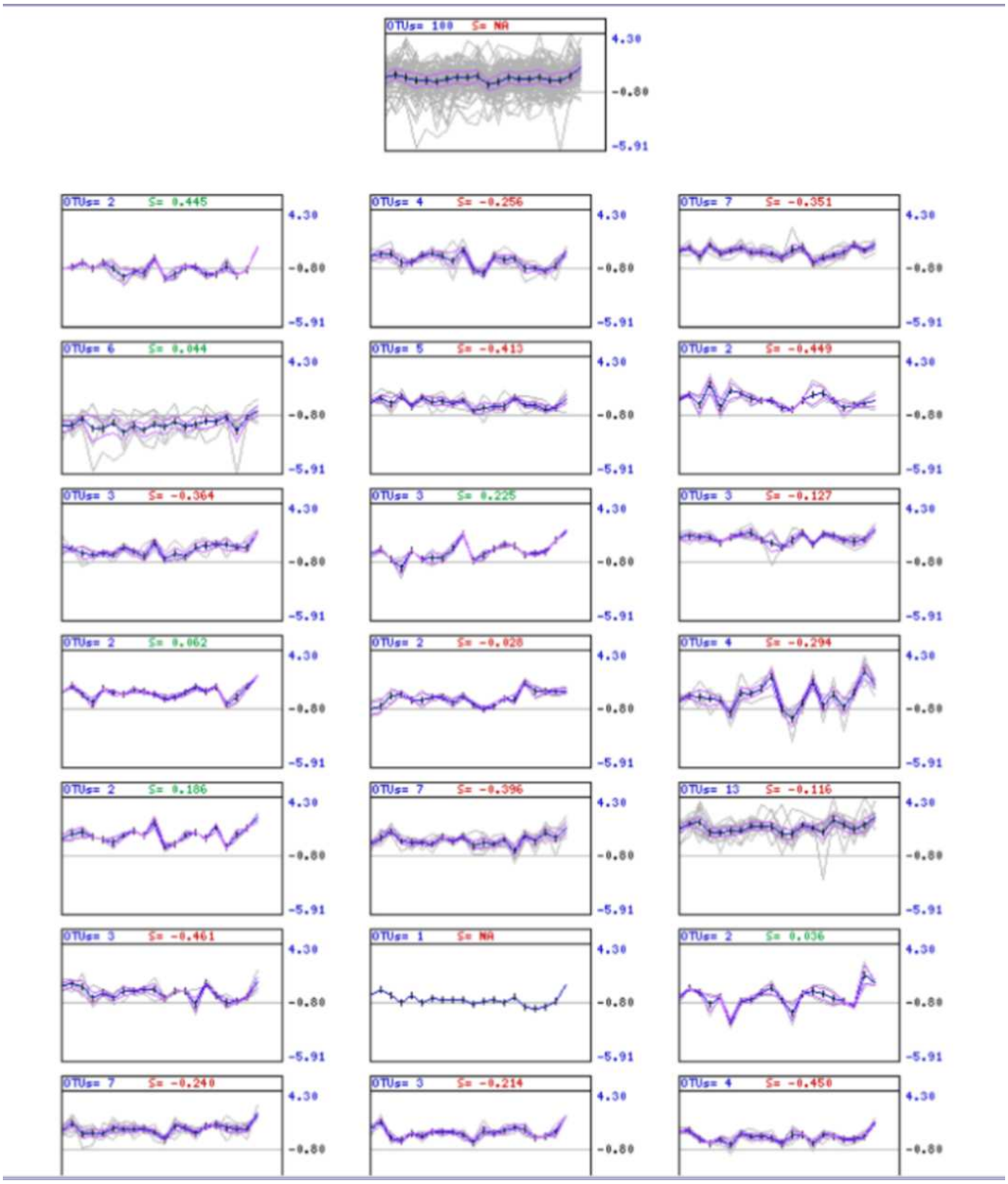


Figure 2.16: GEPAS: Self-organising Map Results Displayed as a List of plots of clusters

