CLUSTERING OF TIME-COURSE GENE EXPRESSION DATA WITH DISSIMILAR REPLICATES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OZAN ÇINAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
STATISTICS

JUNE 2013

Approval of the thesis:

CLUSTERING OF TIME-COURSE GENE EXPRESSION DATA WITH DISSIMILAR REPLICATES

submitted by **OZAN ÇINAR** in partial fulfillment of the requirements for the degree of **Master of Science in Statistics Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İnci Batmaz _____
Head of Department, **Statistics**

Assoc. Prof. Dr. Özlem İlk _____
Supervisor, **Statistics Department, METU**

Assist. Prof. Dr. Cem İyigün _____
Co-Supervisor, **Industrial Engineering Department, METU**

Examining Committee Members:

Assist. Prof. Dr. Yeşim Aydın Son _____
Health Informatics Department, METU

Assoc. Prof. Dr. Özlem İlk _____
Statistics Department, METU

Assist. Prof. Dr. Cem İyigün _____
Industrial Engineering Department, METU

Assist. Prof. Dr. Zeynep Kalaylıoğlu _____
Statistics Department, METU

Assist. Prof. Dr. Ceylan Yozgatlıgil _____
Statistics Department, METU

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Ozan Çınar

Signature :

ABSTRACT

CLUSTERING OF TIME-COURSE GENE EXPRESSION DATA WITH DISSIMLAR REPLICATIONS

Çınar, Ozan
M.Sc., Department of Statistics
Supervisor: Assoc. Prof. Dr. Özlem İlk
Co-Supervisor: Assist. Prof. Dr. Cem İyigün
June 2013, 118 pages

Clustering the genes with respect to their profile similarity leads to important results in bioinformatics. There are numerous model-based methods to cluster time-series. However, those methods may not be applicable to microarray gene expression data, since they provide short time-series which are not long enough for modeling. Moreover, distance measures used in clustering methods consider the dissimilarities based on only one characteristic and ignore the time-dependencies. Furthermore, genes may show differences among the replications which carry important information. Detecting interesting genes might involve heavy computational burden. In this study, a clustering method is proposed where every gene is accepted as a short time-series with several replications. The distance between the short time-series of replications is measured with the information coming from both the Euclidean distance and the slope distance. The numerical experiments show that the proposed approach can find the clusters very fast with a low percentage of misclassification. Several tests show that the method is also successive in detecting the genes with dissimilar replicates or constant shapes. Finally, different approaches are proposed for determining the number of clusters in a given data set. Simulation studies show that these methods are helpful to detect the number of clusters when it is not known a priori.

Keywords: Microarray Data; Short Time-Series; Clustering; Replication; Cluster Validity

ÖZ

FARKLI TEKRARLI ZAMAN AKIŞLI GEN İFADE VERİLERİNİN KÜMELENMESİ

Çınar, Ozan
Yüksek Lisans, İstatistik Bölümü
Tez Yöneticisi: Doç. Dr. Özlem İlk
Ortak Tez Yöneticisi: Yrd. Doç. Dr. Cem İyigün
Haziran 2013, 118 sayfa

Genlerin zaman serisi portrelerinin benzerliklerine göre kümelenmesi biyoenformatik alanında önemli sonuçlara ulaştırmaktadır. Zaman serilerini kümelemek için geliştirilmiş modellemeye dayalı bir çok yöntem bulunmaktadır. Fakat, mikrodizin gen ifade verileri modellemeye yetecek kadar uzun zaman serileri sağlamadığından bu yöntemler elde edilen verilere uygun olmamaktadır. Dahası, kümeleme yöntemlerinde kullanılan uzaklık ölçümleri uzaklığı tek bir nitelikle belirtmekte ve zaman bağımlılığını gözardı etmektedir. Ayrıca, genler farklı tekrarlarda farklı portreler göstererek önemli bilgiler sunabilir. İlginç genleri belirlemek oldukça ağır bilgisayar işlemleri gerektirebilir. Bu çalışmada, her bir geni bir çok tekrardan oluşan kısa zaman serileri olarak kabul eden bir kümeleme yöntemi sunulmuştur. Tekrarlı kısa zaman serileri arasındaki uzaklıklar hem Öklit hem de eğim farklılıklarından gelen bilgilerle ölçülmektedir. Sunulan yaklaşımın portreleri oldukça kısa zamanda ve küçük hatalarla bulduğu sayısal örneklerle gösterilmiştir. Ayrıca, bir çok test yöntemin, farklı tekrarlı veya sabit portreli genleri de ayırt edilebildiğini göstermiştir. Son olarak, küme sayısını tespit edebilmek için farklı yaklaşımlar sunulmuştur. Simulasyon çalışmaları önsel olarak bilinmediğinde küme sayısını bulmada bu yöntemlerin faydalı olduğunu göstermiştir.

Anahtar Kelimeler: Mikrodizin verileri; Kısa zaman serileri; Kümeleme; Tekrar; Kümeleme geçerliliği.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor and co-advisor, Assoc. Prof. Dr. Özlem İlk and Assist. Prof. Dr. Cem İyigün for everything they did throughout this study. Their diligence and great vision brought this study into existence; and their patience, charity and grace kept me going ahead. I can never find enough words to express my gratitudes to them and how happy I am to had the chance to work with them.

I would like to thank to every faculty and research assistant in the Department of Statistics at METU who have spent respectable efforts to grow me up as a statistician.

Furthermore, I give special gratitudes to precious faculties, Assoc. Prof. Dr. Tolga Can, Prof. Dr. Ayhan Sol, Assoc. Prof. Dr. Çağdaş Devrim Son and Assist. Prof. Dr. Yeşim Aydın Son, who introduced me into new horizons.

I also would like acknowledge my dear fellows, Didem Egemen and Saygın Karagülle, with my best wishes and hopes to be together in the future.

My warm and sincere thanks to my best friends, Metin Altıparmak, Cihan Çetindağ and Mehmet Yıldırım for being always with me.

Last but not the least, I would like to express my appreciations from the deepest place of my heart to my family, İpek, Yasemen and Atila Çınar. This thesis is the product of neverending support and love of Çınar and all relative families.

Finally, I would like to dedicate all my efforts that I have spent throughout this study to the people who defends *"nature"* with my wishes to be forgiven for could not being with them physically.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Discovery of DNA and genetics improved the abilities of the mankind to understand the livings and their environments beyond the imagination. With the help of such abilities, studies have been hold to examine the evolution and the developments of the organisms. As the structures of the bodies are understood with those studies, human beings have been able to resolve their curiosities and create solutions to the problems. Studies on the human genetics and their relationship with the bacterial organisms, for example, helped to reveal the symptoms and treatments of many diseases.

1.1. Microarrays and Analysis of Microarrays

Advanced studies have become necessary in genetics in order to find more effective solutions for the problems. Improvements in technology played an important role to hold the advanced studies. Many instruments have been designed to examine the biological situations in a more detailed way. One of those instruments has been coined in the mid-nineties and called as DNA arrays, a.k.a. microarrays. Microarrays provide the ability to scope the behaviors of thousands of genes in a metabolism with a single experiment, which is a huge contribution to science. Baldi and Hatfield (2002) stated, on this manner, that microarrays had a similar impact in science as the microscopes have had in the last centuries.

Microarrays display the activity level of each gene in an individual. The experiments can be hold at different conditions such as in the presence of a disease in interest. Moreover, a comparison can be made to detect the genes which change their activity levels throughout different conditions to find the effective genes in the formation of the disease in interest. With the help of such studies, the target genes for a treatment can be found to make that treatment more effective and to reduce its side effects. Besides, the relationships and dependencies between the genes can be discovered by examining their activity levels. Those discoveries would be helpful to build the biological pathways of the organisms. In respect to all these studies, the analysis of microarray experiment products may play important roles in finding solutions to a variety of issues. However, microarrays provide data sets with tens of thousands of rows which are challenging to analyze. Those big data sets create difficulties in different steps of an analysis such as interpretation and computation.
Furthermore, the analysis of the results of microarray experiments got more challenging with the introduction of time-series studies into microarrays. In such studies, the activity levels of the genes are followed through time and a time-series of expression levels is

obtained for every gene. Bar-Joseph (2004) cited that collecting the information from the genes by following them in time provides more information than the stable expression levels especially for determining all genes expressed in a condition and catching the interactions between the genes. Those time-series gene expression studies help to discover the developments of biological processes such as cancer, to detect the genes which take role in an infection or a disease and to see the interactions between the genes. A methodical approach for the analysis would be helpful in the hard-challenging nature of the time-series. In this requirement, Bar-Joseph (2004) put forward a systematic approach to gather proper information from time course microarray gene expression levels. According to this approach the process of a time course microarray analysis can be divided into four parts which can be seen in Figure 1.1.

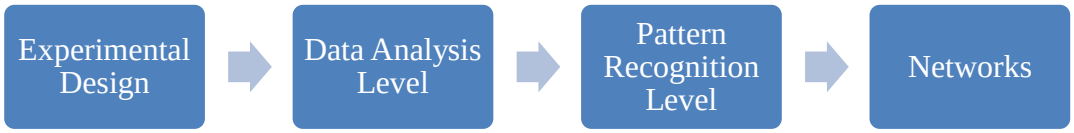


Figure 1.1. Flowchart of microarray analysis

In the first step, the experimenter sets up the design of the experiment. The features such as the number of replicates, conditions and the number of arrays in the experiment are decided in this step. Further, the number of time points and the time lengths between those time points are determined here. Those selections are very important in a time-series studies since they determine the number of observations in time-series and sample rate of the measurements. Also, synchronization of the genes is an important problem on cell cycle experiments. All the genes in the experiment should be set to the same phase of the cell cycle at the beginning of the experiment by synchronization. The second step in the flowchart includes the techniques to make the data set collected from the experiments more practical. One drawback of the microarrays is that they may provide a very noisy data. Therefore, in this step, the experimenter works on the individual genes to reduce the noise level of the signals. Normalization of the genes is also held in this step. Last but not least, the significances of the reactions of the genes are studied in this part. Most of the genes do not show a significant reaction in their expression levels throughout the conditions or time in a microarray experiment. However, the genes which change their activity levels with respect to the condition changes are examined in order to get important results. Thus, the uninteresting genes are filtered in this step in order to lower the complexity of the analysis. The third step considers grouping the genes with similar profiles together. After the separation of the genes, the analyst has the chance to work on small number of groups including similar genes instead of all the genes individually. For this grouping purpose, clustering methods are widely used in the literature. Grouping the genes with similar profiles through time may help to see the genes with similar functions in the organism. Hence, the clustered display of the genes also assists to interpret the function of an unknown gene and

identify the genes specific for a disease. In the fourth step, interactions and dependencies between the genes are investigated. With the help of these investigations, scientists can build the descriptive and predictive models to examine the biological pathways of different organisms. These biological networks aid several cases such as determining the target genes for a treatment or determining the side effects of the treatments. To conclude, these four steps are the main divisions of a microarray analysis procedure. Furthermore, each step serves as a pre-analysis for the further steps, eventually.

1.2. Problem Definition

This thesis looks into the third step of the microarray analysis, the pattern recognition. A clustering algorithm is presented throughout the thesis. Clustering similar genes together has been a useful method to obtain biologically meaningful results from a gene expression data set. Several well-known clustering methods such as hierarchical and k-means clustering have been applied for achieving that goal. Each of those clustering algorithms basically groups the genes with respect to the similarities between the genes by defining the dissimilarities between them with a distance metric such as Euclidean distance on the activity levels of the genes.

However, the occurrence of time-series in microarrays brought several challenges to the clustering studies. The first challenge arose with the nature of the time-series. In such studies, it is known that every observation on a time-series is dependent on the previous measurements. As a result of this, a clustering algorithm should take this dependence into consideration. The default distance metrics, on the contrary, use each measurement separately in a time-series. In consequence, they accept every measurement independent of each other. Moreover, by accepting each measurement separate then the others, the clustering methods ignore not just the time dependency but they also neglect the time lengths between those time points. Changes in time lengths in different time intervals are very occasional in time course microarray experiments. Therefore, considering the time lengths is an important aim for a clustering algorithm, too. Accordingly, the distance measures used in those clustering algorithms should be modified to be aware of the time dependencies and time lengths between the intervals of a time-series.

In order to solve problems about time-series, other methodologies have been proposed by many authors. Some of these methodologies were based on clustering the genes by using mathematical and statistical models. There are, nevertheless, several drawbacks for these model based clustering algorithms. The main drawback for them is their dependencies on the number of time points. In order to build model on the time-series, these methods need a sufficient number of time-points. Ernst et al. (2005) showed that more than 80 % of the microarray experiments had used less than or equal to 8 time points. The “short time-series” can not provide sufficient data for the model based clustering algorithms. Based on this fact, the model based methods may not be useful for clustering the microarray short time-series. Therefore, a clustering algorithm would be more useful and appropriate if it does not have any limitations on the number of time points.

The third challenge about the clustering studies on microarrays emerges as a result of the use of replications. Usage of replications is a technique that is commonly applied in microarray experiments. Different types of replication techniques might be used in microarrays, such as technical and biological replications. For technical replications, the same experimental unit is used for each replication while a different experimental unit is used for each replication in biological replications. Most of the time, the same number of replications is held for each gene in an experiment. The most important problem about the replications is that some genes may show variations among their replications. Such variations may carry valuable information especially when biological replications were chosen for the experiment because of that the variations might be the result of a difference in the unique biological pathway to that experimental unit. Thus, being aware of such genes may provide a very useful ability to examine such experimental units. On this purpose, a clustering algorithm would be very beneficial if it can detect the genes which show variations among their replicates.

Another challenge about the clustering studies is determining the number of clusters. In a real data set, the number of different patterns is unknown as prior information. This prevents the user from being certain about the correct number of clusters. Choosing a small number of clusters may result in clusters with more than one significantly different pattern within each cluster which make the user unable to see all profiles. On the other hand, selecting a high number of clusters can end up with more than one cluster with a similar pattern with too much detailed differences. Both cases would make the interpretations harder. Therefore, offering possible number of clusters without any prior information is a great advantage for a clustering algorithm. With the optimum number of clusters the user would obtain as many clusters as to show the significant patterns of the genes but not too many clusters to avoid the redundancy. Cluster validation techniques can be used to propose methodologies to detect the number of clusters when it is unknown.

Finally, the constant genes will be studied in this thesis, even though it is not one of the aims of the third step in the microarray analysis procedure. As mentioned before, most of the genes in a microarray experiment do not show a reaction through time and keep their baseline levels. The genes which show a reaction and change their activity levels during the experiments are yearned to be in the analysis, though. Based on this fact, the genes which do not show a significant change in their activity levels throughout the experiment are filtered out in order to reduce the complexity of the analysis and computational burden. However, discovering constant genes is also possible by assigning them into specific clusters within the clustering studies.

In this thesis, an algorithm is proposed to attack these problems. The remainder of this thesis is as follows. The literature review is held about the clustering methods on microarrays in the second chapter. The main methodologies and background information for time course microarray gene expression experiments are presented in the next chapter. Next, the approach proposed in this thesis is put forward in Chapter 4 and it is followed with Chapter 5 where numerical examples on several simulated and a real data set are demonstrated. Finally, the thesis is concluded with discussions and future works in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

The behaviors of genes are identified with their expression levels. Specific profiles that emerge on the expression levels of genes might be the reasons for the developments of conditions such as diseases. Microarray gene expression experiments give the simultaneous expression levels of tens of thousands of genes at a specific condition as products. Therefore, examinations on the expression levels of the genes and their profiles through time may lead to crucial knowledge on such phenomena. With the help of these analyses, the effective genes on the generation of a specific condition can be detected.