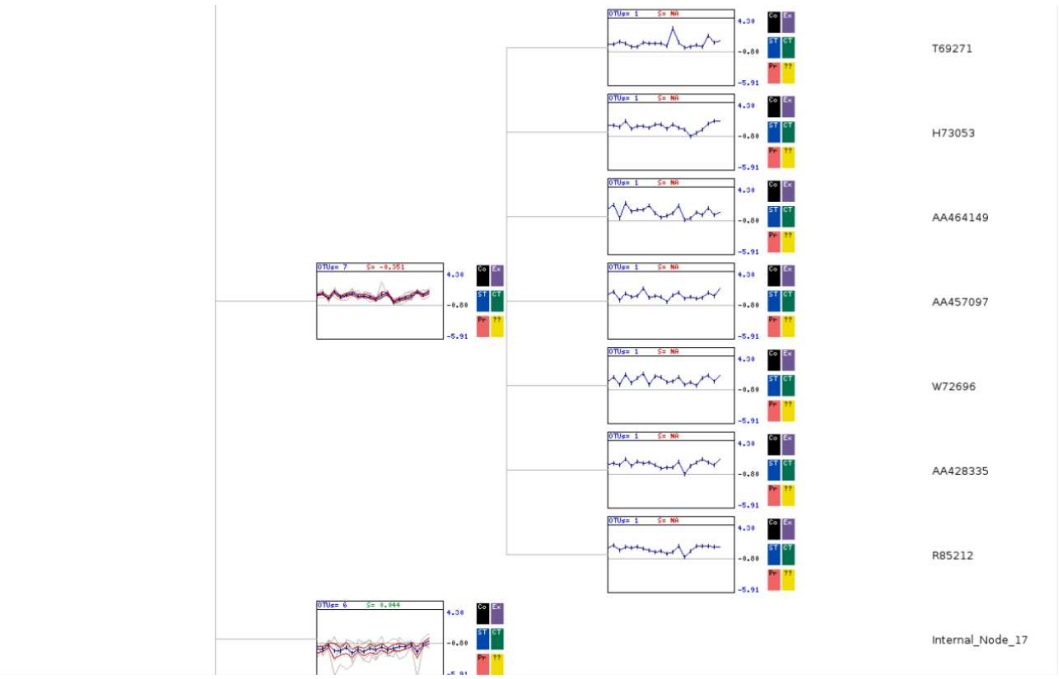


Figure 2.17: GEPAS: Self-organising Map Results Displayed as a Tree

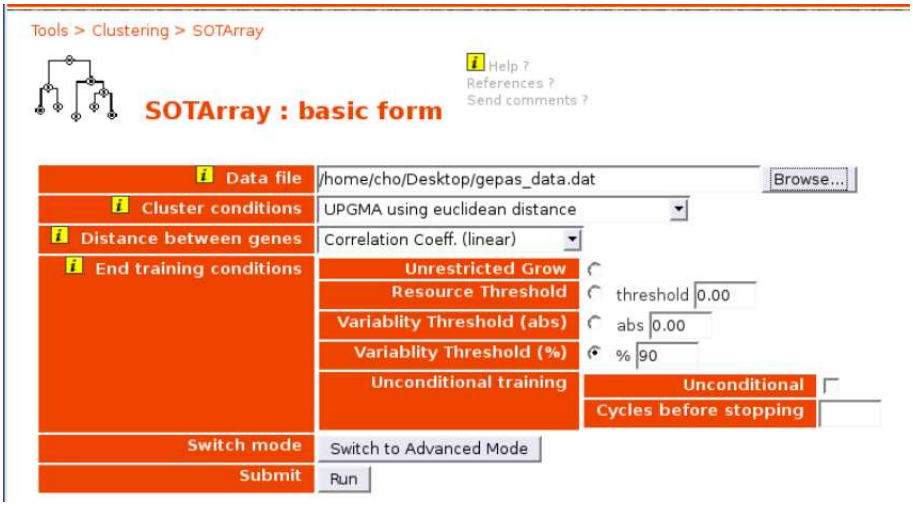


Figure 2.18: GEPAS: Self-organising Tree Algorithm Data Uploading Form

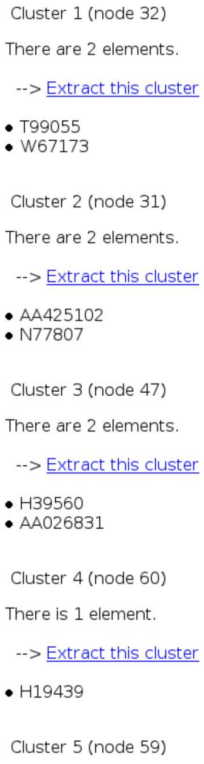


Figure 2.19: GEPAS: Self-organising Tree Algorithm Cluster HTML File

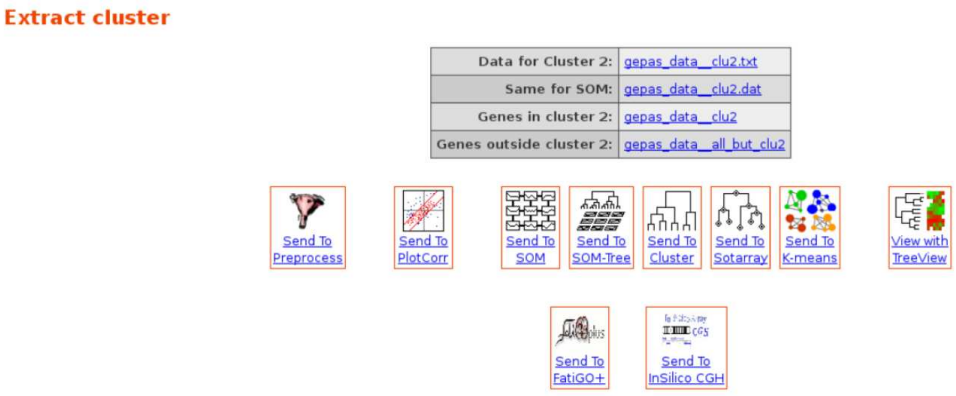


Figure 2.20: GEPAS: Options Available for an Extracted SOTA Cluster

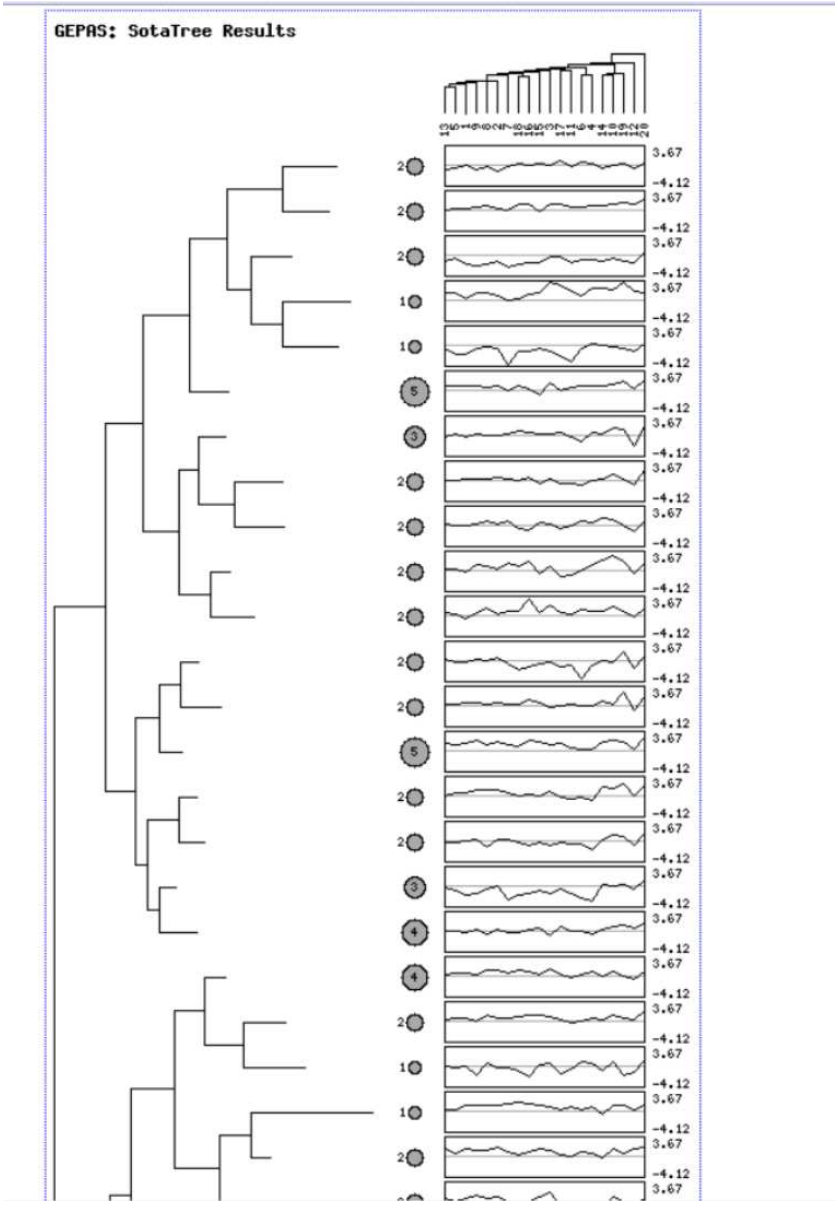


Figure 2.21: GEPAS: Tree from SotaTree

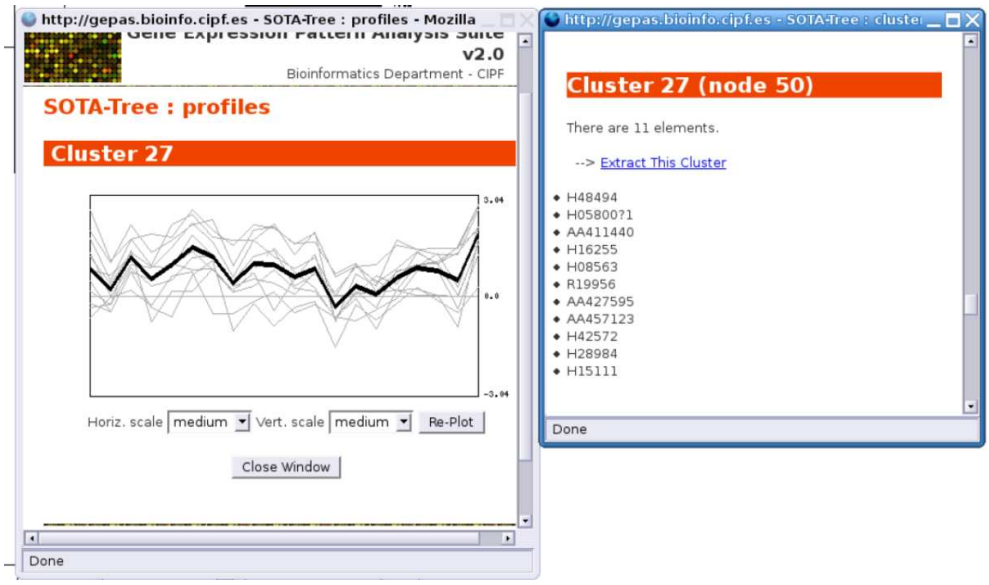


Figure 2.22: GEPAS: Cluster Profile Plot and List of Genes from SotaTree

2.2.2 Expresssion Profiler: Next Generation

Expression Profiler: Next Generation (EP:NG) is a web-based platform for microarray gene expression and other functional genomics-related data analysis. It is available online at *http://www.ebi.ac.uk/expressionprofiler* [18]. Through its chainable components, among other things, it provides possibilities for data transformation and normalization, clustering, pattern discovery, visualisation, and statistical significance testing (see Figure 2.24). On clustering, EP:NG provides agglomerative hierarchical clustering, k-means clustering, and k-mediods clustering options.

Agglomerative Hierachical Clustering

For agglomerative hierarchical clustering, EP:NG allows clustering of genes, or conditions, or both genes and conditions simultaneously. The options available for distance measures are Euclidean distance, Euclidean distance squared, Average distance, Square root of "Average distance", Manhattan distance, Correlation-based distance (Uncentred), Absolute value of correlation-based distance, Linear correlation-based distance (Pearson), Absolute value of Linear correlation-based distance, Chord distance, Euclidean distance on normalised vectors, Spearman's rank correlation-based distance, Number of attributes with opposite sign and Manhattan distance for non zero values. Options provided for clustering algorithms are Average linkage (average distance, UPGMA), Complete link-