One way to find those genes is to comparing the expression levels or profiles of genes with condition in interest against those of the genes at a control case. This comparison would help to find the target genes in a treatment. However, since there are too many genes used in those experiments, working on all of those genes individually is very challenging. Several methods have been proposed to lower the complexity of such data sets and clustering is one of those methods. With the clustered form of these data sets, the analyst can work on fairly small number of groups of genes compared to the number of all individual genes. Moreover, cluster analyses help to find the genes which show similar responses through time which, in turn, may be functionally related with disease. Furthermore, examining the different profiles in separate clusters may provide valuable knowledge about the relationships between the profiles. As a result of the related genes, the biological pathways of the organisms can be resolved more accurately. The clusters of profiles may also be helpful to assign presumed functions to novel genes whose functions are unknown since the genes with similar profiles are more likely to be functionally related. Therefore, classifying the genes with respect to the similarities in their profiles is important in biological analysis.

Clustering have been applied and studied on microarrays over several years. The aim of these clustering studies was to catch the fundamental patterns in expression levels of genes in a data set (Tamayo et al., 1999). One of the earliest clustering studies on gene expression data set has been hold by Khan et al. (1998). By using hierarchical clustering method on the stable expression levels, their algorithm detected the genes which are effective on the development of alveolar cancer. This study was a healthful example to show the importance of clustering of genes with respect to their gene expression profiles. Moreover, its results can be used as prior knowledge for deciding if an unknown gene might be effective on alveolar cancer since it defines the effective profiles.

Such studies showed the importance of microarrays and increased the tenancy of microarrays in genomic researches. As the usage of microarrays increases, new data collection methods have also been introduced. A milestone in microarray studies was starting to collect the gene expression levels of genes through time. Two of the pioneer studies were held by Cho et al. (1998) and Spellman et al. (1998). In the former study, it was mentioned that the unexpected biological events may be the consequences of the physiological changes at the genes during the cell cycle periods. Both studies observed the expression levels of the genes of yeast through cell cycle periods during the mRNA transcription. Those researches introduced the time-series studies into microarrays. Collecting the information of microarrays as time-series made the interpretation of the results more challenging. However, Bar-Joseph (2004) mentioned that collecting time course data set instead of stable expression levels is worthwhile to get significant biological results.

Next, studies on the clustering of these time-series gene expression levels started. There were several very well-known clustering studies in the literature at that time, such as hierarchical and k-means clustering. Those clustering methods have been applied to time course gene expression data sets. Eisen et al. (1998) used a hierarchical clustering approach to cluster the time course gene expression level obtained from the genes of yeasts. In order to define the similarity between the time-series of genes, they used a metric similar to the Pearson correlation metric which defines the distance with respect to the correlation between the measurements through time. With this approach they clustered the genes which show similar patterns through time. However, this technique could only catch their similarity with respect to their shapes. Therefore, it ended up with the clusters which include genes with similar patterns at different magnitude levels. In another study (Tamayo et al., 1999), the authors used Self–Organizing Maps (SOM) to see the groups of profiles in a time course gene expression data set. The study displayed a practice of SOM on time-series by using only Euclidean distance metric. The algorithm was eventually tested on a real data set of which results were known from a previous study and was shown to be successful on dividing the genes into meaningful clusters.

These methods, however, may not be suitable for clustering the time-series data sets. As mentioned before, all clustering methods need a distance metric in order to define the similar objects. Most of the metrics catch the similarity based on a single characteristic. For example, the study of Eisen et al. (1998) defined the similar genes which had a similar shape regardless of their magnitude levels. More importantly, most of the clustering methods use Euclidean distance. This metric finds the difference between the expression levels at each time point and takes their summation as the distance between two time-series. This approach, nevertheless, is not a proper method for defining the dissimilarity between the time-series. That is because, in such studies, it is known that each measurement is dependent on the observations at previous time points. Since Euclidean distance, nonetheless, accepts each measurement at each time point separately, it ignores the time dependencies. Furthermore, using only the Euclidean distance fails to handle another feature of the time course gene expression experiments. In time-series gene expression experiments unequal time spaces are used occasionally. In those studies, the time lengths vary at different time

intervals. However, Euclidean distance ignores the time lengths between the time points. As a result, the well-known clustering methods mentioned before could not handle these problems with their default settings.

Later, new methods have been proposed to overcome such problems during the clustering. Most of these methods depend on modeling the time-series statistically. With this approach, several model based clustering methods have been introduced to the literature. One of the first of those approaches is coined by Yeung et al. (2001). The model based clustering studies assumed that the expression levels are generated by a finite mixture probability distribution. By reciting that the Gaussian mixture model was a powerful tool for this aim, the authors assumed that each group with a specific profile in a data set was distributed with a multivariate normal distribution. The geometric aspects of the profiles for these groups were determined by the covariance parameter of the multivariate normal distributions. Therefore, the covariance parameter was parameterized by using eigenvalue decomposition method. This provided five different models which had different features for the parameters of the covariance. An EM algorithm was held to assign the genes to the clusters. After creating clusters by using five different models, the best model was chosen with respect to the results from Bayesian Information Criteria (BIC) for each model. This provided a great advantage to their algorithm which was the ability of deciding the number of clusters with model adequacy checking. Rand indexes (Rand, 1971) of different number of cluster sets were examined to decide the number of clusters. However, it was declared in the study that the number of parameters to be estimated might be inflated especially when the numbers of genes in clusters are small. In such cases, the number of parameters to be estimated might be problematic and may exceed even the number of objects to be clustered. Furthermore, the algorithm depended on the assumption that the expression levels were distributed with a multivariate normal distribution. In the case of the dispersions from multivariate normal distribution, a suitable transformation would be needed. Following that, some of the real data sets were stated which did not hold this assumption. In the case of dispersions from this assumption, the algorithm yielded an overestimation for the number of clusters.

In another study, Ramoni et al. (2002) used a Bayesian autoregressive approach. They assumed that the profiles of a pair of time-series can be similar when they are originated from the same stochastic process. Therefore, they tried to find out a representative stochastic process for each cluster and assign the genes which might be coming from a specific stochastic process to the same cluster. After that, a Bayesian approach was used to find out the number of significant representative stochastic processes which also declares the number of clusters. They compared the likelihood probabilities of the set with m number of clusters with the set of $m+1$ clusters in order to create a new cluster.

Time-series microarray experiments follow the expression levels of the genes in discrete time points. Bar-Joseph et al. (2003) tried to present them as continuous series and applied a clustering algorithm. The main purpose in this study was imputing the missing values in a microarray experiment and clustered status of genes was used with this goal. In order to present the time-series as continuous, polynomial splines were used. It was mentioned that the low degree of polynomials were better for creating smooth lines and did not have the

over fitting problem. Thus, cubic B splines were applied to interpolate the discrete time-series. After fitting a spline for each gene, imputing the missing expression values was aimed. However, the authors also wanted to use background information while imputing the missing cases in order to make the application more robust. Hence, other functionally similar genes were used to guess the missing expression values of a gene. However, those functionally similar genes may be unknown as prior information and clustering took into part in such cases to define the functionally similar genes. The clustering algorithm in this work required the number of clusters given in advance. Then, the algorithm worked iteratively with a modified EM algorithm by assigning a random gene to a cluster and calculating the probability of that gene belonging to that cluster. This process was terminated when the convergence was obtained. Although, the algorithm worked well on a real data set, the authors stated a disadvantage. In order to fit splines to genes, their algorithm needed data sets with large number of time points. Consequently, this disadvantage made their algorithm inappropriate for short time-series gene expression data sets.

In a different study, Luan and Li (2004) tried to identify the genes with periodic expression profiles. It was stated that some genes show periodic biological processes, such as circadian rhythmic regulation or cell cycle regulation, along with the genes which do not show such processes and called as aperiodic. In this study, the periodic genes were aimed to be detected by assigning them into a specific cluster. In order to reach this goal, shape-invariant cubic B-Splines were used to model the expression levels as a function of time. Reference periodic gene profiles were used during the construction of the model. The authors stated that, although, their method was very useful in the noisy nature of microarray gene expression experiments, there were some disadvantages. First, their method needed guide genes; therefore, it could not build models without prior information. Second, all the periodic genes were assumed to have the same shape. This prevented the successful decomposition of the genes with different amplitudes or phases in their expression levels. Finally, their algorithm was stated to be inappropriate for a typical time course microarray experiment which has small number of time points.

As mentioned before, comparing the expression levels of genes in different conditions leads to important biological results. In that purpose, clustering genes through conditions is also a valuable aim for clustering studies. Heard et al. (2005) digested on this purpose and proposed a method to cluster the genes through both profiles and the conditions. Because of the bi-dimensionality of the clustering, this procedure was called as co-clustering. A curve based Bayesian model was offered to cluster the genes by fitting a regression model to each gene. The regression models were fit with a two-stage EM type algorithm. At the first stage, the cross-condition variations were fixed and an optimal clustering set of the expression profiles were found with a Bayesian hierarchical clustering for each condition. At the second stage, their algorithm found the cross-condition covariance which showed the similarity of the profiles of genes through the conditions by using Markov chain Monte Carlo simulations. Finally, the algorithm divided the conditions into groups with the aid of cross-validation scores. Their algorithm had several advantages as a clustering algorithm. First, using a Bayesian approach helped the user to decide some uncertain aspects in the nature of clustering studies such as the number of clusters. Next, their regression approach could

handle the unequal time spaces which are common in microarrays. Moreover, their algorithm could assign the constant genes, which do not show significant changes throughout the experiments, into a separate cluster. This provided reduced complexity while doing further analysis due to smaller number of genes once these constant genes were ignored. On the other hand, however, their algorithm clustered the genes with respect to their shapes, which might end up with the genes which showed similar patterns but in different magnitudes into the same cluster. Moreover, on a trial with a real data set which included 2392 genes, the authors stated that their algorithm ended up with 159 clusters which might create redundancy by showing the profiles in too much detail.

Next, Hakamada et al. (2006) proposed a Mathematical Model Based Clustering (MMBC) method. The aim on this study was to classify the genes with respect to their functions in mRNA transcription. The best-fitted polynomials, which were stated to be one of the simplest methods to fit models to time-series, were adapted. However, it was impossible to extract the information on onset and cessation times which were related with their functions in mRNA transcription. As a result, mathematical kinetic models proposed by Maki et al. (2004) were used since those models included information about onset and cessation times. K-means clustering was applied on the parameters of these mathematical kinetic models with Euclidean distance metric. During the clustering, the parameters of onset and cessation times were inputted. Furthermore, the clustering study was held separately in three ways where both of the parameters or only one of them was used as the input. The algorithm was tested on a real data set from Chu et al. (1998) which was known to include 7 clusters. The success of the algorithm was measured with Silhouette and Rand values. However, their methodology used the prior knowledge on the number of clusters since the algorithm could not decide this. Next, the authors stated that using such a dimension reduction when using at most two parameters of the models may create loss of information problem.

It can be seen that model based methods have serious disadvantages for clustering the time course gene expression profiles. The most important problem with them is their need in sufficient time points in time-series. Ernst et al. (2005) showed that more than 80 percent of microarray time-series experiments included less than 8 time points. This arose the need of non-model based clustering algorithms on time course microarray experiments. In the direction of this need, Ernst et al. (2005) developed a non-model based clustering method, Short Time-series Expression Miner (STEM). This algorithm used the logarithms of the ratios of the expression values to the baseline expression value in time-series for each gene. This led to 0 values at the first time point. Later, this algorithm required the entry of two user defined parameters. First of these parameters, *c*, notified the greatest possible unit change during the intervals. For example, when *c* is inputted as 1, there were three possibilities for the expression change for each interval: one unit increase, one unit decrease or keeping the expression level same. Considering all the combinations of the possible changes on every time interval, the algorithm found all the possible profiles. The second parameter maintained the number of clusters in the data set. Finally, this algorithm assigned each gene to the most similar profile to create the clusters.

Similar to the previous method, Peddada et al. (2005) suggested a different non-model based clustering algorithm, Order Restricted Inference for Ordered Gene Expression (ORIOGEN). This methodology demanded the candidate profiles, such as increasing, cyclical or umbrella shapes, to cluster the profiles of gene expressions. ORIOGEN clustered the time-series of the genes to the candidate profile with the most appropriate magnitude of the correlation coefficient. Using the correlation coefficient, nevertheless, ignored the magnitude difference between the profiles and clustered the genes only with respect to the shapes of the time-series.

Finally, Irigoien et al. (2011) paid attention to another important problem in microarrays. As mentioned in the introduction, replications have a wide usage in microarray experiments. The variations among the replications may reveal the uncertainties in the differences among the biological pathways of the patients. Considering such possible variations, they proposed a methodology to cluster the time course gene expressions. In Irigoien et al. (2011), the dissimilarities between the profiles were defined with procrustes analysis which considers both the magnitude and shape differences. Using this distance metric, the algorithm followed four steps. In the first step, their algorithm detected the constant genes among the data set and filtered them out. In the next step, each gene was examined within their replicates, and the genes with high distances between the replicates were accepted as the genes with differences among the replications. Those genes were also filtered in order to be analyzed later. The third step clustered the genes left, which were not constant and did not have variations among the replications. During the clustering, an agglomerative clustering approach was used with the distance calculated from procrustes analysis. This clustering step also defined the significant profiles among the data set. Finally, in the last step, the genes filtered in the second step were assigned to the most similar profiles emerged in the third step. However, if a gene with differences among the replications was similar to none of the profiles, a new cluster was created for that gene. This algorithm was tested on a simulated and a real data set and evaluated to be successful. However, according to their statements, all the four steps may lead to long computational times for big data sets.

Each study mentioned in this section showed the advantages and disadvantages of both model and non-model based clustering algorithms. In the light of this information, a clustering algorithm is proposed in this thesis with several contributions. This clustering algorithm is aimed to be used on time source gene expression data sets; therefore, it considers the natural features of time-series such as time dependency and varying lengths of time intervals. Furthermore, the algorithm uses distance metrics which considers both the shape and magnitude distances between the time-series. Third, it is able to detect the genes which show variations among their replicates. An important feature of the algorithm is that it is a non-model based algorithm. Thus, the algorithm does not depend on any assumptions or the number of time points in a data set. Finally, some techniques are added to the methodology to detect the number of clusters when it is unknown a priori.

CHAPTER 3

BACKGROUND

Microarray and clustering in time course microarray studies have several features need to be considered. This chapter will give information on these features. In the first section, the constant genes are illustrated. This is followed by the section where the replications are introduced. Further, different components of clustering studies, which are clustering algorithms, distance measures and cluster validation techniques, are explained in the next three sections.

3.1. Differentially Expressed Genes

One of the main aims of microarray studies is to detect the responsible genes in the development of a condition in interest such as cancer by analyzing the reactions of genes to condition changes. Hence, the analysis should be taken on the genes which show reactions. Such genes are mentioned to be as differentially expressed (DE) genes. The basic approach in detecting the responsible genes would be comparing the expression profiles of genes by taking a case/control study. The genes which show a profile at the condition in interest different than its profile at the control study might be considered as the susceptive ones. A hypothesis test can be hold for each gene to see if it shows different profiles with respect to the purpose of finding the responsible genes. Naturally, a multiple test adjustment is needed while holding hypotheses tests to every gene. However, the number of genes in a microarray study is about tens of thousands most of which are non-DE genes. It can be deduced that, the adjustment might be too conservative, since the number of genes is too high. Furthermore, the existence of too many non-DE genes may lead to other problems for the analyst. Such genes may create measurement bias, increase the false discovery rate (FDR) or reduce the sensitivity of the analysis (Calza et al., 2007). Moreover, higher number of constant genes requires longer computational times. In order to overcome these problems, the number of genes is desired to be decreased by filtering the non-DE genes. The genes to be filtered should be chosen with caution, though, in order to prevent the loss of information.

There are several methods to filter those constant genes. The most basic one of them is applying a threshold to the expression levels to the genes. For example, Heard et al. (2005) advised that the genes whose expression level changes are lower than 2-fold change throughout the experiment may be filtered out of the data set. Different thresholds might be chosen to define the DE genes. However, this type of filtering is not accepted as reliable

anymore, since there is not a rational reason while choosing the level of fold change to be used as threshold value. Furthermore, fold change works with means of the replications and it does not take the variability into account. For the methodology proposed in this thesis, filtering is integrated into the clustering algorithm by assigning the constant genes in separate clusters from the DE genes.

3.2. Replications

Replications have a common usage in statistical analyses. The main purpose of using the replications is estimating the variability between the typical experimental units during the data collection. Replications are also considered to be useful in microarray studies. Replications are used in microarrays by reproducing the data collection methods several times on each gene. On the behalf of gathering a balanced data set, usually, the same number of replications is used for each gene. Moreover, in time course microarray experiments, time points are also kept the same in time-series of all replications and genes.

As mentioned earlier, microarrays may produce very noisy data which may create high variances between the replications. Moreover, there might also be problems with the data collection technique that arise from the instruments used to observe the expression levels. Both of these situations lead to measurement errors and replications are useful to recognize them. Furthermore, there are several methods to reduce the bias coming from the noisy nature of microarrays and data collection technique failures. Most of these methods are held in the second step of the microarray analysis flowchart mentioned in the introduction in order to get a data set as unbiased as possible. However, the variations may also carry important biological information. Sometimes, the genes may show reasonable different profiles in different replications depending on the biological history or pathway of the experimental units. Therefore, such differences among the replications may be very beneficial to enlighten the causes or effects of biological contrasts between the experimental units. This leads the use of several replications even though the cost of experiments increases as the number of arrays increases through the replications (Nguyen et al. 2010).

There are three techniques for replications and different emphases on the variability among the replications are given with those replication techniques (Yang and Speed, 2003). The first technique, *Within-Slide Replication,* collects replicates from the same experimental unit and hybridizes them in the same DNA array. This method gives the least variation between the replicates since the same experimental unit is used on the same array. With this approach, nonetheless, the analyst obtains the dependent and unique observations. Thus, the most secure information about the noise level can be derived with this replication technique. *Technical Replication,* which is the second replication technique, again uses the genes of the same experimental unit at each replication. However, another DNA array is used for each replication. Since a different array is used for each replication less dependence is obtained than the first replication technique. In this technique, the only variability between replications comes from the different arrays. Hence, this replication method is very useful to estimate the measurement error. The last replication method is called as *Biological Replicates* and has two different ways to apply, type I and type II. With type I biological

replication, a gene from a different cell or tissue in a single experimental unit is used for each replication. On the other hand, for the type II biological replication, genes from the same cell or tissue are replicated from different experimental units. For both types, different DNA arrays are used for the replications. These techniques provide the least dependent expression levels among the replications. Furthermore, it helps to see the differences between the expression levels of different tissues or experimental units which may lead to important biological results mentioned before. Type II biological replication is commonly used in microarrays. With the help of this approach, the analyst can have the chance to analyze the changes in the profiles of gene expressions from patients with different biological backgrounds.

3. 3. Clustering Algorithms

Clustering is the analysis of dividing subjects into several meaningful groups. It is very useful to understand and analyze high-dimensional data sets. The product of a clustering analysis is the division of the objects into different groups. The objects within a cluster are similar to each other while dissimilar to the objects in other clusters. Therefore, homogeneity within the clusters and heterogeneity between the clusters are desired at the end of a cluster analysis. This results in time course microarray studies with similar pattern genes being grouped in a specific cluster. Thus, each cluster includes genes with a unique profile. Clustered form of the genes is very helpful for the analysts since it reduces the number of objects to be analyzed. Moreover, the clusters may be useful to detect the genes which have similar functions in the biological pathway. That is why the genes with similar profiles can be assumed to have similar functions. Furthermore, by comparing and analyzing different profiles, the dependent genes can be discovered in an organism which would be helpful to build the biological pathways. Finally, with the assumption of genes with similar profiles would have similar functions, the clustering studies would be beneficial to presume the functions of unknown genes.

As it can be seen, clustering studies are very beneficial to use on time course gene expression data sets. There are several well-known and useful clustering algorithms in the literature. K-means, hierarchical clustering and Self-Organizing Maps (SOM) are the most common ones of these algorithms and have been used to cluster the genes in microarray data sets (Baldi and Hatfield, 2002). A review of these methods will be presented in the following subsections.

3. 3. 1. K-means Clustering

K-means algorithm aims to divide the objects into a pre-selected number of clusters. Therefore, the desired number of clusters is inputted by the user before starting the clustering process. Suppose that *k* shows the number of clusters desired. After adjusting *k*, k-means algorithm starts with defining *k* of the individuals in the data set as the initial cluster centers (centroids). Then it works iteratively where the distances from each object to every centroid are measured. With respect to these distances, each object is assigned to the closest centroid. After assigning every object to a cluster, the centroid coordinates are recalculated

as the average of the coordinates of the members in that cluster at the end of the iteration. This procedure is run until none of the objects changes its cluster within an iteration. The resulting centroids present the clusters and the objects become assigned to the cluster with the closest centroid (Tan, 2006). A pseudo algorithm for k-means is given below.

1. Set the number of clusters, *k*.
2. Choose *k* points as the initial centroids randomly

Repeat

3. Assign every point to the closest centroid
4. Recalculate the centroid coordinates with the new members

Until None of the objects change the cluster

K-means is one of the most basic clustering algorithms and has a low complexity. Although it seems a very useful clustering method, it has two important drawbacks. First disadvantage is the need for setting the number of clusters in advance. This constraints the vision of the user on the data set. The second and more important drawback is the hindrance on reproducing the same result. Since k-means starts by setting initial centroid randomly, replicating the same result in different trials is not guaranteed.

3. 3. 2. Hierarchical Clustering

Hierarchical clustering is another widespread clustering algorithm. It aims to build a dendrogram of the objects which shows the dissimilarities between them in a hierarchical way. This hierarchy between the objects can be built in two directions where these directions also give the names of two approaches. In the first approach, the hierarchy is built by starting from the top and goes to the bottom which is called divisive approach (top-down approach). This approach accepts the whole data set as a cluster at the beginning of the algorithm and divides the data set into two most separated divisions. At each step, the most distant set of objects are separated from the others. This procedure continues until every individual object is left alone. A pseudo code for divisive hierarchical clustering is given below.

Group the whole data set in a cluster ($m = 1$)

Repeat

1. Divide the data set into the most separated $m + 1$ clusters
2. Calculate the distances between the clusters
3. Increase m by one unit

Until Every individual object constitutes a cluster

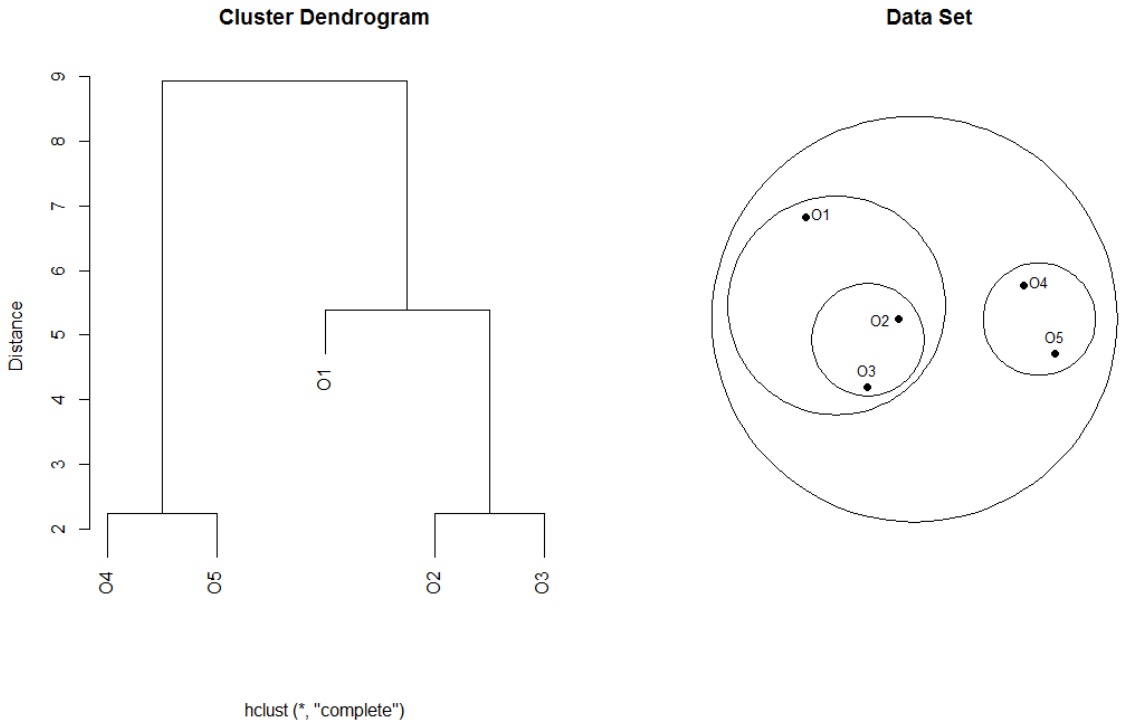
On the other hand, the second and more common approach starts to cluster the objects from the bottom and continues to the top which is known as agglomerative approach (bottom-up approach). Unlike the divisive approach, agglomerative approach takes each individual object as a cluster at the beginning of the clustering and continues to group the most similar objects at each step until all objects are clustered into one. The pseudo code for agglomerative hierarchical clustering is given below.

Compute the distances between each pair of objects in the data set
Repeat
1. Find and bind two closest clusters together
2. Recalculate the distances between the clusters
Until All elements are grouped in one cluster

Both approaches create the dendrogram which gathers all of the objects in a single tree. Branches of this tree show the most similar objects side by side under the same branch of that dendrogram (see Figure 1). When the dendrogram is finished, the user can cut this tree at any level in order to reach the desired number of clusters. Therefore, hierarchical clustering does not need any pre-defined number of cluster to function because the dendrogram tree can show any number of cluster sets.

Figure 3.1 shows an example of hierarchical clustering process on five objects. Five objects are placed in 2-dimensional space in Figure 3.1(b) and notations from *O1* to *O5* present the objects. The dendrogram on these five objects in Figure 3.1.a shows that there are two groups with two most similar objects. The first group contains *O4* and *O5* while the second group contains *O2* and *O3*. Furthermore, the last object left out, *O1*, is closer to the second group than the first group. Thus, *O1* gets bounded to the second group which contains *O2* and *O3*, later. Finally, hierarchical clustering groups all objects in one last and biggest cluster. As explained earlier, the user can cut the dendrogram at any level due to the desired number of clusters. For example, if three clusters are needed for the example set in Figure 3.1.b, the dendrogram in Figure 3.1.a should be cut from the level of 4 for the distance in the y-axis. This would be ended with three clusters where the first cluster contains *O4* and *O5*, the second cluster contains just *O1* and the third cluster contains *O2* and *O3*. As a second example, if two clusters are wanted for this data set, then the dendrogram can be cut from the level of 6 for the distance shown in the y-axis which creates two clusters where *O4* and *O5* are in one cluster and, *O1*, *O2* and *O3* are in the other cluster.

As aforementioned, hierarchical clustering builds the dendrogram of the objects within the data set which shows the closeness of the objects in a hierarchical way by using the dissimilarities between them. This provides a great advantage to hierarchical clustering, that is, the dendrogram can be built without the need of pre-defined number of clusters. Furthermore, hierarchical clustering uses a deterministic approach which provides the reproducible results. Such advantages bring reliability on using hierarchical clustering algorithm for clustering studies.

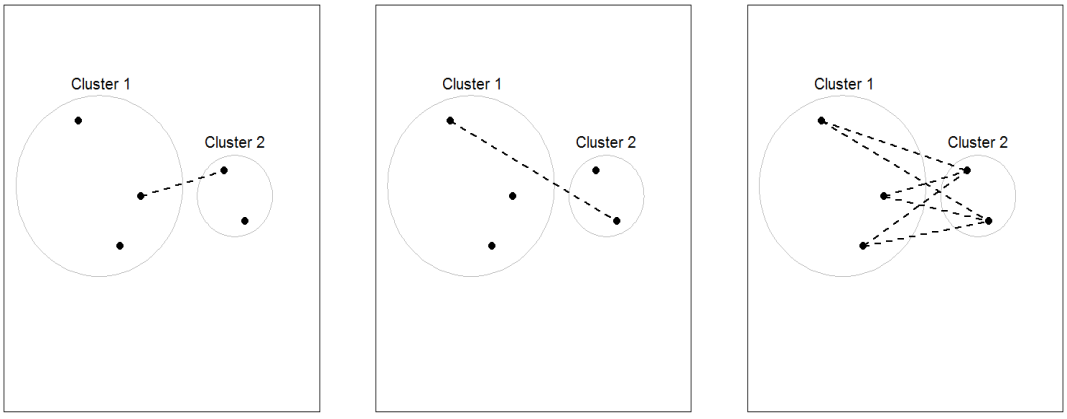


a. Dendrogram b. Two-dimensional presentation

Figure 3.1. A hypothetical data set

As it can be understood from the pseudo codes of two approaches, hierarchical clustering separates or binds the groups of objects during the process of creating the dendrogram. Therefore, the algorithm needs definitions to measure the distances between the groups of objects. These distances are measured with linkage methods. Hierarchical clustering uses four linkage methods mainly to measure the distances between clusters. The first linkage method, *single linkage*, accepts the distance between two clusters as the distance between the two closest objects in these two groups. Next, *complete linkage* takes the farthest two points, contrary to the *single linkage*, and assigns it as the distance between the two clusters. The third method calculates the average of all distances between each pair between the clusters and defines it as the distance between the clusters which is called as *average linkage* (Figure 3.2). These three methods will be defined in the Subsection 3.5.2 in more details.

The last linkage method, *Ward's method*, is different from the other methods since it does not compute the distances between the clusters. With the *Ward's method*, the groups of objects are bounded or separated while keeping the sum of square errors as small as possible. Therefore, this linkage method tries to minimize the within-cluster variance during the building process of dendrogram (Szekely and Rizzo, 2005).



a. b. c.

Figure 3.2. Pictorial representation of a. Single Linkage; b. Complete Linkage; c. Average Linkage

3. 3. 3. Self-Organizing Maps (SOM)

Self-Organizing Map is a dimension reduction method developed by Kohonen (1982). SOM are used to present a data set where the objects more than two characteristics are projected on to single or more commonly 2-dimensional spaces. Since, the objects with similar characteristics are projected on the new space closer to each other, SOM can also be used as a clustering algorithm.

The algorithm of SOM starts with creating a grid on which the multidimensional data is projected. The dimensions of this grid are defined by the user. According to the dimensions given to the algorithm, SOM creates nodes with the number of multiplication of the dimensions. For example, suppose that the user wants to project an n-dimensional data on a 2 dimensional space where the dimensions of the grid are 4 and 6. Therefore, there will be 4 * 6 = 24 nodes in the grid (see Figure 3.3).

At the initial phase of the process, the algorithm assigns n-dimensional characteristics to each node, randomly. After this, the number of iterations that is to be hold throughout the algorithm is set by the user. In an iteration, one observation from the data set is selected randomly. The distances between this object and each node are calculated and the observation hits to the closest node. This means that that observation's coordinates affect and change the centroid of that node. However, the effect of the observation which hits gets smaller as the iterations past. For example, in the first iteration, the centroid of the node is replaced by the coordinates of the hitting object. At the last iteration, nonetheless, the characteristics of the node hit by the randomly selected observation in that iteration are affected very little.