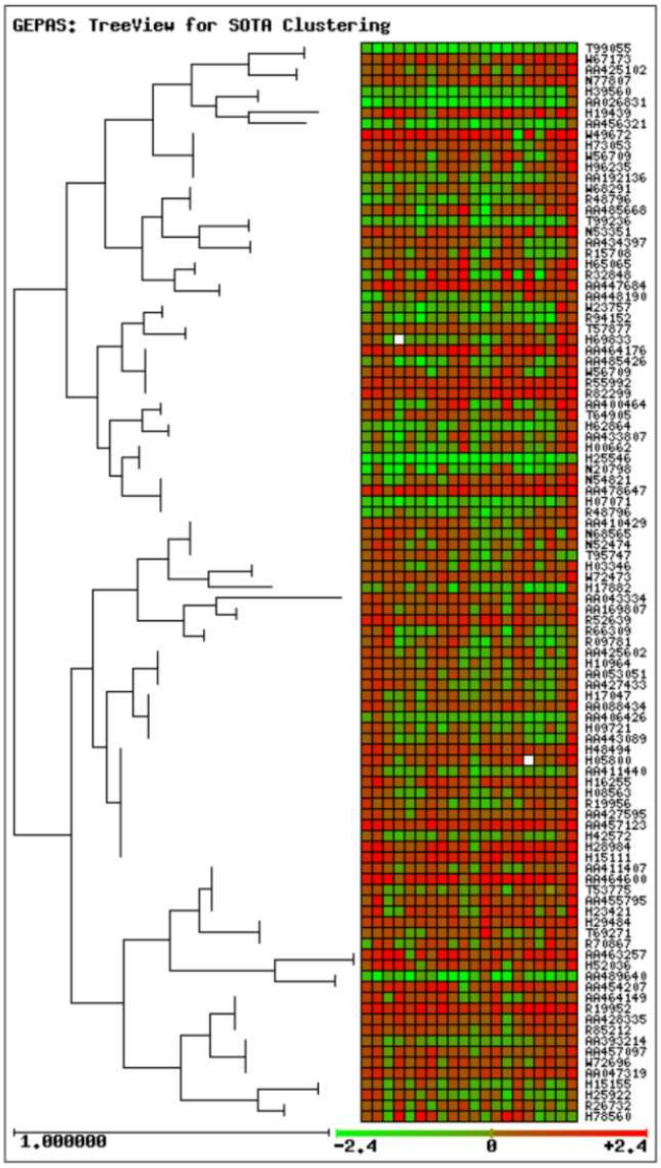


Figure 2.23: GEPAS: TreeView Output for SOTA Clustering

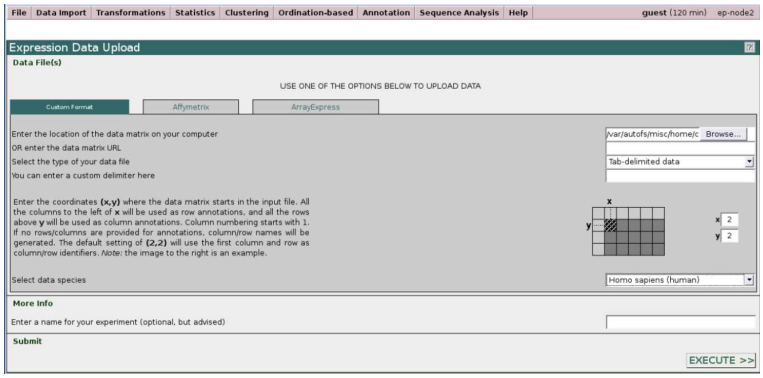


Figure 2.24: Expression Profiler: Data Analysis Options and Data Upload Form

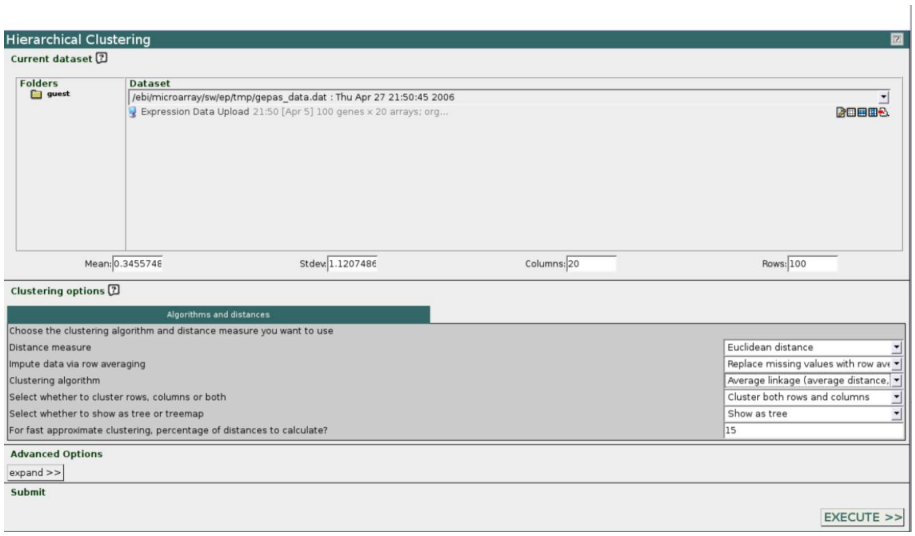


Figure 2.25: Expression Profiler:Agglomerative Hierarchical Clustering Options Form

age (maximum distance), Single linkage (minimum distance) and Average linkage (weighted group average, WPGA). Figure 2.25 shows the agglomerative hierarchical clustering options selection Form.

The results are visually displayed as both a dendrogram and a heatmap. Figure 2.26 shows a visual display of the output results of agglomerative hierarchical clustering of genes. Once the agglomerative hierarchical clustering tree is displayed, the user can easily zoom in on interesting subtrees by clicking on an appropriate node on the tree. This action allows the user to save the list of genes belonging to the cluster forming that subtree.

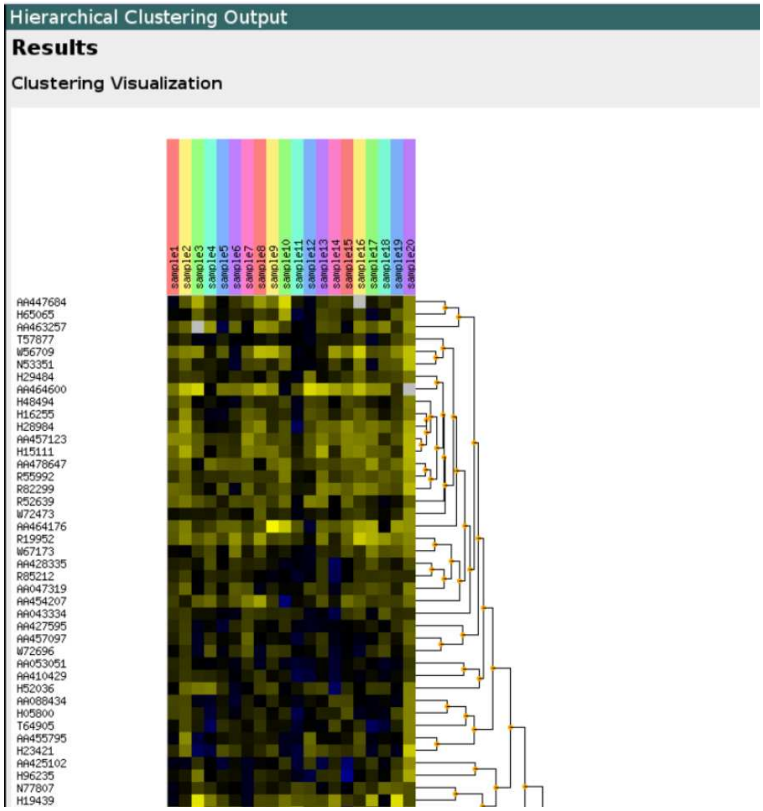


Figure 2.26: Expression Profiler: Agglomerative Hierarchical Clustering Output

k-means and k-mediods Clustering

For k-means, the options available for distance measures are Euclidean distance and correlation measure based distance (uncentred), while those available for k-mediods are the same as in agglomerative hierarchical clustering. EP:NG further provides options for initialisation of the centres of the prespecified k clusters to be obtained. The options available for both techniques are: initialising by most distant (average) genes, initialising by most distant(minimum) genes, and initialising by random genes. The results, from both methods, are displayed visually as heatmaps; k separate heatmaps are shown on the same page. Apart from heatmaps of the clusters, a list of genes belonging to each cluster is also provided. Figure 2.27 shows how the k-means clustering results are displayed.

Clustering Comparison

Lastly, EP:NG also has a clustering comparison component. This component implements an algorithm that takes two *k-groups* clustering results and matches the clusters by membership. This component helps the user to evaluate the optimal number, k, of clusters in the dataset [18].

2.2.3 MIDAW

MIDAW, which stands for Microarray Data Analysis Web Tool, is a web interface that has a series of integrated statistical algorithms that are useful for processing and interpretation of microarray data [19]. The tool is publicly accessible online at *http://muscle.cribi.unipd.it/midaw/*.

As far as clustering is concerned, MIDAW provides options for only agglomerative hierarchical clustering and k-means clustering (see Figure 2.28).

Agglomerative Hierarchical Clustering

On agglomerative hierarchical clustering, for distance measures, the user has a choice between the Euclidean distance and Pearson correlation-based distance. There are three options available for clustering method. These are single linkage, average linkage and complete linkage clustering.

The results are visually displayed in two ways. First, as a dendrogram of clustering by samples; second, as a heatmap of hierarchical clustering on genes (see Figure 2.29). A selection of a region on the heatmap gives an HTML table displaying the expression levels and descriptions of the genes