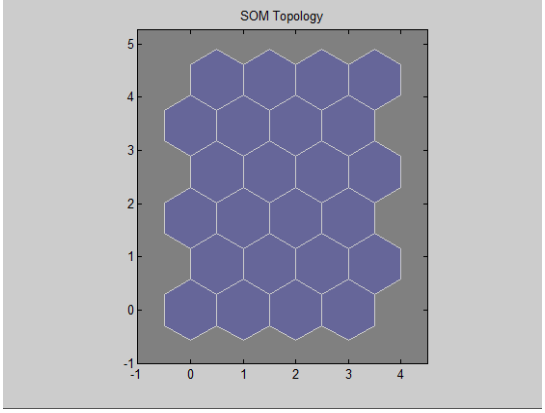


Figure 3.3. Exemplary grid with dimensions 4 and 6

Furthermore, when a node gets hit by an observation, the centroids of the neighbor nodes are also get affected. The closer neighbor nodes to the node which got hit, nevertheless, affected by the hitting observation more than the farther neighbor nodes. The levels of the affections for the neighbor nodes are determined by the Gaussian distribution.

At the end of the process, each node gets a specific centroid. Thus, each node can be accepted as a cluster. In order to obtain the clustered set of the observations in the data set, each observation is assigned to the closest node. A pseudo code for the Self-Organizing Maps is given below.

1. Initialize the centroids of the nodes randomly
2. Set the number of iterations

Repeat

3. Select an object from the data set randomly
4. Assign the object selected in Step 3 to the closest node.
5. Recalculate the centroid of that node and the neighbor nodes.

Until the number of iterations reaches to the pre-defined value.

6. Assign each observation to the node with the closest centroid.

Self-Organizing Maps is a very useful and strong algorithm for clustering. However, the two disadvantages for the k-means clustering also hold for SOM. Firstly, the number of nodes is selected before the algorithm starts. Furthermore, each node obtained at the end of the algorithm may not stand as a specific cluster. It would be very presumptive to obtain several nodes representing very similar characteristics. This would lead to group some of the nodes together which means another clustering analysis is needed. Secondly, randomness again plays an important role in the algorithm while assigning the initial centroids and selecting the observation at each iteration. This may leave the analyst unable to reproduce the results of the clustering.

The last three subsections provided the information on three of the most common non-model clustering algorithms in the literature. One common feature of these clustering algorithms is that they do not need any prior information. That means they do not need reference profiles to cluster the genes. The clustering methods in the literature may be divided into two parts as supervised and unsupervised methods. The supervised clustering methods need prior information or reference objects in order to cluster the objects. The unsupervised methods, on the other hand, hold the clustering process without such information. Since the reference profiles or prior information may not be gathered in microarray studies, the unsupervised clustering method may be accepted as more beneficial on these studies. This supports the use of the three unsupervised clustering algorithms in this section for time course microarray investigations.

Finally, as it can be understood from the information of the clustering methods, every algorithm needs to define the dissimilarities between the objects to be clustered. In order to measure the dissimilarities, several distance metrics have been proposed in the literature. The next subsection will present the most common of these distance metrics.

3. 4. Distance Measures

The distance metrics measure the closeness of the objects based on different characteristics. These metrics are divided into two groups as similarity or dissimilarity (distance) metrics. Similarity metrics measure how similar any pair of individuals are and generally take values between 0 and 1. The value 0 means "no similarity" while 1 means a "complete similarity". On the other hand, dissimilarity metrics measure how far the objects are. Therefore, the dissimilarity metrics can be taken as the distance between pairs. Objects with dissimilarity score close to 0 are considered to be similar. The similarity between the objects gets lower as the dissimilarity score increases (Tan, 2006).

The most well-known distance metric is the Euclidean distance (L_2 norm) which gives the absolute distance between any two points in the space. It uses the following formula:

$$\boldsymbol{Euclidean\ Distance\ (L_2\ norm)}: d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{y}_i)^2} \qquad (\mathbf{3.1})$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are n-dimensional points as $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ and $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n)$. Euclidean distance is a member of a group known as L norm distance. Since Euclidean distance is also known as L_2 norm it takes sum of the squares of the distances between two points under each dimension, and takes the square root of that summation. Another member of that family, for example, is the L_1 norm which is also known as Manhattan distance. As it can be deduced from its name, it takes the distances between two points under each dimension with their absolute values, and sums them up (Eq. 3.2).

$$\textbf{Manhattan Distance } (\boldsymbol{L_1 \text{ norm}}): d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} |\boldsymbol{x}_i - \boldsymbol{y}_i| \qquad (\mathbf{3.2})$$

An important modification can be done on the L_2 distance by leaving the distance without taking the square root. This leads to a different metric known as squared Euclidean distance, as

$$\textbf{Squared Euclidean Distance } (\boldsymbol{L_2 \text{ norm}}): d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{y}_i)^2 \qquad (\mathbf{3.3})$$

With the squared forms of the distances the dissimilarities between the objects increases exponentially as the distance between them increases. Therefore, less emphasis is given to the larger distances (Tan, 2006). The three metrics obtained from the Equations 3.1, 3.2 and 3.3, defines the dissimilarities between the objects based on their absolute differences. In time-series studies, therefore, these metrics will measure the magnitude differences between the observations at follow-up points.

Another metric is known as the cosine-angle distance. It basically measures the cosine value between the two vectors. As the angle between two vectors gets closer to 0, the metric will be closer to 1, which indicates that two vectors get similar to each other. Thus, it can be classified as a similarity metric. The cosine-angle distance between two series can be found as follows:

$$\textbf{Cosine} - \textbf{Angle Distance}: d(\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{y}_i}{\sqrt{\sum_i^n \boldsymbol{x}_i^2} \sqrt{\sum_i^n \boldsymbol{y}_i^2}} \qquad (\mathbf{3.4})$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are two n-dimensional objects. Moreover, Pearson correlation distance measures the correlation between the two objects. It can be measured by using the dot product of two normalized vectors, and gives a result between -1 and 1 (Baldi and Hatfield, 2002). A Pearson correlation measure close to 1 means that the vectors have similar shape, whereas they are directly opposite as the Pearson correlation value gets closer to -1. Pearson correlation value of 0 means that the vectors are independent from each other (Do and Choi, 2007). Eq. 3.5 shows the formula for Pearson correlation distance between two objects.

$$\textbf{Pearson Correlation Distance}:$$
$$d(\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{\sum_{i=1}^{n} (\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{y}_i - \overline{\boldsymbol{y}})}{\sqrt{\frac{\sum_{i=1}^{n} (\boldsymbol{x}_i - \overline{\boldsymbol{x}})^2}{n}} \sqrt{\frac{\sum_{i=1}^{n} (\boldsymbol{y}_i - \overline{\boldsymbol{y}})^2}{n}}} \qquad (\mathbf{3.5})$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are two n-dimensional objects; $\overline{\boldsymbol{x}} = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i}{n}$ and $\overline{\boldsymbol{y}} = \frac{\sum_{i=1}^{n} \boldsymbol{y}_i}{n}$. Finally, another metric was proposed by Möller-Levet et al. (2005) and called as Short Time Series Distance. It was defined to measure the shape dissimilarities between the short time-series. This metric, basically, measures the distance between the slopes of the time-series at each time

interval, and uses their sums over different time points. STS distance can be measured as follows:

Short Time Series Distance:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n-1} \left(\frac{\boldsymbol{x}_{i+1} - \boldsymbol{x}_i}{t_{i+1} - t_i} - \frac{\boldsymbol{y}_{i+1} - \boldsymbol{y}_i}{t_{i+1} - t_i} \right) \quad (\mathbf{3.6})$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are two time-series with n time points; and t_i shows the follow-up time of the i^{th} observation. Contrary to the first three metrics defined in this section, the metrics obtained from Eq 3.4, 3.5, and 3.6 catch the dissimilarities between the objects based on their shapes. In conclusion, each distance metric defines the dissimilarity between the objects by using one of the characteristics which are, generally, the magnitude and slope differences.

3. 5. Cluster Validation

An important challenge about the clustering studies is deciding on the number of clusters. In a clustering study, the number of clusters is not known. In microarrays this problem arises too, since the number of different profiles is not known as prior information. Therefore, a cluster algorithm should also be helpful to decide on the number of clusters. This challenge can be solved by measuring the cluster qualities at the end of the clustering algorithm. A high quality of a set of clusters is obtained when homogeneity within clusters and heterogeneity between clusters are obtained. This means that the within cluster distances should be small while the between cluster distances should be large. When both of these are optimized, one could reach the number of clusters as much as needed to show different profiles without the redundancies. There are mainly three different validation techniques in the literature which are *Silhouette method*, *Dunn's based index* and *Davies–Bouldin index* (Bolshakova and Azuaje, 2003). These three methods are reviewed in the following subsections.

3.5.1. Silhouette Index

Silhouette index gives a quality measure to each observation after the clustering. Suppose that, a set of observations are grouped in u (u = 1, 2, …, U) clusters and C_u show the uth cluster and i_u shows the ith observation in the uth cluster. A silhouette index for that observation is obtained as:

$$s(i_u) = \frac{b(i_u) - a(i_u)}{\max\{a(i_u), b(i_u)\}} \quad (\mathbf{3.7})$$

where $a(i_u)$ shows the average distance between the ith object in the uth cluster with the other objects in the uth cluster; $b(i_u)$ shows the minimum average distance between the ith observation in the uth cluster to all observations in another cluster C_m (m = 1, 2, …, U; $u \neq m$). With this equation the silhouette index, $s(i_u)$, can only take the values between -1 and 1

for each object. A silhouette index closer to 1 means that object is clustered in a correct one. A silhouette index closer to -1 shows the otherwise.

By using this silhouette index, an average quality can be assigned for each cluster by taking the average of the silhouette indexes of all of the observations in that cluster. Suppose that S_u shows the average silhouette index for cluster u, and it can be found as follows:

$$S_u = \frac{1}{n_u} \sum_{i=1}^{n_u} s(i_u) \qquad (\mathbf{3.8})$$

where, n_u shows the number of observations in the uth cluster.

Moreover, a global silhouette index score can be found for the set of U clusters where U can take values from 1, where all observations are grouped in one cluster, to the number of all observations, where all individual observations are accepted as a specific cluster. A global silhouette score for U number of clusters, GS_U, can be measured as follows:

$$GS_U = \frac{1}{U} \sum_{i}^{U} S_i \qquad (\mathbf{3.9})$$

With this approach, the number of clusters with maximum silhouette score can be chosen as the set of clusters.

3.5.2. Dunn's and Davies-Bouldin Indices

Dunn's and Davies-Bouldin indexes are other two validation measures to show the quality of the cluster sets. Both indexes give a global validation score by considering within and between cluster distances. Aim of these indexes is to find the optimal cluster sets by using the maximum between cluster and the minimum within cluster distances. Suppose that a data set is divided into U clusters and C_u shows the uth cluster. The Dunn's index value for this set of clusters can be found as follows:

$$D(U) = min_{1 \le i \le u} \left\{ min_{1 \le j \le u} \left\{ \frac{B(C_i, C_j)}{max_{1 \le m \le u} W(C_m)} \right\} \right\} \qquad (\mathbf{3.10})$$

where C_i, C_j and C_m show ith, jth and mth clusters, respectively (i ≠ j); $B(C_i, C_j)$ shows the distance between the C_i and C_j; and $W(C_m)$ shows within distance for C_m. By using this formula, Dunn's index finds the ratio of the minimum between cluster distances to the maximum within cluster distance. Therefore, it aims to maximize the between cluster distance while keeping the within cluster distance minimum. Consequently, the cluster sets with larger Dunn's index values should be selected.

The Davies-Bouldin index for a set of *U* clusters, on the other hand, can be found with the following formula:

$$DB(U) = \frac{1}{u}\sum_{i=1}^{u} max_{i \neq j}\left\{\frac{W(C_i) + W(C_j)}{B(C_i, C_j)}\right\} \quad (\mathbf{3.11})$$

where, again, u shows the number of clusters; $W(C_i)$ shows within distance for C_i; and $B(C_i, C_j)$ shows the distance between the C_i and C_j. Unlikely to the Dunn's index, Davies-Bouldin index finds the maximum ratio of within to between cluster distances. Therefore, cluster sets with small number of Davies-Bouldin indexes are preferred to be the optimal set of clusters.

Both of the Dunn's and Davies-Bouldin indexes use two different distance measures as within and between cluster distances. In order to find these distances, there are several ways which depend on the distances between the objects in those clusters. The distance measures defined in Section 3.4. can be used to define the distances between the objects. The following two subsections will give information about the within and between cluster distances.

3.5.2.1. Between cluster distances

Between cluster distances define the separation of different clusters. A well set of clusters should include clusters which are clearly separated from each other. When this separation is obtained, it shows that every cluster represent a unique profile. There are five main distance measures to define the between cluster distances, which are *single linkage*, *complete linkage*, *average linkage*, *centroid linkage* and *average of centroids linkage.*

Single linkage: Single linkage is defined by the minimum distance between the two objects from different clusters. It can be found as follows:

$$B_1(C_i, C_j) = min\left\{d(\boldsymbol{x}, \boldsymbol{y})_{\boldsymbol{x} \in C_i, \boldsymbol{y} \in C_j}\right\} \quad (\mathbf{3.12})$$

where C_i and C_j are ith and jth clusters, respectively; $\boldsymbol{x}$ and $\boldsymbol{y}$ are n-dimensional vector observations from the ith and jth cluster, respectively; and $d(\boldsymbol{x}, \boldsymbol{y})$ defines the distance between observations $\boldsymbol{x}$ and $\boldsymbol{y}$.

Complete linkage: Contrary to the single linkage, complete linkage is defined by the maximum distance between two observations in different clusters and can be found as follows:

$$B_2(C_i, C_j) = max\left\{d(\boldsymbol{x}, \boldsymbol{y})_{\boldsymbol{x} \in C_i, \boldsymbol{y} \in C_j}\right\} \quad (\mathbf{3.13})$$

where C_i and C_j are i[th] and j[th] clusters, respectively; $\boldsymbol{x}$ and $\boldsymbol{y}$ are n-dimensional vector observations from the i[th] and j[th] cluster, respectively; and $d(\boldsymbol{x}, \boldsymbol{y})$ defines the distance between observations $\boldsymbol{x}$ and $\boldsymbol{y}$.

Average linkage: Average linkage calculates the average distance between the pair of objects where the objects are from the different two clusters. Eq 3.14. shows the procedure to calculate the distance between two clusters by using the average linkage method.

$$B_3(C_i, C_j) = \frac{1}{n_i * n_j} \sum_{x=1}^{n_i} \sum_{y=1}^{n_j} d(\boldsymbol{x}, \boldsymbol{y}) \qquad (\mathbf{3.14})$$

where n_i shows the number of objects in the i[th] cluster; n_j shows the number of objects in the j[th] cluster; $\boldsymbol{x}$ and $\boldsymbol{y}$ are n-dimensional vector observations from the i[th] and j[th] cluster, respectively; and $d(\boldsymbol{x}, \boldsymbol{y})$ defines the distance between observations $\boldsymbol{x}$ and $\boldsymbol{y}$.

Centroid linkage: Centroid linkage defines the distance between the centroids of two clusters. Centroid of a cluster can be found as the average of the measurements of the objects in that cluster:

$$\boldsymbol{v}_i = \frac{1}{n_i} \sum_{x=1}^{n_i} \boldsymbol{x} \qquad (\mathbf{3.15})$$

where $\boldsymbol{v}_i$ shows the n-dimensional centroid of the i[th] cluster; n_i shows the number of objects in the i[th] cluster; $\boldsymbol{x}$ shows the n-dimensional vector objects from the i[th] cluster. By using the centroids obtained by Eq. 3.15 the centroid linkage between two clusters can be found as:

$$B_4(C_i, C_j) = d(\boldsymbol{v}_i, \boldsymbol{v}_j) \qquad (\mathbf{3.16})$$

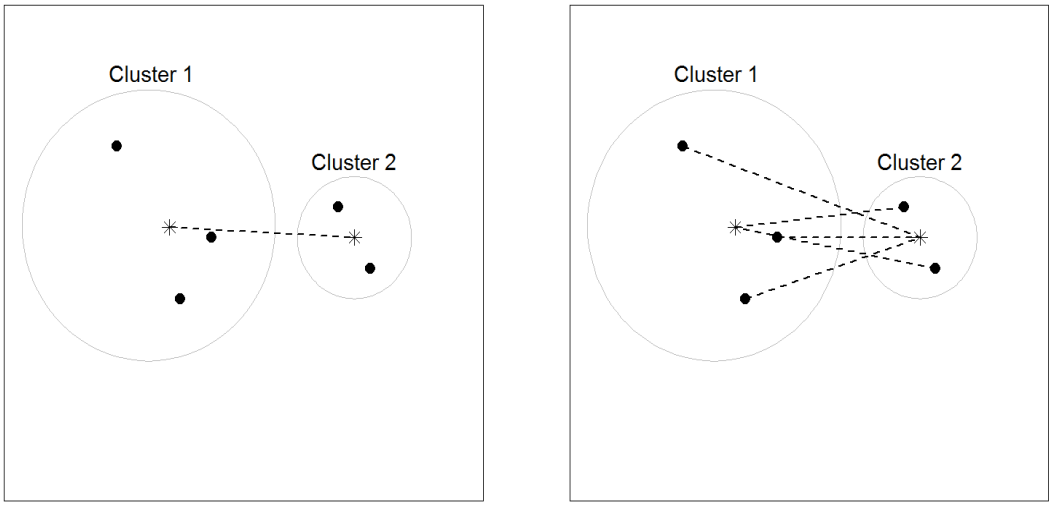
where $d(\boldsymbol{v}_i, \boldsymbol{v}_j)$ shows the distance between the centroids of i[th] and j[th] clusters.

Average to centroids linkage: Similar to the centroid linkage, average to centroids linkage also calculates the distance between two clusters by using their centroids. The centroids of the clusters are again found as in Eq. 3.15. Average to centroids linkage measures the average distance between the centroid of one of the clusters to the all objects in the other cluster, mutually for both clusters (Eq. 3.17).

$$B_5(C_i, C_j) = \frac{1}{n_i + n_j} \left(\sum_{x=1}^{n_i} d(\boldsymbol{x}, \boldsymbol{v}_j) + \sum_{y=1}^{n_j} d(\boldsymbol{y}, \boldsymbol{v}_i) \right) \qquad (\mathbf{3.17})$$

where n_i shows the number of objects in the i[th] cluster; n_j shows the number of objects in the j[th] cluster; $d(\boldsymbol{x}, \boldsymbol{v}_j)$ shows the distance between an object in the i[th] cluster to the j[th] cluster's

centroid; and $d(\boldsymbol{y}, \boldsymbol{v}_i)$ shows the distance between an object in the jth cluster to the ith cluster's centroid.



a. b.

Figure 3.4. Pictorial representation of some linkages a. Centroid linkage; b. Average to centroids linkage

3.5.2.2. Within Cluster Distances

Within cluster distances measure the compactness of the clusters. A good set of clusters should contain compact clusters, that is, each cluster should contain a specific group of objects. In time course microarray studies, this means that each cluster represents a specific pattern through time. The compact clusters can be obtained when the within cluster distances are as small as possible. There are three methods to measure the within cluster distances: *Complete diameter*, *Average diameter* and *Centroid diameter*. The next titles will introduce these three measures.

Complete diameter: Complete diameter finds the farthest objects in the clusters and assigns the distance between them as the within cluster distance for that cluster. It can be found as:

$$W_1(C_i) = max_{\boldsymbol{x},\boldsymbol{y} \in C_i}\{d(\boldsymbol{x},\boldsymbol{y})\} \qquad (\mathbf{3.18})$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are n-dimensional vector objects in the ith cluster; and $d(\boldsymbol{x},\boldsymbol{y})$ shows the distance between the objects $\boldsymbol{x}$ and $\boldsymbol{y}$.

Average diameter: This method calculates the average distances between each pair of objects in a cluster and assign it as the within cluster distance (see Eq. 3.19)

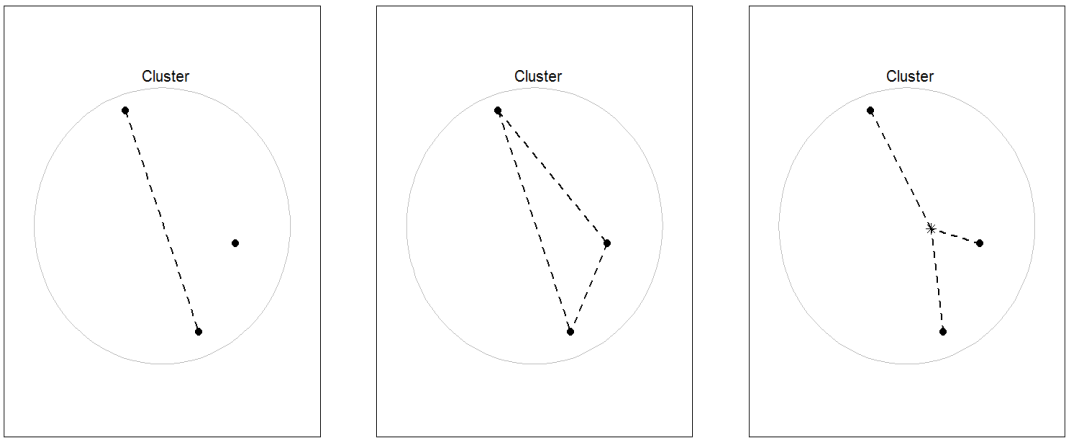
$$W_2(C_i) = \frac{1}{n_i * (n_i - 1)} \sum_{\boldsymbol{x}=1}^{n_i} \sum_{\boldsymbol{y}=1}^{n_i} d(\boldsymbol{x}, \boldsymbol{y}) \qquad (\mathbf{3.19})$$

where n_i shows the number of objects in the cluster i; $\boldsymbol{x}$ and $\boldsymbol{y}$ are n-dimensional vector objects in the cluster i ($\boldsymbol{x} \neq \boldsymbol{y}$); $d(\boldsymbol{x}, \boldsymbol{y})$ shows the distance between the objects $\boldsymbol{x}$ and $\boldsymbol{y}$.

Centroid diameter: Centroid diameter uses the centroids of the clusters. The centroids of the clusters are defined as similar to the centroid and average to centroid linkage methods and as in Eq. 3.15. This measure calculates the twice of the average distances between the objects in a cluster to the centroid of that cluster. Eq. 3.20 shows the formula for the centroid diameter.

$$W_3(C_i) = 2\left(\frac{\sum_{\boldsymbol{x}=1}^{n_i} d(\boldsymbol{x}, \boldsymbol{v}_i)}{n_i}\right) \qquad (\mathbf{3.20})$$

where $\boldsymbol{x}$ is a n-dimensional vector object in cluster i; n_i shows the number of objects in the cluster i; $\boldsymbol{v}_i$ is the centroid of the i^{th} cluster; and $d(\boldsymbol{x}, \boldsymbol{v}_i)$ shows the distance between the object $\boldsymbol{x}$ and the cluster centroid.



a. Complete diameter b. Average diameter c. Centroid diameter

Figure 3.5. Pictorial presentation of within cluster distances

CHAPTER 4

METHODOLOGY

As a conclusion of the problem definition section of the first chapter, clustering on time course gene expression data sets is an important challenge for the analysts. The second chapter gave information on some approaches to solve this problem and it implied that a non-model based clustering methodology would be very beneficial. The next chapter, Background, reviewed the useful tools for clustering studies and an important challenge in these studies, replications.

In consequence, a clustering methodology is proposed in this chapter. There are several features considered with this methodology in order to be appropriate for the problem defined. First of all, the methodology takes the time dependencies and the length of time intervals into consideration; therefore, it is safe to use this methodology on time-series. Next, it uses a well-known non-model based and unsupervised clustering algorithm, which lets the user to hold the analysis without any prior or reference information. Moreover, this is also very beneficial to obtain the clustering results in a computationally short time. Thirdly, the methodology handles the replication to detect the genes with varying profiles among their replications. Finally, by using cluster validation techniques, the methodology helps the user to detect the number of clusters in a data set.

Before starting to explain the methodology, the notations used in this approach are to be instructed first. As explained before, the time course microarray experiments provide a short time-series for each gene used in the experiment. Suppose that $\mathbf{x}_i(rep = r)$ denotes a row vector of the time course expression values of the rth replication ($r \in \overline{1:R}$) of gene i ($i \in \overline{1:G}$). For simplicity, $\mathbf{x}_i^r$ will be used instead of $\mathbf{x}_i(rep = r)$. It should be noted that, the same number of time points and the same successful time values are used for all genes and for all replications. This is consistent with a general microarray experiment and provides the same length for all $\mathbf{x}_i^r$ for every value of i and r. Another notation, $x_{i,T(k)}^r$, will be used to show a single expression value of the rth replication of gene i at the kth time point ($k \in \overline{1:K}$). Based on these notations, $\mathbf{x}_i^r$ becomes equal to the row vector of [$x_{i,T(1)}^r$, $x_{i,T(2)}^r$,..., $x_{i,T(K)}^r$]. As aforementioned, either equally or unequally spaced time points may be used in microarray experiments. In the first case, the lengths of time between the time points do not change throughout the experiment. On the other hand, in the second case, the lengths of time vary between different successful time points. However, the same successful time points are used for all genes and for all replicates. The T function, therefore, is used to show the

successful times of each time point. Table 1 shows two examples, one for each case, and the use of the *T* function on these cases.

Table 4.1. Examples for equally and unequally spaced time points

	Equally Spaced	Unequally Spaced
k	*T(k)*	*T(k)*
1	1st hour	1st hour
2	2nd hour	2nd hour
3	3rd hour	4th hour
4	4th hour	8th hour
5	5th hour	16th hour
6	6th hour	48th hour

The first column in Table 4.1 shows that six time points are used in both of the experiments. The second column shows an example for an equally spaced time point experiment. The time points in this case are adjacent to each other with the same time interval, i.e., 1 hour. Finally, the last column shows an illustration for an unequally spaced time points. The time lengths between each time interval get longer as the experiment continues in this case. Although the same *k* values are used for both cases, *T*(*k*) takes different values in two cases.

Note that discrete time points are used in these experiments. These discrete time points can not supply the information about the behaviours of the genes between follow-ups. This prevents the user to catch the possible non-linear behaviours. However, those time points are determined by the experts to show the expression levels at the important follow-ups. Therefore, we are only interested in the expression levels at the discrete follow-up points and assume that there are linear expression level changes between successive time points.

The following sections explain the approaches used in our clustering methodology, starting with the approach which handles the possible differences among the replications of genes.

4. 1. Handling the Replications

In replicated microarray experiments, usually type II Biological Replicates are used. That is, a different patient is used for each replication. Possibly, each patient may have a different biological background and pathway. For example, a patient with a specific exposure in his/her past may have a different pathway than the other patients in the experiment. As a consequence of such specificities, some genes may show changes in expression profiles in different replications even though its function stays the same from patient to patient. Those genes may show the effects of these specific conditions to the patients such as the effects of

an exposure. Therefore, examining the experimental units with different patterns may lead very important biological results.

Consequently, detecting such genes carries vital importance for further analysis. Detection of these genes can be possible in clustering studies by assigning such genes into a specific cluster. Note that, there might be several genes which show the same difference at the same experimental unit, hence it is possible to group these genes in a unique cluster. The variations among the genes can be observed in two types. Figure 4.1 shows three genes where two of them show the two types of differences among the replications.

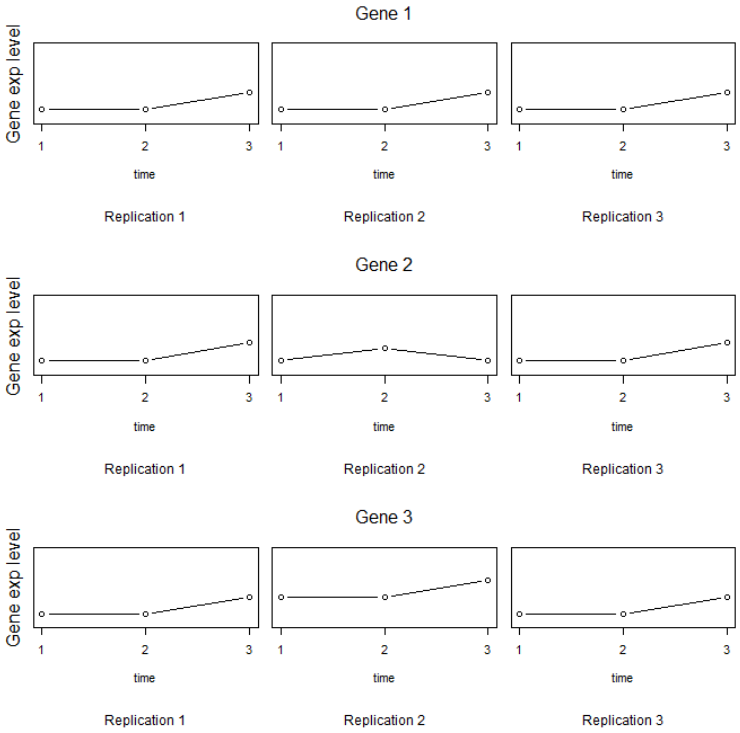


Figure 4.1. Three example genes with three replications and three time points

Each gene in Figure 4.1 has three replications, and each time-series for these genes includes three time points. *Gene 1* in this figure keeps its profile at all of the replicates. Each replication of *Gene 1* shows a constant pattern at the first time interval and increases its expression level at the second time interval. *Gene 2* also shows this pattern with its first and third replicates. However, its second replicate has a different profile. Its expression level increases in the first interval while decreases to the baseline in the second interval. Therefore, *Gene 2* shows completely a different pattern within its second replicate which is the first type of the differences among the replications. Moreover, *Gene 3* again shows the same pattern with *Gene 1*, within its first and third replicates. The second replicate of *Gene 3* also shows the same shape with the first and third replicates except for a change in the

magnitude level. Such situations can be acknowledged as the second type of differences among the replications, and they may carry as beneficial information as the first type. As a consequence each gene in Figure 4.1 is important to be detected by a clustering algorithm and should be assigned into different clusters no matter how similar their first and third replicates are.

For the methodology in this thesis, an approach is proposed to handle the replications of the genes. This approach joins all replicates of each gene consecutively before starting the clustering. Therefore, each gene is represented as a whole time-series which includes all of the replicates (see Eq. 4.1)

$$\mathbf{X}_i = \bigcup_{r=1}^{R} \mathbf{x}_i^r \qquad (\mathbf{4.1})$$

Eq. 4.1 shows that the same order of replications is used while joining the replicates. This approach can be visualized with Figure 4.2 which shows the joined form of the three genes represented in Figure 4.1.