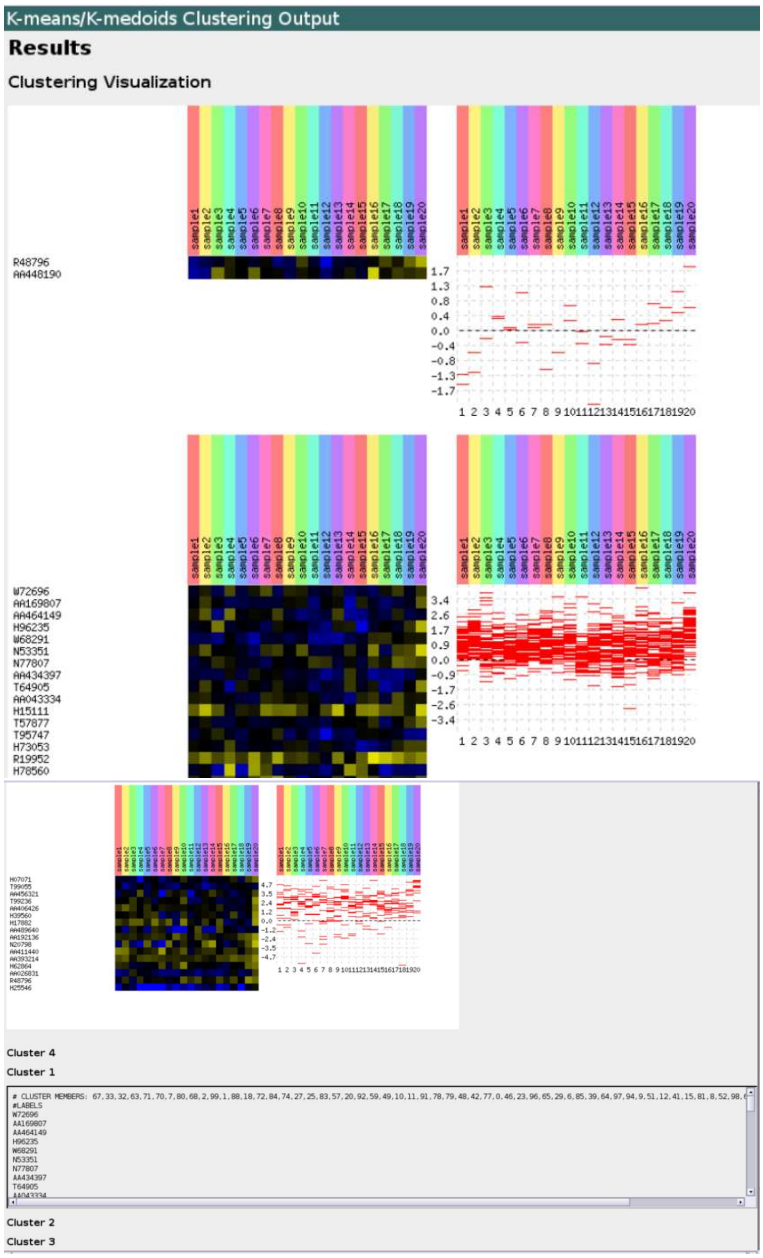


Figure 2.27: Expression Profiler: k-means Clustering Output

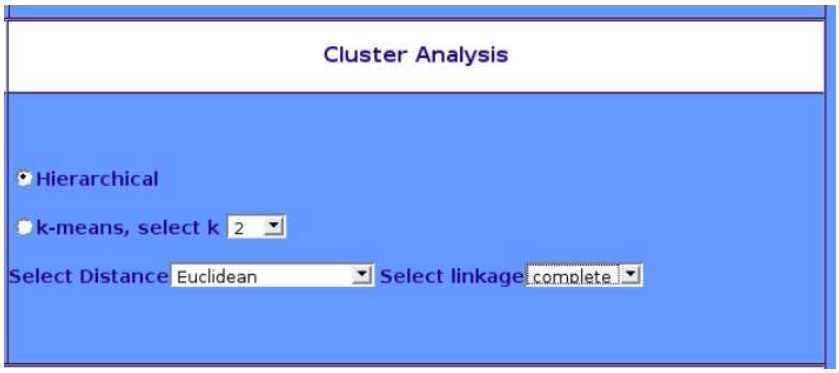


Figure 2.28: MIDAW: Clustering Options

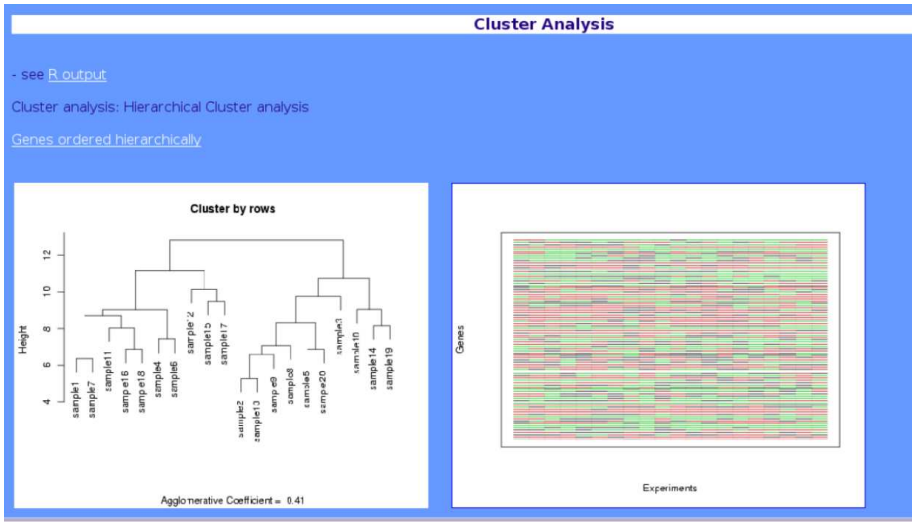
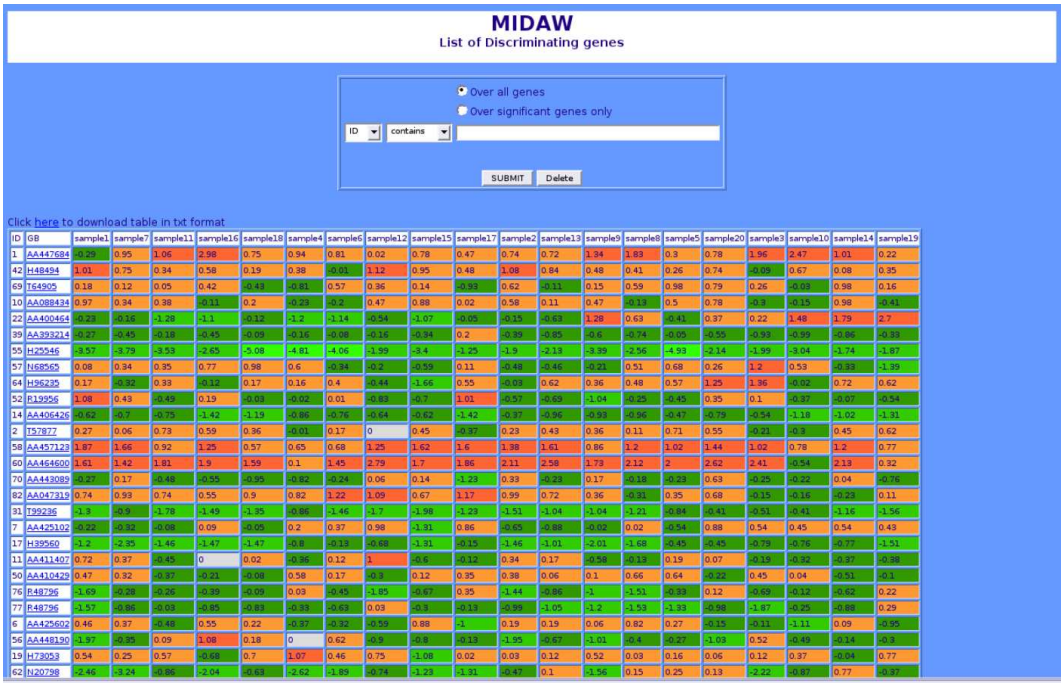


Figure 2.29: MIDAW: Agglomerative Hierarchical Clustering Results Output

of the selected region. Furthermore, the results are also provided as a hierarchically ordered list of discriminating genes (see Figure 2.30).

k-means Clustering

The k-means clustering option in MIDAW uses the same distance measures listed under agglomerative hierarchical clustering. The output for the results are k cluster plots showing the expression profiles of all the genes belonging to cluster $1, \ldots, k$ (see Figure 2.31). In addition, MIDAW provides an HTML table showing all the genes in the whole dataset with their corresponding clusters (see Figure 2.32).



ID	GB	sample1	sample7	sample11	sample16	sample18	sample4	sample6	sample12	sample15	sample17	sample2	sample13	sample9	sample8	sample5	sample20	sample3	sample10	sample14	sample19
1	AA447684	0.29	0.95	1.06	2.98	0.75	0.94	0.81	0.02	0.78	0.47	0.74	0.72	1.34	1.83	0.3	0.78	1.96	2.47	1.01	0.22
42	H48494	1.01	0.75	0.34	0.58	0.19	0.38	-0.01	1.12	0.95	0.48	1.08	0.84	0.48	0.41	0.26	0.74	-0.09	0.67	0.08	0.35
69	T64905	0.18	0.12	0.05	0.42	-0.43	-0.81	0.57	0.36	0.14	-0.93	0.62	-0.11	0.15	0.59	0.98	0.79	0.26	-0.03	0.98	0.16
10	AA088434	0.97	0.34	0.38	-0.11	0.2	-0.23	0.2	0.47	0.88	0.02	0.58	0.11	0.47	-0.13	0.5	0.78	-0.3	-0.15	0.98	-0.41
22	AA400464	-0.23	-0.16	-1.28	-1.1	-0.12	-1.2	-1.14	-0.54	-1.07	-0.05	-0.15	-0.63	1.28	0.69	-0.41	0.37	0.22	1.48	1.79	2.7
39	AA393214	-0.27	-0.45	-0.18	-0.45	-0.09	-0.16	-0.08	-0.16	-0.34	0.2	-0.39	-0.85	-0.6	-0.74	-0.05	-0.55	-0.93	-0.99	-0.86	-0.33
55	H25546	-3.57	-3.79	-3.53	-2.65	-5.08	-4.81	-4.06	-1.99	-3.4	-1.25	-1.9	-2.13	-3.39	-2.56	-4.93	-2.14	-1.99	-3.04	-1.74	-1.87
57	N68565	0.08	0.34	0.95	0.77	0.98	0.6	-0.34	-0.2	-0.59	0.11	-0.48	-0.46	-0.21	0.51	0.68	0.26	1.2	0.53	-0.33	-1.39
64	H96235	0.17	-0.32	0.33	-0.12	0.17	0.16	0.4	-0.44	-1.66	0.55	-0.03	0.62	0.36	0.48	0.57	1.25	1.36	-0.02	0.72	0.62
52	R19956	1.08	0.43	-0.49	0.19	-0.03	-0.02	0.01	-0.83	-0.7	1.01	-0.57	-0.69	-1.04	-0.25	-0.45	0.35	0.1	-0.37	-0.07	-0.54
14	AA406426	-0.62	-0.7	-0.75	-1.42	-1.19	-0.86	-0.76	-0.64	-0.62	-1.42	-0.97	-0.96	-0.93	-0.96	-0.47	-0.79	-0.54	-1.18	-1.02	-1.31
2	T57877	0.27	0.06	0.73	0.59	0.36	-0.01	0.17	0	0.45	-0.37	0.23	0.43	0.36	0.11	0.71	0.55	-0.21	-0.3	0.45	0.62
58	AA457123	1.87	1.66	0.92	1.25	0.57	0.65	0.68	1.25	1.62	1.6	1.38	1.61	0.86	1.2	1.02	1.44	1.02	0.78	1.2	0.77
60	AA464600	1.61	1.42	1.81	1.9	1.59	0.1	1.45	2.79	1.7	1.86	2.11	2.58	1.73	2.12	2	2.62	2.41	0.54	2.13	0.32
70	AA443089	-0.27	0.17	-0.48	-0.55	-0.95	-0.82	-0.24	0.06	0.14	-1.23	0.33	-0.23	0.17	-0.18	-0.23	0.63	-0.25	-0.22	0.04	-0.76
82	AA047319	0.74	0.93	0.74	0.55	0.9	0.82	1.22	1.09	0.67	1.17	0.99	0.72	0.36	-0.31	0.35	0.68	-0.15	-0.16	-0.23	0.11
31	T99236	-1.3	-0.9	-1.78	-1.49	-1.35	-0.86	-1.46	-1.7	-1.98	-1.23	-1.51	-1.04	-1.04	-1.21	-0.84	-0.41	-0.51	-0.41	-1.16	-1.56
7	AA425102	-0.22	-0.32	-0.08	0.09	-0.05	0.2	0.37	0.98	-1.31	0.86	-0.65	-0.88	-0.02	0.02	-0.54	0.88	0.54	0.45	0.54	0.43
17	H39560	-1.2	-2.35	-1.46	-1.47	-1.47	-0.8	-0.13	-0.68	-1.81	-0.15	-1.46	-1.01	-2.01	-1.68	-0.45	-0.45	-0.79	-0.76	-0.77	-1.51
11	AA411407	0.72	0.37	-0.45	0	0.02	-0.95	0.12	1	-0.6	-0.12	0.34	0.17	-0.58	-0.13	0.19	0.07	-0.19	-0.32	-0.37	-0.98
50	AA410429	0.47	0.32	-0.37	-0.21	-0.08	0.58	0.17	-0.3	0.12	0.35	0.38	0.06	0.1	0.66	0.64	0.22	0.45	0.04	-0.51	-0.1
76	R48796	-1.69	-0.28	-0.26	-0.39	-0.09	0.03	-0.45	-1.85	-0.67	0.35	-1.44	-0.86	-1	-1.51	-0.33	0.12	-0.69	-0.12	-0.62	0.22
77	R48796	-1.57	-0.86	-0.03	-0.85	-0.83	-0.33	-0.63	0.03	-0.3	-0.13	-0.99	-1.05	-1.2	-1.53	-1.33	-0.98	-1.87	-0.25	-0.88	0.29
6	AA425602	0.46	0.37	-0.48	0.55	0.22	-0.37	-0.32	-0.59	0.88	-1	0.19	0.19	0.06	0.82	0.27	-0.15	-0.11	-1.11	0.09	-0.95
56	AA448190	-1.97	-0.35	0.09	1.08	0.18	0	0.62	-0.9	-0.8	-0.13	-1.95	-0.67	-1.01	-0.4	-0.27	-1.03	0.52	-0.49	-0.14	-0.9
19	H73053	0.54	0.25	0.57	-0.68	0.7	1.07	0.46	0.75	-1.08	0.02	0.03	0.12	0.52	0.03	0.16	0.06	0.12	0.37	-0.04	0.77
62	N20798	-2.46	-3.24	-0.86	-2.04	-0.63	-2.62	-1.89	-0.74	-1.23	-1.31	-0.47	0.1	-1.56	0.15	0.25	0.13	-2.22	-0.87	0.77	-0.97