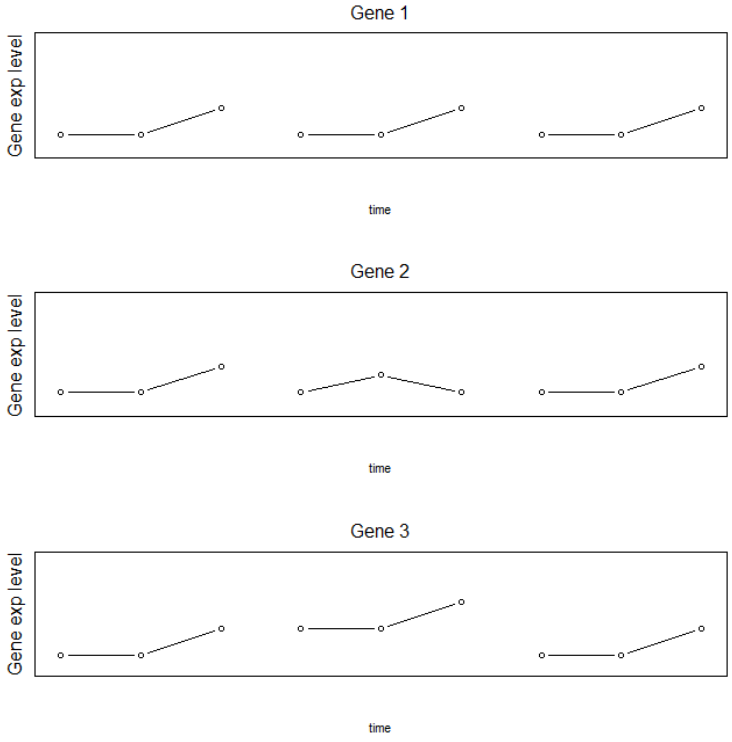


Figure 4.2. Joined form of the three genes in Figure 4.1

As it can be seen from Figure 4.2, the edges which show the replications in Figure 4.1 were disappeared after the joining process. This time each row shows a representation of a gene which includes all of its replicates. It should be noted that, during the joining process, no artificial profiles were created. For instance, the last follow-ups of the first replications are not linked to the first follow-ups of the second replications to prevent any artificial time interval between the replications. The genes with replications will be shown with this approach in all of the graphical representations throughout this thesis.

The motivation behind this approach is to project the possible differences among the replications of a gene. It is expected that such differences between the replications will be caught when the distances are measured between the pair of genes. Such differences are going to increase the distance between the genes.

After combining the replications, the genes in the data set get ready to be clustered. However, as aforementioned, in order to cluster the objects in a data set, the dissimilarities between them should be calculated first. Therefore, the next section will state the distance metrics used in the clustering algorithm proposed in this thesis.

4. 2. Distance Metrics

Calculating the dissimilarities between the objects to be clustered is a must for clustering studies. Since clustering means grouping the similar objects, a definition must be stated to assess the similar objects. The dissimilarities can be calculated with the help of the distance measures. Section 3.4. in the previous chapter reviewed several distance measures used in the literature. For the clustering algorithm proposed in this thesis, two of those distance measures are used.

Squared Euclidean distance is a powerful tool to see the similar objects. When it is applied to the time-series microarray data sets, it can measure the distances between the times-series based on the closeness of the expression levels at each time point. The squared Euclidean distance can be measured on the time-series with the notations introduced in the beginning of this chapter as follows:

$$d(\mathbf{X}_i, \mathbf{X}_j) = \sum_{r=1}^{R} \sum_{k=1}^{K} (x_{i,T(k)}^{r} - x_{j,T(k)}^{r})^2 \qquad (\mathbf{4.2})$$

As mentioned before, squared Euclidean distance will take squares of the expression level differences at each time point in the pair of time-series and sums them up to find the total difference between the pairs. Note that, two summation operators are used in Eq. 4.2. The second summation operator takes the summation of the squared expression level differences at each time point. The first summation operator, on the other hand, sums the total differences through the replications used in the study which intuitively provides handling the replications presented in Section 4.1. Moreover, summing the total differences at each replication separately prevents creating an artificial time interval between the replicates mentioned in, again, Section 4.1.

However, there are several drawbacks of using only the squared Euclidean distance. First, it does not consider the shape similarities of the expression profiles of the genes in the first place. Therefore, squared Euclidean distance may fail to separate the dissimilar genes with respect to their shapes when their magnitudes are close. Furthermore, squared Euclidean distance fails to catch the important features of the time-series which leads to the next two drawbacks. First, it ignores the time dependencies. Since it measures the expression level differences at each time point separately, squared Euclidean distance con not take the dependencies between the measurements into consideration. Finally, again due to considering each time point separately, this distance does not include the information on the lengths of time intervals between the time points. All of these drawbacks make using only this distance metric to cluster the time course data sets inappropriate.

Due to the drawbacks of using the distance metric, *d(x, y)*, alone, a second metric is used along with it. As explained before, Short Time Series (STS) distance measures the slope distance between the pair of time-series at each time interval and sums them up. Since it takes the differences between the slope of the time-series, STS distance can catch the similarities between the genes with respect to their shape characteristics. The STS distance can be measured with the notations used in this thesis as follows:

$$s(\mathbf{X}_i, \mathbf{X}_j) = \sum_{r=1}^{R} \sum_{k=1}^{K-1} \left[\frac{x_{i,T(k+1)}^{r} - x_{i,T(k)}^{r}}{T(k+1) - T(k)} - \frac{x_{j,T(k+1)}^{r} - x_{j,T(k)}^{r}}{T(k+1) - T(k)} \right]^2 \qquad (\mathbf{4.3})$$

Similar to Eq. 4.2, two summation operators are used in STS distance, as well. The second summation operator in Eq. 4.3 sums the slope differences through the time intervals among the time-series. Note that, the upper limit of that summation operator is *K*-1, since the number of time intervals should be one less than the number of time points in a time-series. However, the first summation operator in Eq. 4.3, again sums up the slope distances through the replications of genes in order to achieve handling the replications. Identical to the squared Euclidean distance metric in Eq. 4.2, summing slope distances from each replication do not create any artificial time interval between the replications. Furthermore, since STS distance uses the information between the time points, it takes the time dependencies into consideration. Moreover, this distance metric also includes the time lengths between the time points. Therefore, it can be used with both types of data collection methods in time-series, equal and unequally time spaces.

Two distance metrics are shown to be used on measuring the dissimilarities between the time-series so far. Each of these metrics measures the dissimilarities from different perspectives. The distance metric, *d(x, y)*, defines the dissimilarity based on the magnitudes of the expression profiles whereas shape metric, *s(x, y)*, defines that based on the shapes of the profiles.

For the clustering algorithm in this thesis, both of these metrics are used to define the dissimilarities. Two distance matrices are obtained by using these two distance metrics. The first distance matrix, **D**, is a G x G symmetric matrix and includes the magnitude differences

between each pair of genes in the data set by using the squared Euclidean distance. The second distance matrix, **S**, which is also a G x G symmetric matrix, defines the shape dissimilarities between the expression profiles of genes by using the Short Time Distance metric. In order to find the dissimilarities between the genes in the clustering algorithm, a convex combination of these two matrices is obtained. However, before the combination of the metrics, standardization is applied to both matrices. It is observed that the ranges of the matrices can have great amount of differences, and this creates a dominance of the matrix with wider range on the other one. To prevent such dominations, the values in both matrices are divided to the range of the related matrix. Furthermore, this standardization scales the values in both matrices between 0 and 1. After this standardization, the combination matrix, denoted by $\overline{\mathbf{D}}$, which is also a G x G symmetrical matrix is obtained as follows (Eq. 4.4):

$$\overline{\mathbf{D}} = w * \mathbf{D} + (1 - w) * \mathbf{S} \qquad (\mathbf{4.4})$$

where $w \in [0, 1]$ and $1 - w$ are weights for each metrics. By adjusting different values for w in Eq. 4.4, the user can give different emphasis on the distance metrics. For example, if an analyst wants to give more emphasis on the magnitude similarities, that analyst may give values close to 1 for w. On the other hand, w can be set close to 0 in order to catch the shape similarities with more emphasis.

4. 3. Clustering Algorithm

Previous section presented the methodologies used for obtaining the distance matrix which shows the dissimilarities between each pair of genes. A hierarchical clustering algorithm by using this distance matrix is proposed in this section. As explained earlier, hierarchical clustering builds the dendrogram of the objects to be clustered. This dendrogram can be cut at any level depending on the desired number of clusters. In a time-series microarray study, this dendrogram would show the similarities between expression profiles of genes in a hierarchical way. Therefore, the clusters obtained from this algorithm would show the specific profile patterns and the genes which show these patterns. As aforementioned, hierarchical clustering needs a linkage method in order to define the distances between the groups of objects while constructing the dendrogram. The methodology proposed in this thesis uses the *Ward's method* for the linkage method as default. However, the user is free to choose any linkage method while using this algorithm. Figure 4.3 gives the pseudo code for the algorithm proposed in this thesis, **Algorithm CGR**.

Another important challenge in clustering studies is detecting the number of clusters when it is not known a priori. In order to decide on the number of clusters, cluster validation techniques can be used and numerous validation techniques were presented in Section 3.5. A cluster validation methodology modified from these previously mentioned techniques will be used. The following section presents these techniques.

Algorithm Clustering Genes with Replications (CGR):

STEP 0: Initialization

G: the number of genes
K: the number of time points
R: the number of replications
w: Weight for the Squared Euclidean distance while combining the metrics
Input *G* x *m* matrix (where *m* = *K*R*) of preprocessed gene expression data

STEP 1: Handling the Replications

Converting the input data into a 3-dimensional *(G x K x R)* data where the new dimension represents the replicates

STEP 2: Distance measure

1. **For** i = 1 to G
   For j = 1 to G
   $\mathbf{D}(\mathbf{X}_i, \mathbf{X}_j) = \sum_{r=1}^{R} \sum_{k=1}^{K} (x_{i,T(k)}^{r} - x_{j,T(k)}^{r})^2$

   $\mathbf{S}(\mathbf{X}_i, \mathbf{X}_j) = \sum_{r=1}^{R} \sum_{k=1}^{K-1} \left[\frac{x_{i,T(k+1)}^{r} - x_{i,T(k)}^{r}}{T(k+1) - T(k)} - \frac{x_{j,T(k+1)}^{r} - x_{j,T(k)}^{r}}{T(k+1) - T(k)} \right]^2$

2. $\mathbf{D} = \frac{\mathbf{D}}{[\max(\mathbf{D}) - \min(\mathbf{D})]}$
3. $\mathbf{S} = \frac{\mathbf{S}}{[\max(\mathbf{S}) - \min(\mathbf{S})]}$
4. $\bar{\mathbf{D}} = w * \mathbf{D} + (\mathbf{1} - \mathbf{w})\mathbf{S}$

STEP 3: Clustering

Hierarchical clustering by using $\bar{\mathbf{D}}$

Figure 4.3. A pseudo code of the algorithm

4. 4. Cluster Validation

A clustering algorithm should result in clusters where objects within each cluster are as compact as possible, whereas each cluster is as dissimilar as possible to the other clusters. Consequently, when such a clustering algorithm is used with the time course microarray studies, each cluster presents a specific expression profile while none of the two clusters show the same profile. However, the number of specific profiles is not known for most of

the microarray clustering studies. This leads to the uncertainty of choosing the number of clusters, especially when unsupervised clustering algorithms are used.

In order to decide on the number of clusters, which is the number of profiles in time-series gene expression data sets, one approach is to measure the qualities of the clusters. Two quality measures can be deduced from the previous explanations. First, since each cluster should contain only one specific profile, the variance within a cluster should be as small as possible. Furthermore, on the account that a specific profile should not be presented in more than one cluster, the variances between clusters should as much as possible in a set of clusters. Therefore, the two variance measures can be used to obtain the optimum number of clusters which provides the best set of clusters.

Since the variances within and between the clusters are directly related to the distances between the objects in these clusters, distances measured in **Algorithm CGR** are used for cluster validations. A distance measure for the within cluster variance and two distance measures for between cluster variances are proposed in this section.

A small within cluster variance, which implies a compact cluster, can be obtained when the distances between the objects in that cluster are as small as possible. Our algorithm sums up the distances between each pair of genes within a cluster to present the within cluster variance. This measure is named as *sum(within)* distance and can be calculated as:

$$sum(within)_{N_c} = \underset{n = 1, \dots, N_c}{max} \left\{ \sum_{i=1}^{|\mathbb{C}_n|} \sum_{j=1}^{|\mathbb{C}_n|} \bar{\mathbf{D}}(\mathbf{X}_i, \mathbf{X}_j) \right\} \qquad (\mathbf{4.5})$$

where N_c is the number of clusters; n is the given cluster; $\mathbf{X}_i$ and $\mathbf{X}_j \in n$ ($i \neq j$); $|\mathbb{C}_n|$ is the number of genes in cluster n. The *sum(within)* distance, presented in Eq. 4.5, calculates the total distances between each pair of genes within all clusters and assigns the highest of them as the within cluster variance for a set of clusters. For compact clusters, a small *sum(within)* value is preffered.

To assess the cluster quality, the between cluster variances should also be included in the analysis. Two different distance measures are used to measure the between cluster distances. The first one finds the distance between the closest pair of genes from different clusters and assigns minimum of them as the between cluster variance for that set of clusters. The second distance measure, however, calculates the average distances between each pair of genes from two clusters and accepts the smallest average distance as the between cluster variance. These two distances are called as *min(between)* and *mean(between)* distances and can be calculated as Eq. 4.6 and Eq. 4.7, respectively

$$min(between)_{N_c} = \underset{n_I = 1, \dots, N_c}{min} \left\{ \underset{n_J = 1, \dots, N_c}{min} \left(min\left(\bar{\mathbf{D}}(\mathbf{X}_i, \mathbf{X}_j) \right) \right) \right\} \qquad (\mathbf{4.6})$$

where N_c is the number of clusters; n_I and n_J are the given clusters ($n_I \neq n_J$); $\mathbf{X}_i \in n_I$ and $\mathbf{X}_j \in n_J$.

$$mean(between)_{N_c} = \min_{n_I = 1, \dots, N_c} \left\{ \min_{n_J = 1, \dots, N_c} \left(\frac{\sum_{i=1}^{|\mathbb{C}_{n_I}|} \sum_{j=1}^{|\mathbb{C}_{n_J}|} \bar{\mathbf{D}}(\mathbf{X}_i, \mathbf{X}_j)}{|\mathbb{C}_{n_I}| * |\mathbb{C}_{n_J}|} \right) \right\} \quad (\mathbf{4.7})$$

where N_c is the number of clusters; n_I and n_J are the given clusters ($n_I \neq n_J$); $\mathbf{X}_i \in n_I$ and $\mathbf{X}_j \in n_J$; $|\mathbb{C}_{n_I}|$ is the number of genes in cluster n_I and $|\mathbb{C}_{n_J}|$ is the number of genes in cluster n_J.

As a result, the user can obtain estimations for the within cluster variation by using Eq. 4.5 and between cluster variation by using Eq. 4.6 and 4.7 for a set of clusters. However, joint use of these measures is necessary to reach to the correct cluster sets. We propose a cluster validation score which finds the ratio of the within cluster variance to the between cluster variance in order to find the optimum set of clusters. We can obtain two validation scores, given in Equations 4.8 and 4.9, by using either one of the between cluster variance measures.

$$VD1_{N_c} = \frac{sum(within)_{N_c}}{min(between)_{N_c}} \quad (\mathbf{4.8})$$

$$VD1_{N_c} = \frac{sum(within)_{N_c}}{mean(between)_{N_c}} \quad (\mathbf{4.9})$$

Since, small within cluster variances and high between clusters variances are desired, small validation scores would show better sets of clusters. In a cluster study, the number of clusters can be chosen from 1 to the number of objects in the data set. Therefore, the analyst can find validation scores for each set of clusters from 1 to the number of objects, and choose the number of clusters with respect to those values. We propose that, on the graph of the validation scores, the number of clusters which has small validation score and that leads to significant decrease in the score can be selected as the optimum set of clusters.

As consequently, this chapter presents the methodology proposed for clustering the time course gene expression data sets. The next chapter demonstrates the applications of these methodologies on simulation and real data sets in order to display the usage of the methodology.

CHAPTER 5

NUMERICAL EXAMPLES

Previous chapter presented the methodology proposed in this thesis. In this chapter, the applications of this methodology on several data sets are exhibited. There are two main sections in this chapter. The first section displays the applications of **Algorithm CGR** on three simulated and a real data set. Cluster validation techniques are demonstrated on two simulated and a real data set in the second section.

5. 1. Clustering the Genes

As mentioned in the prologue of this chapter, this section illustrates the **Algorithm CGR** on three simulation studies and one real life study. In the first simulation study, a simple data set is generated mainly to show the importance of the metrics. Next, the algorithm is tested on a similar simulation study on Irigoien et al. (2011) to compare the success of our algorithm with different algorithms. Third, different scenarios will be generated on a different simulation study in order to show the advantages and disadvantages of **Algorithm CGR** under different situations. Finally, the algorithm is examined on a real data set which was also studied by Irigoien et al. (2011).

5. 1. 1. Simulation Study 1

The first simulated data set contained 90 time-series with three follow-up points. Replications were not used in this simulation study. The hypothetical time-series generated for this simulation study can be seen in Figure 5.1.

There were several patterns in this data set. Firstly, the genes were grouped in three magnitude levels. The time-series were generated around the expression levels of 4, 10 and 16. This led to the three groups based on the magnitude levels as low, medium and high levels. Furthermore, the time-series experienced different shapes in each of those magnitude levels. There were also three groups with respect to the shapes of the time-series as decreasing, increasing or constant shapes. Each of the shape groups were seen in each magnitude levels, hence there were 9 different patterns in this simulated data set. Ten time-series were generated for each of these patterns (Figure 5.2).

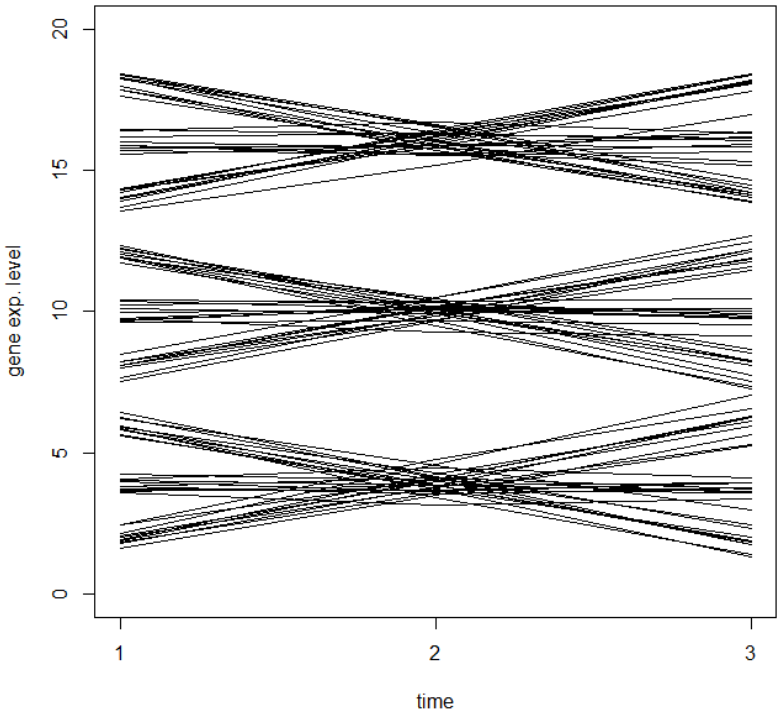


Figure 5.1. First hypothetical data set with 90 time-series

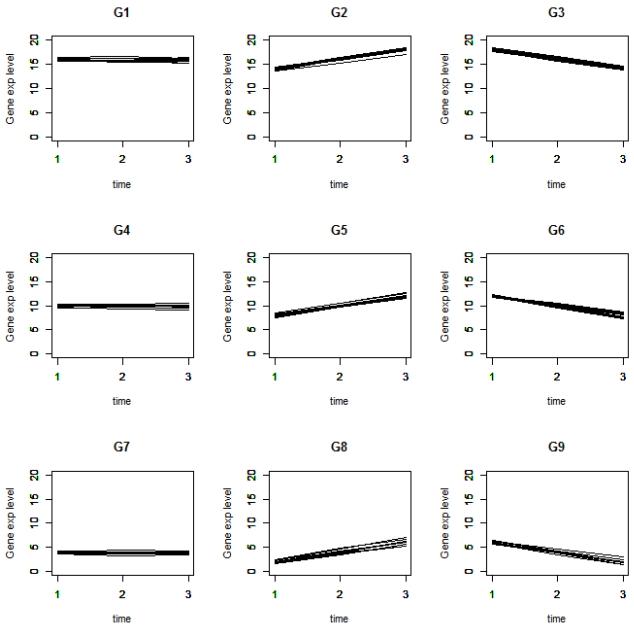


Figure 5.2. Nine different patterns in the hypothetical data set in Figure 5.1

Algorithm CGR was used to see if we can detect these 9 profiles with using 9 clusters. Before using both metrics together, however, each metric was used alone to illustrate their abilities. In the first case, only the squared Euclidean distance was used (w = 1) and 9 clusters were searched. The outcome as 9 clusters were displayed in Figure 5.3.

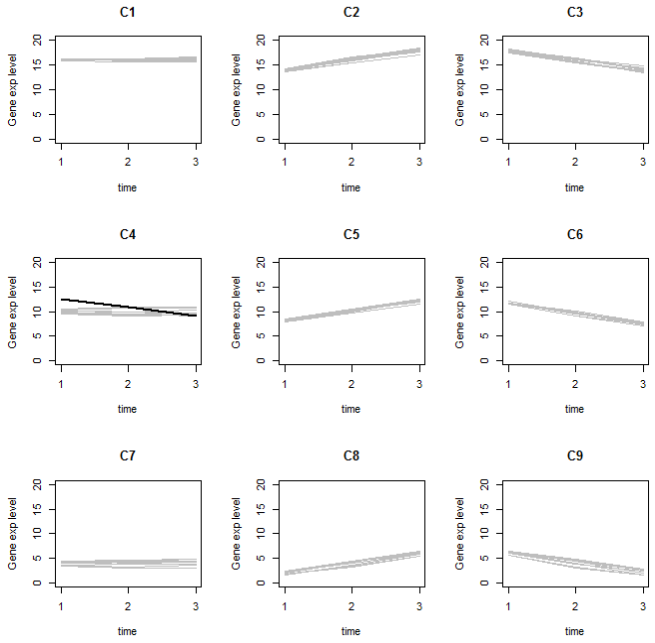


Figure 5.3. Clustering results on the simulation set 1 with w = 1

The results showed that, the algorithm was able to detect the 9 profiles. However, there was a mistake within the results. In the fourth cluster (left column of the middle panel) one gene showed a decrease (thick black line), whereas the other ones were constant genes (thin gray lines). The constant genes at the medium expression level constituted the majority of this cluster; thus, the decreasing gene in that cluster was a strange one for C4. Moreover, there was another cluster, C6, which included the decreasing genes at the medium expression level, hence the strange gene in C4 should have been in C6. However, since the magnitude difference between them was not large enough, that gene which should be in C6 was clustered in C4. That was an example for the drawback of using only the squared Euclidean distance metric: It may fail to divide the close groups into different ones even when their slopes are different.

In the next study, the same data set was tried to be partitioned into 9 clusters by using only the Short Time Series distance. For this reason, w was set to be 0. The clusters obtained from this approach were displayed in Figure 5.4.

The results showed that when only the shape metric was used, the algorithm could divide the different shapes into different clusters. However, the results were incorrect. Excluding the squared Euclidean distance resulted in groups of genes with similar shapes but from different levels. The panels in Figure 5.4 showed that, each cluster presented only one shape, however, there was not a clear magnitude level in the clusters. For example, C1, C2 and C3 included the constant shape genes. C1 and C2, nevertheless, contained the constant shape genes from each magnitude level while C3 held the genes from middle and high levels. In a correct division, each cluster should contain only one of these magnitude levels.

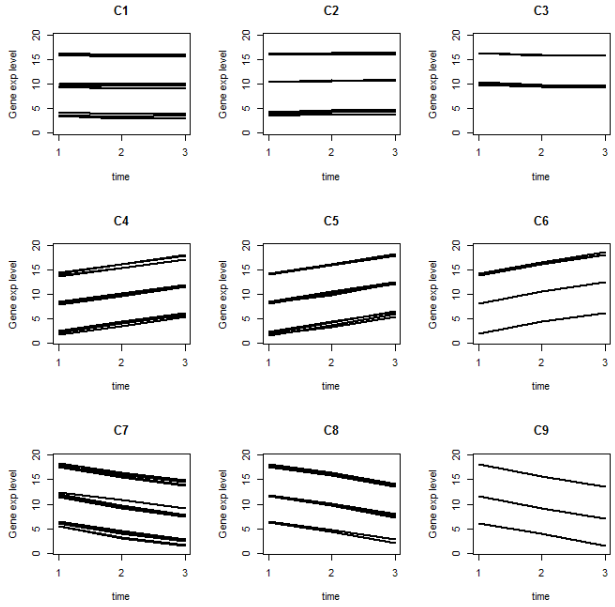


Figure 5.4. Nine clusters from the data set displayed in Figure 5.1 with $w = 0$

Consequently, previous examples showed that the algorithm had difficulties when only one of the metrics was used. As the last example for that simulated data set, **Algorithm CGR** was applied on these 90 time-series by using both metrics with equal weights, i.e., 0.5.

Figure 5.5 shows the results of clustering study on the first simulation data with equal weights to both distance and shape metrics. It can be clearly seen that each cluster contained the genes from only one shape and magnitude groups. Therefore, the panels in Figure 5.5 showed that **Algorithm CGR** was successful in dividing the profiles into different clusters when w was taken as 0.5.

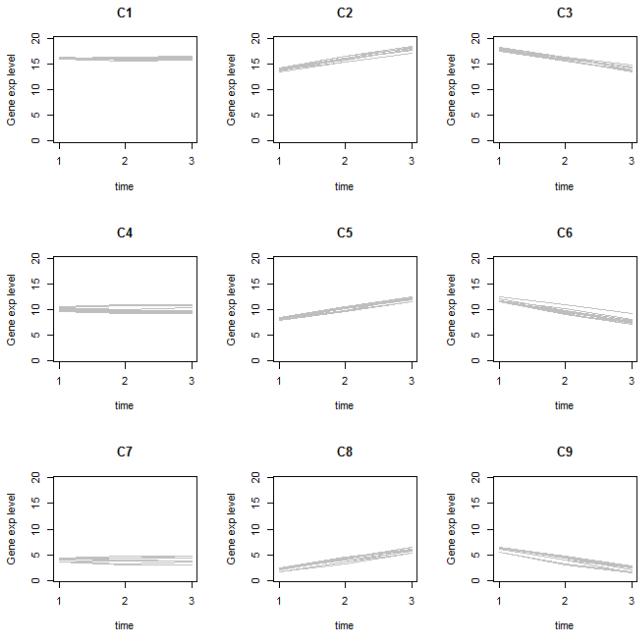


Figure 5.5. Clustering results on the simulation set 1 with $w = 0.5$

The first simulation study showed that the algorithm was able to detect the patterns in the data set when both metrics were used. However, as it was mentioned before, replications are commonly used in microarray experiments. Moreover, the products of microarray experiments provides higher number of genes compared to 90 genes in this simulation set. Thus, further simulation sets were generated in order to challenge the algorithm in the following subsections.

5. 1. 2. Simulation Study 2

The second simulated data set was generated based on the simulation study in Irigoien et al. (2011). As it was mentioned in the literature review, the purpose of this paper was also clustering the time-series gene expression profiles with possible differences among the replications. Since the data set in Irigoien et al. (2011) was not publicly available, a similar one was generated for this thesis. In this study there were 13 groups of patterns. Each pattern was seen in 20 genes, which resulted a total of 260 genes in the data set. Each time-series was followed on 6 time points and two replications were used for every gene under each time point. The average expression levels of the genes in all 13 groups can be seen in Figure 5.6. There are two series within all panels of this Figure, each corresponding to one replication.

The first group contained only the constant profile genes. The next three groups displayed an increasing pattern with different features. For example, G2 showed a faster increase than the genes in G3. The groups between G5 and G8 showed an up-down profile with peak points at different time points. Furthermore, the genes in G9, G10, G11 and G12 presented

up-down-constant, down-up, sinusoidal and down-up-constant patterns, respectively. The important and common feature about the first 12 groups is that both replications showed the same profile. However, G13, included the genes with different replications. The first replications of the genes in that group displayed a pattern similar to the genes in G2 while the second replicates demonstrated a profile similar to the genes in G7. As a result, a successful clustering study on this data set should be detecting the genes in G13 as a different group than the other ones besides dividing each pattern in the first 12 groups into different clusters.

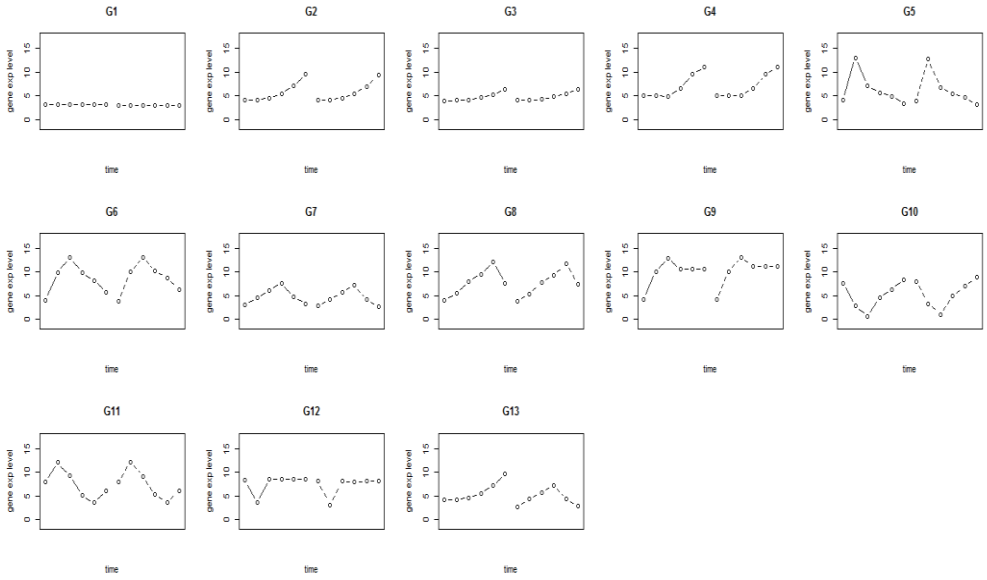


Figure 5.6. Groups in the simulated data set from Irigoien et al. (2011)

Algorithm CGR was applied on this data set with giving equal weights to both metrics (w = 0.5). The result are shown in Figure 5.7. It can be seen from that figure that none of the genes were shown with thick black lines. This meant that all the genes clustered in correct group when the data set was divided into thirteen clusters. The first 12 clusters displayed specific patterns from the first 12 groups. More importantly, the algorithm successfully assigned the genes with differences among the replications into a specific cluster, C13.