Figure 2.30: MIDAW: Hierarchical Clustering Table of Discriminating Genes

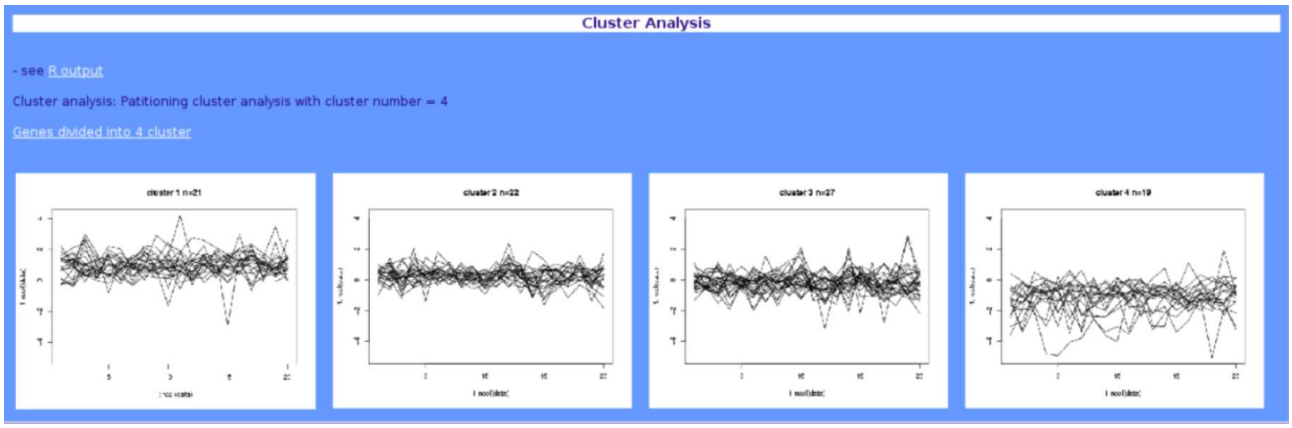


Figure 2.31: MIDAW: k-means Clustering Results Output

Figure 2.32: MIDAW: k-means Clustering Table of Discriminating Genes

Chapter 3

Comparison Analysis of the Performance of Tools for Microarray Data Analysis

Apart from exploring the possibilities that the three tools provide for clustering, it was of great interest also to get a clear picture of their performances with respect to one another. This chapter, therefore, is dedicated to present a comparison analysis of the performance of the three tools.

To assess the performance of the three tools with respect to each other, a dataset consisting of gene expression values of 200 genes measured in 28 samples was picked from the Sorlie-Breast-PNAS-2001-Project breast cancer dataset [31] and analysed using the same clustering techniques. One of the analyses performed was agglomerative hierarchical clustering of conditions (samples). Complete linkage clustering using linear correlation coefficient distance was done on the samples. The results from the three tools are shown in Figure 3.1. Cutting the trees at the fourth node from the root node leads to five clusters shown in Table 3.1.

The results indicate that GEPAS and Expression Profiler produced the same clusters. On the contrary, MIDAW produced different clusters. However, there were some cluster members which were the same in all the clusters produced by the three tools. For instance, 7 out of the 10 samples clustered together in cluster 1 from GEPAS and Expression Profiler were also clustered together in cluster 1 from MIDAW which had 11 members and cluster 3 from MIDAW contained the same members as cluster 4 from GEPAS and Expression Profiler.

Agglomerative hierarchical clustering of the genes performed with GEPAS and Expression Profiler

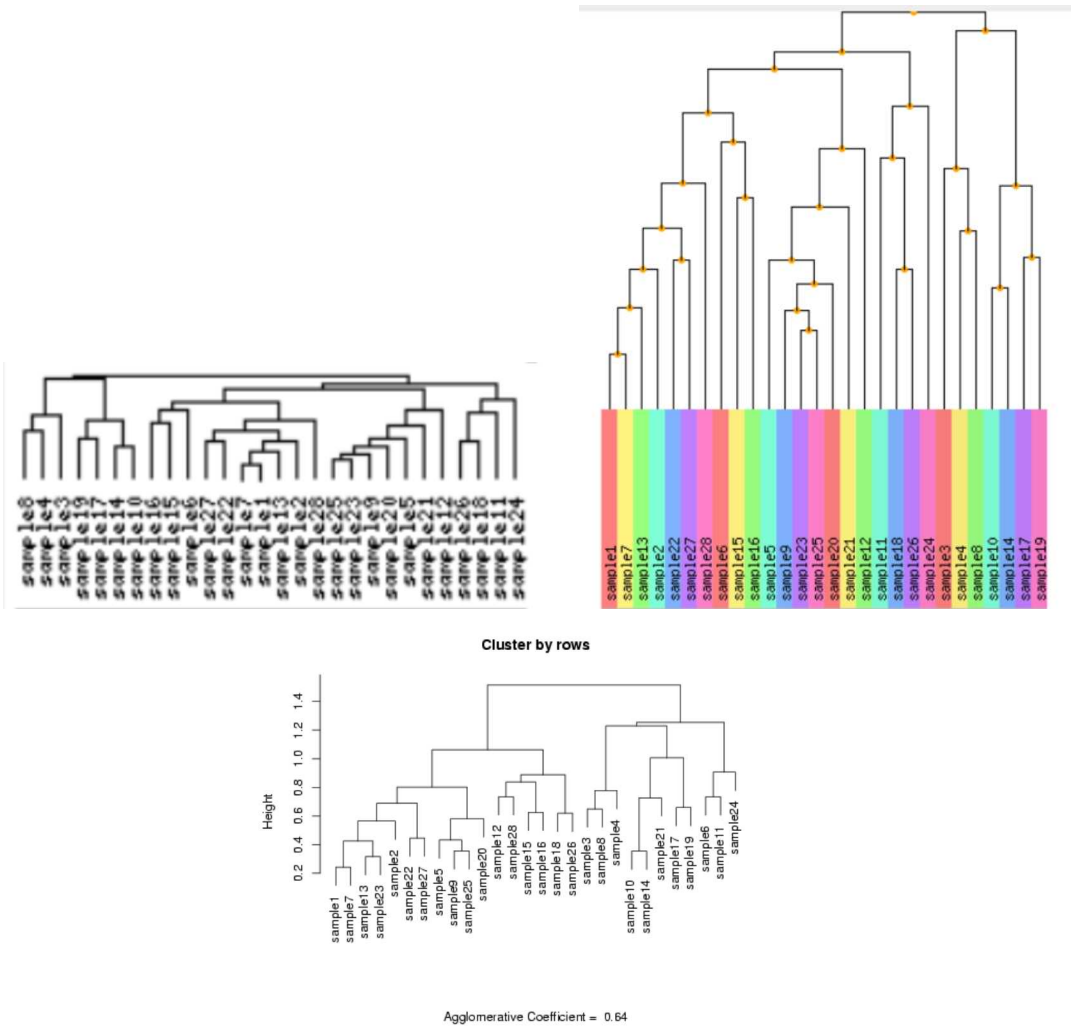


Figure 3.1:

Agglomerative Hierarchical Clustering of Samples

Top left: Results from GEPAS

Top right: Results from Expression Profiler

Bottom: Results from MIDAW

using again complete linkage and linear correlation coefficient based distance further revealed that cutting the tree diagrams generated by the two tools at any level yielded the same results. Figure 3.2 and Figure 3.3 show some of the trees of the congruent clusters produced by the two tools.

Further comparison of the performance of the tools was made with regard to *k-means* clustering. Taking $k = 5$, the number of genes in each of the five clusters generated by each of the three tools are summarised in the Table 3.2.

The results indicated that the three tools produced different results, but there were some common genes in some of the clusters. For example, out of the 21 genes in cluster 4 from GEPAS, 17 matched those in cluster 2 from MIDAW. Also out of 69 genes in cluster 2 from GEPAS, 42 matched those in cluster 1 from MIDAW. Furthermore, all the 21 genes in cluster 4 from GEPAS belonged to cluster 3 from Expression Profiler. Lastly, all 69 genes in cluster 2 from GEPAS were also part of the 117 genes in cluster 1 from Expression Profiler.

Table 3.1: **Five Clusters from Agglomerative Hierarchical Clustering of Samples**

TOOL	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
GEPAS	1, 2, 6, 7, 13, 15, 16, 22, 27, 28	5, 9, 12, 20, 21, 23, 25	11, 18, 24, 26	3, 4, 8	10, 14, 17, 19
Expression Profiler	1, 2, 6, 7, 13, 15, 16, 22, 27, 28	5, 9, 12, 20, 21, 23, 25	11, 18, 24, 26	3, 4, 8	10, 14, 17, 19
MIDAW	1, 2, 5, 7, 9, 13, 20, 22, 23, 25, 27	12, 15, 16, 18, 26, 28	3, 4, 8	10, 14, 17, 19, 21	6, 11, 24

The numbers represent the sample numbers shown on the tree diagrams

Table 3.2: **k-means Clustering Results Summary**

TOOL	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
GEPAS	82	69	23	21	5
Expression Profiler	117	55	23	3	2
MIDAW	66	65	37	23	9

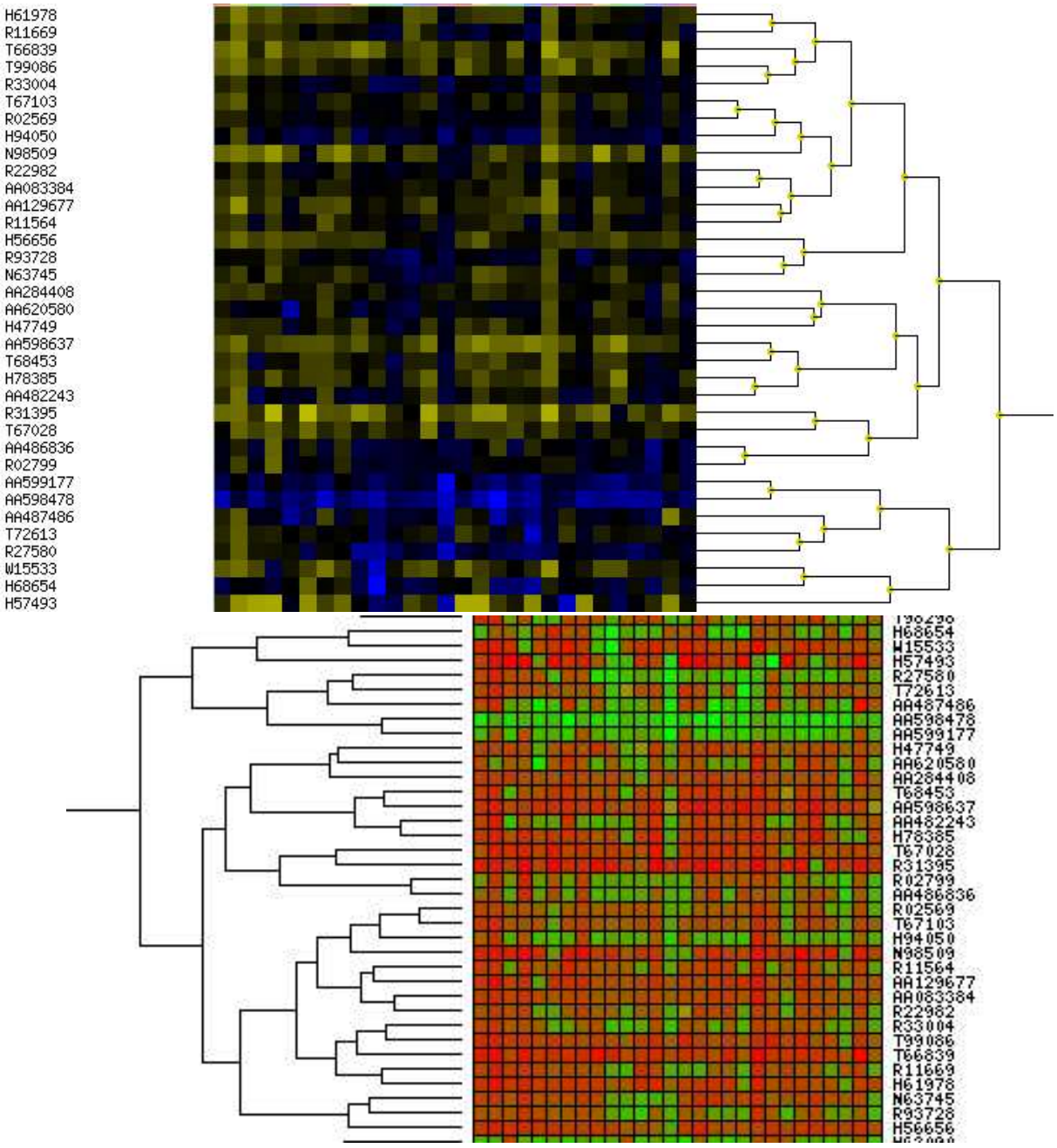


Figure 3.2:

Agglomerative Hierarchical Clustering of Genes

Top: Results from Expression Profiler, Bottom: Results from GEPAS

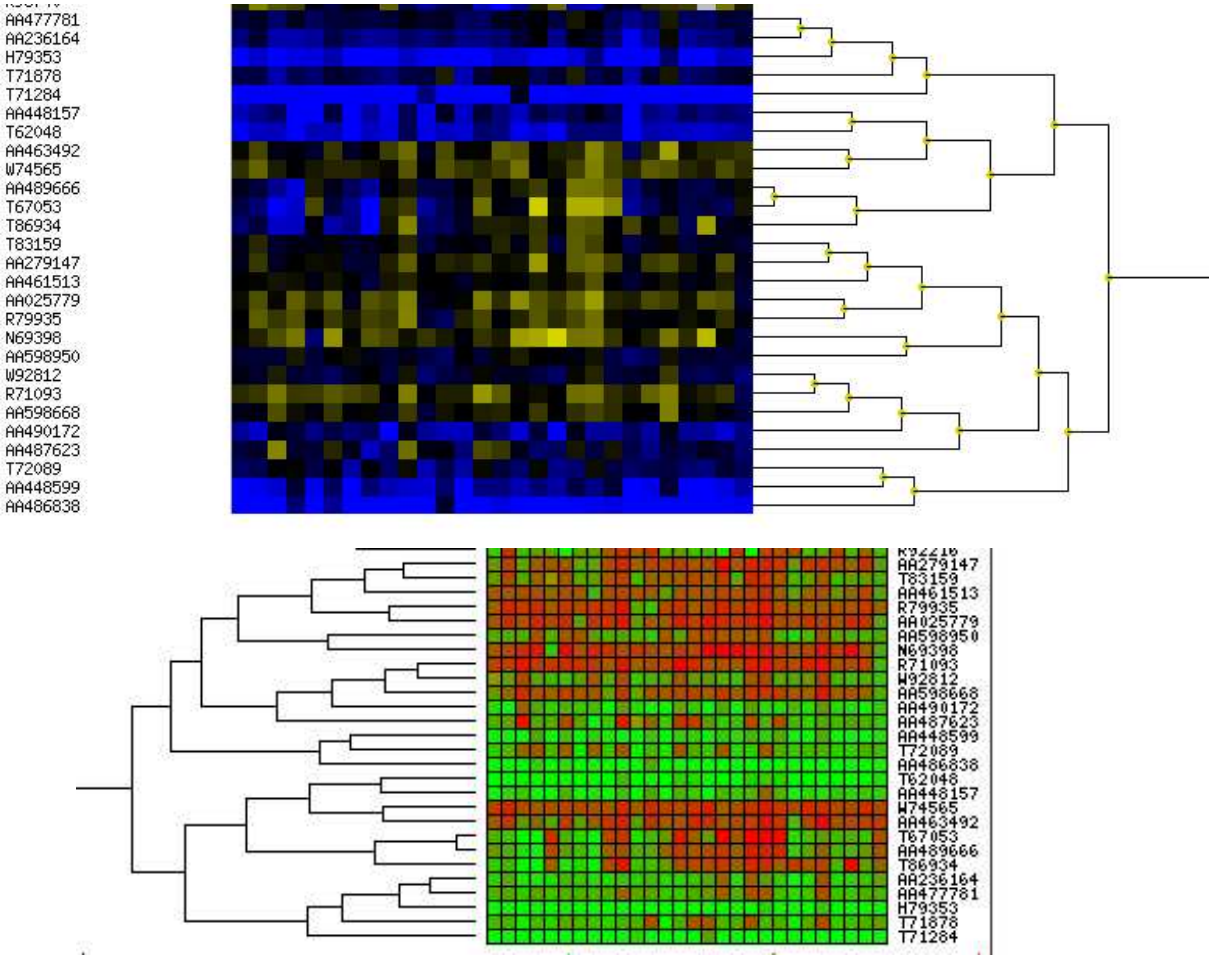


Figure 3.3:

Agglomerative Hierarchical Clustering of Genes

Top: Results from Expression Profiler, Bottom: Results from GEPAS

Chapter 4

Discussions and Conclusion

Microarray technology has become a standard tool in biomedical research because of its exceptional ability to allow scientists to study tens of thousands of genes simultaneously. There are a number of different variations on the microarray technology and the technology is ever advancing. As well as advancement in the microarray technology, substantial progress has been made in techniques and tools for microarray data analysis. There exists various types of analyses of the microarray data and a variety of public tools. In this essay we have presented an overview of three publicly-accessible web-based tools for microarray data analysis, namely GEPAS, Expression Profiler: Next Generation, and MIDAW. In particular, the discussion has been focussed on the possibilities that each of these tools provide for performing clustering.

Much effort has been devoted to exploring the possibilities that each of the three tools provide for clustering. The emphasis has been on measures of (dis)similarity, clustering techniques, visualisation of results and options for further analyses. It has been noticed that all the three tools provide options for both hierarchical clustering and non-hierarchical clustering. However, there are marked differences in the options they provide. GEPAS and Expression Profiler have been found to offer more options on measures of (dis)similarity than MIDAW. All the tools provide the option for k-means clustering, but it is only Expression Profiler that offer the option for k-mediods clustering. Furthermore, GEPAS is the only tool amongst the three which provides options for Self-organising maps and Self-organising tree algorithm clustering techniques. On visualisation of results, both GEPAS and Expression Profiler produce tree diagrams for the results of clustering done on conditions, genes, or both conditions and genes. On the contrary, MIDAW only produces a tree diagram for the results of clustering of conditions. In addition, GEPAS is the only tool of the three which provides more options for further manipulation and analyses of the results. The

GEPAS tool among other things provides options for collapsing and expanding the tree diagrams, and sending the results to other tools within GEPAS or external to it such as FatiGO for further analyses. Overall, in terms of options for performing clustering, GEPAS provides more options followed by Expression Profiler, and finally MIDAW. Hence, in the context of possibilities, GEPAS has an edge over the other two tools. Since there is no clustering method that can suit all situations and that different clustering techniques or even different parameters of the same clustering technique reveal different relationships between the genes or samples in the data [13, 21], the more options the better. Thus, GEPAS allows the user to explore the data much more and discover other interesting relationships that might be missed by the other tools.

Lastly, a comparison analysis of the performance of the three tools was also made. A dataset comprising gene expression values of 200 genes measured in 28 samples picked from the Sorlie-Breast-PNAS-2001-Project [31] breast cancer dataset was analysed with all three tools using the same measures of (dis)similarity and clustering techniques. The analyses performed were complete linkage agglomerative hierarchical clustering of samples (conditions) and genes using linear correlation distance, and k-means clustering. It was observed that on both agglomerative hierarchical clustering of samples (conditions) and genes, GEPAS and Expression Profiler lead to the same clusters. This suggests that the two tools, GEPAS and Expression Profiler, implement the same algorithm for complete linkage agglomerative hierarchical clustering. However, the three tools produced different results for k-means clustering. Nevertheless, some genes were common in the resulting clusters from all the three tools, with GEPAS and Expression Profiler having more common genes in their clusters. All in all, the results of the comparison anlysis of the performance of the three tools illustrated that the results of clustering are influenced by the tool used to perfom the analysis. Different tools lead to different clusters even if the same measures of (dis)similarity and clustering techniques are used, because of variations in the way different tools implement algorithms for various clustering methods. However, the fact that the tools produced different clusters, even when the same measures of similarity and clustering techniques were used, does not tell us which tool is superior. The only way the performance of the tools can be conclusively assessed is through biological validation of the clusters they produce.

In a nutshell, it is hoped that this work has provided a clear picture of the possibilities that the three tools: GEPAS, Expression Profiler and MIDAW, provide for microarray data analysis with emphasis on clustering. Through the properties and usefulness of each of the tools presented here, scientists in microarray research may now be able to make an informed choice on which tool, among the three, to use for different clustering techniques.

Appendix A

Example 1: Agglomerative Hierarchical Clustering Technique.

In this example we illustrate how the agglomerative hierarchical clustering technique works. We consider one of the average linkage clustering algorithms; the Unweighted Pair-Group Method Centroid (UPGMC).

Suppose we have a microarray data consisting of 5 genes whose expression values are measured in 3 samples (conditions) as shown below:

	sample1	sample2	sample3
Gene 1	-3.06	-2.25	-1.15
Gene 2	-1.36	-0.67	-0.17
Gene 3	-0.17	0.48	1.23
Gene 4	1.16	-0.27	0.71
Gene 5	2.09	2.12	2.62

We start by illustrating the calculation of the distance matrix using the Euclidean distance as a measure of similarity.

Using the formula for Euclidean distance, the distance between gene expression vectors:

1 and 2 is $d(1,2) = \sqrt{(-3.06-(-1.36))^2 + (-2.25-(-0.67))^2 + (-1.15-(-0.17))^2} = 6.3468$.

$\vdots$

4 and 5 is $d(4,5) = \sqrt{(1.16-2.09)^2 + (-0.27-2.12)^2 + (0.71-2.62)^2} = 10.2251$.

So, the pair-wise distance matrix for the gene expression vectors in our dataset is as shown below:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0.0 & 6.3468 & 21.4694 & 25.1884 & 59.8323 \\ 6.3468 & 0.0 & 4.6986 & 7.2848 & 27.4707 \\ 21.4694 & 4.6986 & 0.0 & 2.6018 & 9.7293 \\ 25.1884 & 7.2848 & 2.6018 & 0.0 & 10.2251 \\ 59.8323 & 27.4707 & 9.7293 & 10.2251 & 0.0 \end{pmatrix} \end{array}$$

The numbers outside the borders of the matrix represent genes while the entries in the matrix correspond to the distances between the corresponding genes.

Next, we are going to use the Unweighted Pair-Group Method Centroid (UPGMC) algorithm to hierarchically cluster the genes. Initially we treat each gene in the pair-wise distance matrix above as a cluster and merge the pair of closest gene clusters. In this case we merge gene clusters 3 and 4 whose distance is 2.6018 to form a new cluster (34).

We proceed by computing distances between the new cluster (34) and the other clusters using the UPGMC algorithm.

$d((34),1) = \frac{1}{2}(d(3,1) + d(4,1)) = \frac{1}{2}(21.4694 + 25.1884) = 23.3289$

$d((34),2) = \frac{1}{2}(d(3,2) + d(4,2)) = \frac{1}{2}(4.6986 + 7.2848) = 5.9917$

$d((34),5) = \frac{1}{2}(d(3,5) + d(4,5)) = \frac{1}{2}(9.7293 + 10.2251) = 9.9772$

We then update the pair-wise distance matrix so that it becomes:

$$\begin{array}{c} \\ (34) \\ 1 \\ 2 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccc} (34) & 1 & 2 & 5 \end{array} \\ \begin{pmatrix} 0.0 & 23.3289 & 5.9917 & 9.9772 \\ 23.3289 & 0.0 & 6.3468 & 59.8323 \\ 5.9917 & 6.3468 & 0.0 & 27.4707 \\ 9.9772 & 59.8323 & 27.4707 & 0.0 \end{pmatrix} \end{array}$$

We repeat the process by finding the smallest distance between pairs of clusters.

From the updated pair-wise distance matrix above, the smallest distance between pairs of clusters in $d((34),2) = 5.9917$. So, we merge cluster 2 with cluster (34) to form a new cluster (234).

Once again, the distances to all other clusters from this new cluster are calculated.

$d((234),1) = \frac{1}{2}(d(2,1) + d((34),1)) = \frac{1}{2}(6.3468 + 23.3289) = 14.83785$

$d((234),5) = \frac{1}{2}(d(2,5) + d((34),5) = \frac{1}{2}(27.4707 + 9.9772) = 18.72395$

Then the updated pair-wise distance matrix becomes:

$$\begin{array}{c} \\ (234) \\ 1 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccc} (234) & 1 & 5 \end{array} \\ \begin{pmatrix} 0.0 & 14.83785 & 18.72395 \\ 14.83785 & 0.0 & 59.8323 \\ 18.72395 & 59.8323 & 0.0 \end{pmatrix} \end{array}$$

At this stage, the smallest distance between pairs of clusters is $d((234),1) = 14.83785$. Therefore, we merge clusters (234) and 1 to get a new cluster (1234). We now have only two distinct clusters (1234) and 5 and the distance between them is:

$d((1234),5) = \frac{1}{2}(d(1,5) + d((234),5)) = \frac{1}{2}(59.8323 + 18.72395) = 39.278125$

And the final pair-wise distance matrix becomes:

$$\begin{array}{c} \\ (1234) \\ 5 \end{array} \begin{array}{c} \begin{array}{cc} (1234) & 5 \end{array} \\ \begin{pmatrix} 0.0 & 39.278125 \\ 39.278125 & 0.0 \end{pmatrix} \end{array}$$

Thus, the clusters (1234) and 5 are merged to form a single cluster of all five genes (12345) when the distance reaches 39.278125.

Lastly, we summarise the results of applying the UPGMC average linkage clustering algorithm to our data by a dendogram as shown in Figure 4.1.

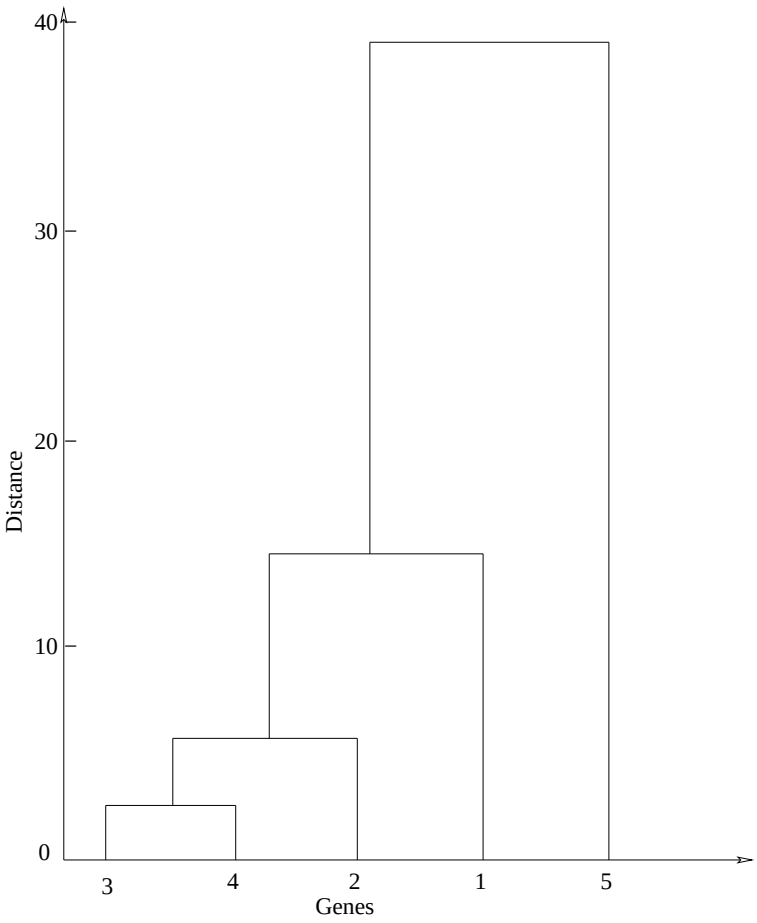


Figure 4.1: Dendogram of UPMC Hierarchical Clustering

Appendix B

Example 2: k-means Clustering

In this example we illustrate k-means clustering performed on the hypothetical data provided in Example 1 in Appendix A. We consider the case $k = 2$.

We start by randomly selecting two gene expression vectors from the five gene expression vectors to serve as the initial centres for the two clusters we want to obtain. The selected gene vectors are *Gene* 1 and *Gene* 3.

Let *Gene* $1 = (-3.06, -2.25, -1.15)$ be the initial centre, $\mathbf{c}_1$, of *Cluster* 1 and
Gene $3 = (-0.17, 0.48, 1.23)$ the initial centre, $\mathbf{c}_2$, of *Cluster* 2. Next, we compute the distances from each gene expression vector to each of the cluster centres using the Euclidean distance and we obtain the following results:

	Gene 1	*Gene* 2	*Gene* 3	*Gene* 4	*Gene* 5
Distance from $\mathbf{c}_1$	0	6.3468	21.4694	25.1884	59.8323
Distance from $\mathbf{c}_2$	21.4694	4.6986	0	2.6018	9.7293

We then proceed by assigning each gene to a cluster whose centre is closest to it. So, this step yields the following clusters:

Cluster 1	**Cluster 2**
Gene 1	*Gene* 2, *Gene* 3, *Gene* 4, *Gene* 5

Next, we calculate the average vector of the gene expression vectors in each cluster and make it the new centre for that cluster. For *Cluster* 1, the centre remains unchanged since there is only one gene expression vector. For *Cluster* 2 the average of the gene expression vectors is given by:

$\frac{1}{4}(Gene\,2 + Gene\,3 + Gene\,4 + Gene\,5)$
$= \frac{1}{4}((-1.36, -0.67, -0.17) + (-0.17, 0.48, 1.23) + (1.16, -0.27, 0.71) + (2.09, 2.12, 2.62))$

$= (0.43, 0.415, 1.0975)$.

So, the new centre for *Cluster* 2 is $\mathbf{c}_2 = (0.43, 0.415, 1.0975)$.

We again calculate the distances from each gene expression vector to each of the updated cluster centres and obtain:

	Gene 1	*Gene* 2	*Gene* 3	*Gene* 4	*Gene* 5
Distance from $\mathbf{c}_1$	0	6.3468	21.4694	25.1884	59.8323
Distance from $\mathbf{c}_2$	24.33358125	5.98788125	0.38178125	1.15228125	7.98063125

We then reassign each gene to the cluster whose centre is closest to it. The updated clusters now become:

Cluster 1	**Cluster 2**
Gene 1	*Gene* 2, *Gene* 3, *Gene* 4, *Gene* 5

Since the composition of the clusters has not changed, we terminate the iteration and the clusters shown in the Table above are the final two clusters for the genes in our dataset, generated by the k-means clustering technique for $k = 2$.

Acknowledgements

I would like to thank my supervisor, Professor Vladimir Bajic and my co-supervisor, Dr Oliver Hofmann for their insightful guidance, direction and assistance.

I also wish to thank Anahita New, Dr Mike Pickles, and Ilhem Benzaoui for reviewing my work and for their helpful suggestions throughout the process of writing this essay.

My gratitude also goes to the management and staff of South African National Bioinformatics Institute (SANBI) at University of the Western Cape for their invaluable assistance. In particular, Peter Van Heusden for helping with transport from AIMS to SANBI throughout the entire essay phase; and Mario, Dale, Cameron, and Oliver for their meaningful contributions.

I also wish to extend my thanks to my student colleagues at AIMS who helped me in many ways throughout my studies, especially my *"Business Associate"*, Evidence, for his invaluable support, advices and encouragement.

Special thanks should go to my former lecturers in the Department of Mathematics at Mzuzu University, Malawi, Dr John A. Ryan, Mr Douglas Raphael Madise, Mr Arts George Luwanda, and, more importantly, Mr Elias Rabson Offen, for providing me a firm foundation in Mathematics and Statistics and supporting my application for the AIMS course.

Finally, words alone can not express the thanks I owe to the management of AIMS, in particular, Professor Fritz Hahne and Professor Neil Turok for providing me the opportunity to undertake this unique and prestigious postgraguate course. I am also indebted to the 2005/2006 AIMS lecturers and tutors who all introduced me to many of the most exciting areas of modern science.

May peace, the mercy of God and His Blessings be upon you.

Bibliography

[1] J.K. Peeters & P.J. Van der Spek, "Growing Applications and Advancements in Microarray Technology and Analysis Tools", *Cell Biochemistry and Biophysics 43*, pp. 149–166, 2005.

[2] A.W. Liew, H. Yan & M.Yang , "Pattern Recognition Techniques for the Emerging Field of Bioinformatics: A Review", *Pattern Recognition 38*, pp. 2055–2073, 2005.

[3] W.P. Kuo, E. Kim, J. Trimarchi, T. Jenssen, S.A. Vinterbo & L. Ohno-Machado "A Primer on Gene Expression and Microarray for Machine Learning Researchers", *Journal of Biomedical Informatics 37*, pp. 293–303, 2004.

[4] http://www.nicerweb.com/doc/class/bio1151/Locked/media/ch06/06_09aAnimalCell.jpg.

[5] http://genetics.gsk.com/graphics/dna-big.gif.

[6] E. Russo & D. Cove, "Genetic Engineering: Dreams and Nightmares", *Oxford University Press*, 1998.

[7] http://www.phschool.com/science/biology_place/biocoach/images/transcription/eunolcol.gif.

[8] S. Escherich & T.J. Yeatman, "DNA Microarray and Data Analysis: An Overview", *Surgery 136*, pp. 500–503, 2004.

[9] M.H. Asyali, D. Colak, O.Demirkaya & M.S. Inan, "Gene Expression Profile: A Review", *Current Bioinformatics 1*, pp. 55–73, 2006.

[10] A. Butte, "The Use and Analysis of Microarray Data", *Nature Review Drug Discovery 1* , pp. 951–960, 2002.

[11] http://phys.chem.ntnu.no/∼bka/images/MicroArrays.jpg.

[12] A. Riva, A. Carpentier, B. Torresani & A. Henaut, "Comments on Selected Fundamental Aspects of Microarray Analysis ", *Computational Biology and Chemistry 29* , pp. 319–336, 2005.

[13] Y.F.Leung & D. Cavalieri, "Fundamentals of cDNA Microarray Data Analysis" *Trends in Genetics 19*, pp. 649–659, 2003.

[14] M. Reimers, "Statistical Analysis of Microarray Data", *Addiction Biology 10*, pp. 23–35, 2005.

[15] J.M. Vaquerizas, L. Conde, P. Yankilevich, A. Cabezon, P. Minguez, R. Diaz-Uriarte, F.Al-Shahrour, J. Herrero & J. Dopazo, "GEPAS, An Experiment-oriented Pipeline for the Analysis of Microarray Gene Expression Data" *Nucleic Acids Research 33, Web Server Issue*, pp. W616–W620, 2005.

[16] J. Herrero, J.M. Vaquerizas, F.Al-Shahrour, L. Conde, A. Mateos, J. Santoyo, R. Diaz-Uriarte & J. Dopazo, " New Challenges in Gene Expression Data Analysis and the Extended GEPAS" *Nucleic Acids Research 32, Web Server Issue*, pp. W485–W491, 2004.

[17] J. Herrero, F.Al-Shahrour, R. Diaz-Uriarte, A. Mateos, J.M. Vaquerizas, J. Santoyo & J. Dopazo, "GEPAS: A Web-based Resource for Microarray Gene Expression Data Analysis" *Nucleic Acids Research 31, No.13*, pp. 3461–3467, 2004.

[18] M. Kapushesky, P. Kemmeren, A.C. Cullane, S. Durinck, J. Ihmels, C. Korner, M. Kull, A. Torrente, U. Sarkans, J. Vilo & A. Brazma, "Expression Profiler: Next Generation -An Online Platform for Analysis of Microarray Data" *Nucleic Acids Research 32, Web Server Issue*, pp. W465–W470, 2004.

[19] C. Romualdi, N. Vitulo, M.D. Favero & G. Lanfranchi, "MIDAW: A Web Tool for Statistical Analysis of Microarray Data" *Nucleic Acids Research 33, Web Server Issue* , pp. W644–W649, 2005.

[20] W. Shannon, R. Culverhouse & J. Duncane, "Analyzing Microarray Data Using Cluster Analysis : Review", *Pharmacogenomics 4(1)*, pp. 41–52 2003.

[21] J. Quackenbush, "Computational Analysis of Microarray Data", *Nature Review Genetics 2*, pp. 418–427, 2001.

[22] B.D. Ripley, "Pattern Recognition and Neural Networks", *Cambridge University Press* , 1996.

[23] T. Hastie, R. Tibshirani & J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction ", *Springer Science & Bussiness Media, Inc.* , 2001.

[24] T. Mary-Huard, F. Picard, & S. Robin, "Introduction To Statistical Methods for Microarray Data Analysis", *Institut National Agronomique Paris-Grignon, http://www.inapg.inra.fr/ens_rech/math/Assets/bib/MPR04.pdf*, 2004.

[25] M.S. Aldenderfer & R.K. Blashfield, "Cluster Analysis", *SAGE Publications, Inc, California*, 1984.

[26] H.B. Burke, "Discovering Patterns in Microarray Data", *Molecular Diagnosis 5*, pp. 349–356, 2000.

[27] D.B. Allison, X. Cui, G.P. Page & M. Sabripour, "Microarray Data Analysis: From Disarray to Consolidation and Consensus", *Nature Review Genetics*, pp. 55–65 2006.

[28] A. Sturn, "Cluster Analysis for Large Scale Gene Expression Studies", *Master Thesis, Institute for Biomedical Engineering, Graz University of Technology*, 2000.

[29] K. Aas. "Microarray Data Mining: A Survey", *Norwergian Computing Center*, http://www.nr.no/files/samba/smbi/microarraysurvey.pdf.

[30] J. Herrero, A. Valencia and J. Dopazo, "A Hierarchical Unsupervised Growing Neural Network for Clustering Gene Expression Patterns", *Bioinformatics 17(2)*, pp.126–136 2001.

[31] http://linus.nci.nih.gov/∼ brb/DataArchive.html.