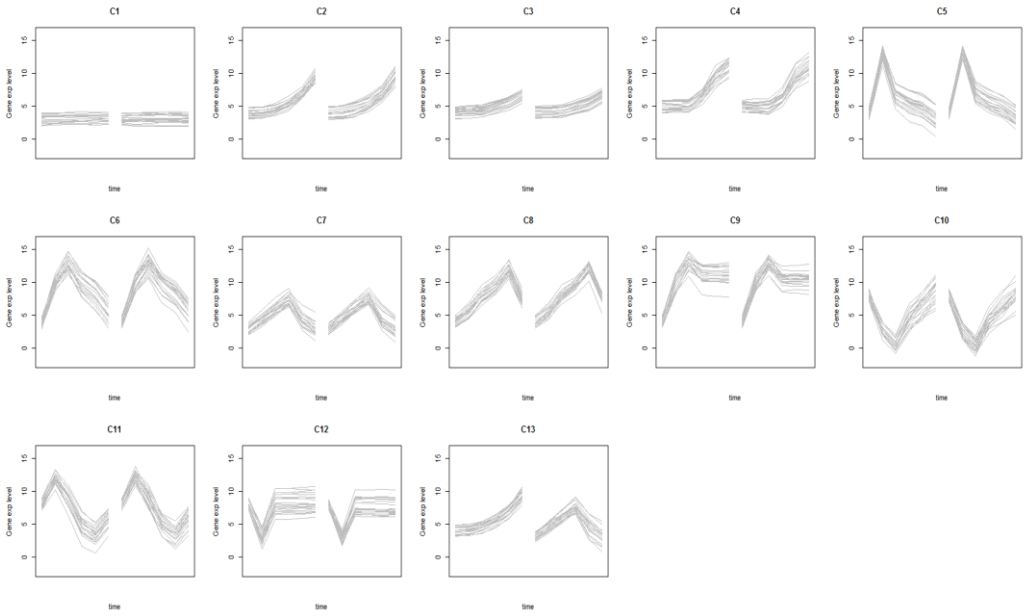


Figure 5.7. Clustering results on the simulation set 2 with $w = 0.5$

Next, different weights than 0.5 were tried for both metrics and the results were obtained. Four different weights (0, 0.25, 0.75 and 1) and their complements to 1 were used for the distance and shape metrics, respectively. Irigoien et al. (2011) provided a table which showed the number of genes which were clustered in wrong groups when several methods were applied on a similar data set. We extended their table by adding the results of **Algorithm CGR**. The "misclustered genes" in each group were calculated for each method. The number of misclustered genes were found as follows: First, the majority of the groups were detected in each cluster. This was found as the group with the most frequent genes in a cluster. Next, the genes which are not from that major group in that cluster were counted as misclustered genes to its original group. For example, in Figure 5.3, the majority of genes in C4 was from G4, since most frequent genes in that cluster were from G4. However, there was another gene, which was shown with thick black line, generated under G6. Since that gene was clustered in a cluster whose majority group was not the same with its original group, this gene was counted as a misclustered gene for G6. By using this approach, the misclustering numbers and percents were calculated for the data set in simulations study 2 when different weights were used (Table 4.1)

Table 4.1 shows the number of misclustered genes for each group and the percent of the total misclustered genes to the total number of genes. The first column in the table shows the method used to cluster the genes. The columns under G1 to G13 display the number of misclustered genes for each group while the percentage of the misclustered genes to the total number of genes are given in the last column. First five rows show the results of **Algorithm CGR** with five different weight selections. The remaining rows were directly taken from Iriogien et al. (2011) and displays the misclustering results of several approaches. Their approach, for example, grouped 2 genes from the 9^{th} group in different clusters. Further,

they tested k-means approach with three different distance measures: Euclidean, correlation and Procrustes distances. The next two rows were two different algorithms named as ORIOGEN and EMMIX. As explained in the literature review, ORIOGEN could not use the replications; therefore, G13 was not used with ORIOGEN. Finally, the remaining rows show the results of mixture models $E_i - M_j$ with $i = 0, 1, 2, 3; j = 1, 2, 3$ following the procedure presented in Celeux et al. (2005) where E_i describes the case for the random effect parameter and M_j describes different mixture parameter cases.

Table 5.1 shows that **Algorithm CGR** was successful on clustering this data set. The only misclustered genes were observed when only the distance metric was used ($\overline{\mathbf{D}} = \mathbf{D}$). In that case, 7 genes from G2 were assigned in the cluster whose majority was constituted by the genes from G4. Note that, those two groups included similar genes especially with respect to theşr esxpression levels, and therefore distinguishing them was challenging. Moreover, the algorithm clustered the genes perfectly, i.e., without any misclustered genes, when the shape metric was used.

Table 5.1. Misclustered number of genes for several methods on the simulation data from Irigoien et al. (2011)

Method	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**	**G8**	**G9**	**G10**	**G11**	**G12**	**G13**	**Percent**
$\mathbf{D}$	-	7	-	-	-	-	-	-	-	-	-	-	-	3.59
$0.75 * \mathbf{D} + 0.25 * \mathbf{S}$	-	-	-	-	-	-	-	-	-	-	-	-	-	0
$0.50 * \mathbf{D} + 0.50 * \mathbf{S}$	-	-	-	-	-	-	-	-	-	-	-	-	-	0
$0.25 * \mathbf{D} + 0.75 * \mathbf{S}$	-	-	-	-	-	-	-	-	-	-	-	-	-	0
$\mathbf{S}$	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Irigoien Procedure	-	-	-	-	-	-	-	-	2	-	-	-	-	1.026
k-means Euclidean	15	-	-	15	-	-	2	-	-	-	-	-	15	24.10
k-means correlation	6	-	15	15	-	-	-	-	-	-	-	-	15	26.15
k-means-Irigoien dist	-	-	-	-	-	2	-	-	-	-	-	-	15	8.72
ORIOGEN	15	-	-	-	9	-	-	-	15	-	15	15	?	35.38
EMMIX	-	-	-	-	-	7	2	-	-	-	-	-	-	4.61
$E_1 - M_1$	15	-	-	15	-	7	4	-	2	-	-	-	-	22.05
$E_1 - M_2$	-	-	-	15	-	-	-	-	-	-	-	-	-	7.69
$E_1 - M_3$	-	-	-	-	1	2	-	-	2	-	-	-	15	10.26
$E_2 - M_1$	15	-	-	15	-	2	-	-	2	-	-	-	-	17.43
$E_2 - M_2$	-	-	-	15	-	-	-	-	2	-	-	-	15	16.41
$E_2 - M_3$	-	-	-	-	-	-	-	-	2	-	-	-	-	1.026
$E_3 - M_1$	-	-	-	15	-	1	-	-	2	-	-	-	-	9.23
$E_3 - M_2$	15	-	-	-	-	-	-	-	2	-	-	2	-	9.74
$E_3 - M_3$	15	-	-	-	-	-	-	-	2	-	-	2	-	9.74
$E_0 - M_1$	15	-	-	-	-	-	-	-	2	-	-	2	-	9.74
$E_0 - M_2$	15	-	-	-	-	-	-	-	2	-	-	2	-	9.74
$E_0 - M_3$	-	-	-	15	1	2	2	-	2	-	-	4	-	13.33

An interesting result of Table 4.1 was that the correct clusters were also found by using only the shape metric ($\overline{\mathbf{D}} = \mathbf{S}$). This may lead to incorrect interpretation that only the shape metric would be sufficient to cluster the gene expression profiles. However, this could be the result since the groups in that data set was too artificial and easy to decompose with respect to their shapes. Therefore, this simulation data set was expanded with two new groups in order to make the clustering more challenging. Figure 5.8 shows the means of the genes in each group with the two new groups.

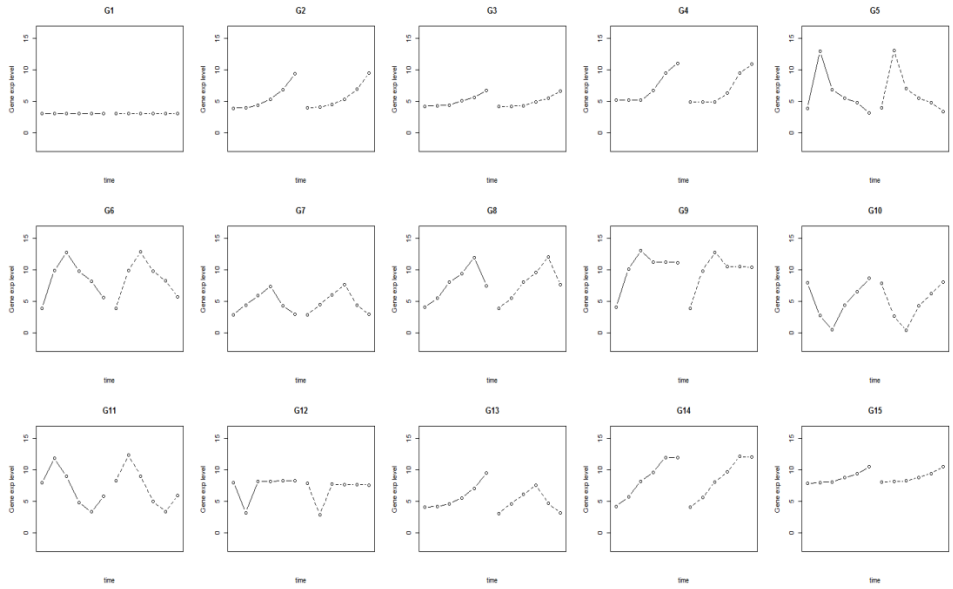


Figure 5.8. Expanded simulated data with two new groups

First 13 groups in Figure 5.8 are the same ones in Figure 5.6. G14 presented a very similar profile to G8. The genes in these groups showed a monotonic increase in the first 5 time points. However, at the last time interval, the genes in G14 kept their expression levels whereas the genes in G8 experienced a decrease. Thus, the only difference between these two groups were at their last time interval. Moreover, they were very close with respect to their magnitude levels. G14 was added to challenge the squared Euclidean distance metric. The second added group, G15, on the other hand, showed the same shape profile with the genes in G3. Their magnitude levels, however, was different. Since their shapes were same, these genes were expected to challenge the shape metric.

With these two new groups, this simulation study was expanded to a data set with 15 groups each with 20 genes. In order to see the responses of **Algorithm CGR** to the added groups, it was run by using the metrics one-by-one. Figure 5.9 shows the results when only the distance metric was used, i.e. $w = 1$.

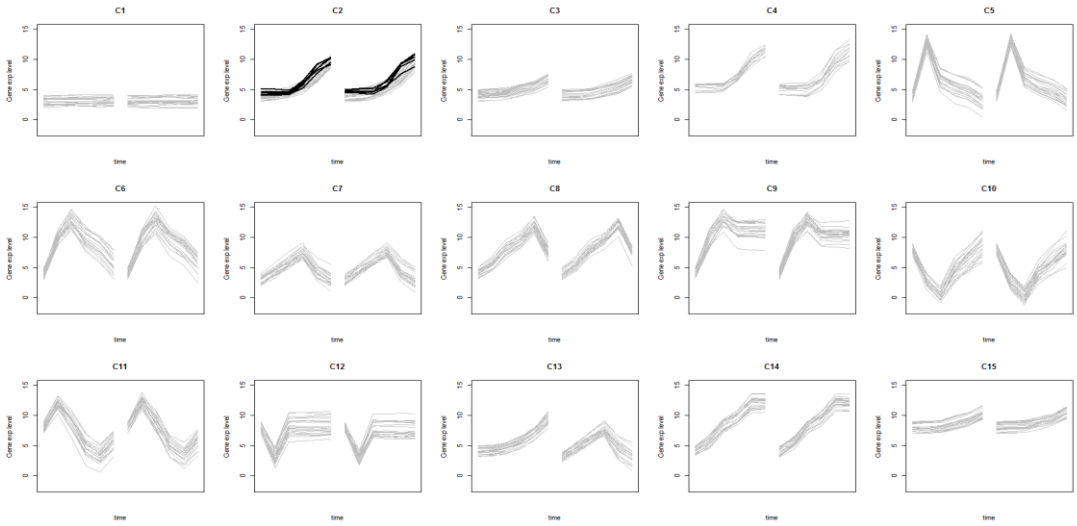


Figure 5.9. Fifteen clusters obtained from the expanded simulation study with $w = 1$

Figure 5.9 shows that **Algorithm CGR** failed to divide the genes into correct clusters. It seemed that C2 included the whole genes from G2 and seven genes from G4. That means that 7 genes from G4 were misclustered in C2. Since the new genes in G14 were similar to those in G8, it might be expected that the algorithm would fail to separate these two groups. However, it should be noted that, in a clustering study every object depends on the other objects. Therefore, adding new genes similar to G8 led to failures in dividing other groups, G2 and G4, in this case.

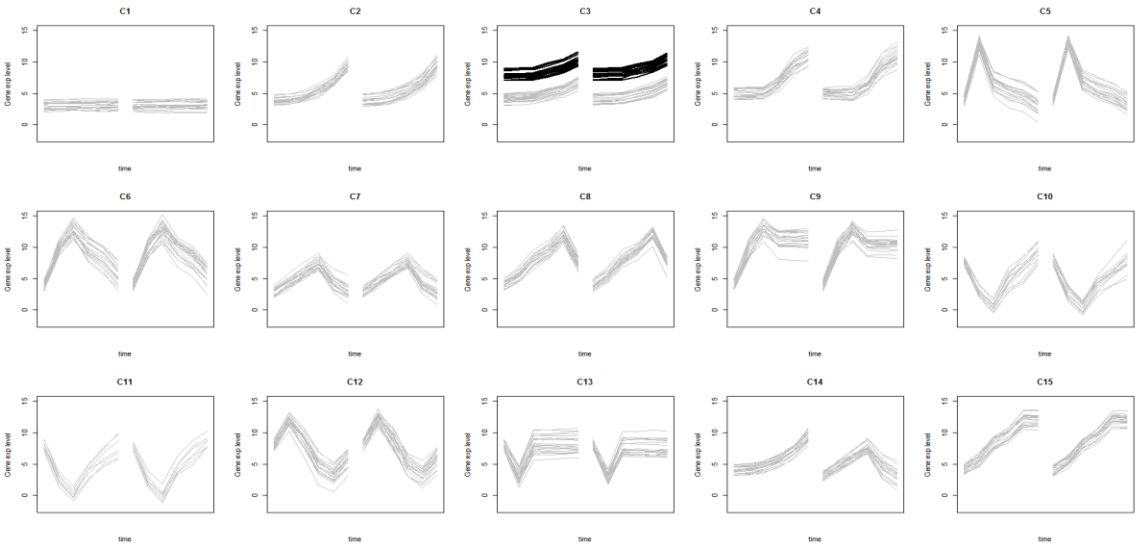


Figure 5.10. Results of **Algorithm CGR** on the expanded simulation data when with $w = 0$

Next, **Algorithm CGR** was used on this data set by using only the STS distance and 15 clusters were drawn (Figure 5.10).

Figure 5.10 shows that the algorithm could not separate the genes from G3 and G15. The genes from both groups were clustered in C3. Therefore, all genes from G15 were misclustered into G3. Moreover, since these two groups emerged in this case, algorithm divided a profile into two groups in order to obtain 15 clusters. It can be seen from Figure 5.10 that the genes from G10 were unnecessarily divided into two clusters: C10 and C11.

These two studies showed that **Algorithm CGR** failed to reach the correct clusters when only one of the metrics was used. Next, these two metrics were used with equal weights to evaluate the algorithm. Figure 5.11 shows the results of the clustering algorithm when both metrics are used with equal weights ($\bar{\mathbf{D}} = 0.5 * \mathbf{D} + 0.5 * \mathbf{S}$).

Algorithm CGR succeeded to divide the genes from different groups into different clusters when equal weights were given to the metrics. Contrary to the previous two results, algorithm could separate the genes from G15 than the genes from G3 and displayed them in C3 and C15, respectively. Moreover, none of profiles were combined with each other. Finally, the genes with different replications in G13 were assigned into a specific cluster, C13, successfully.

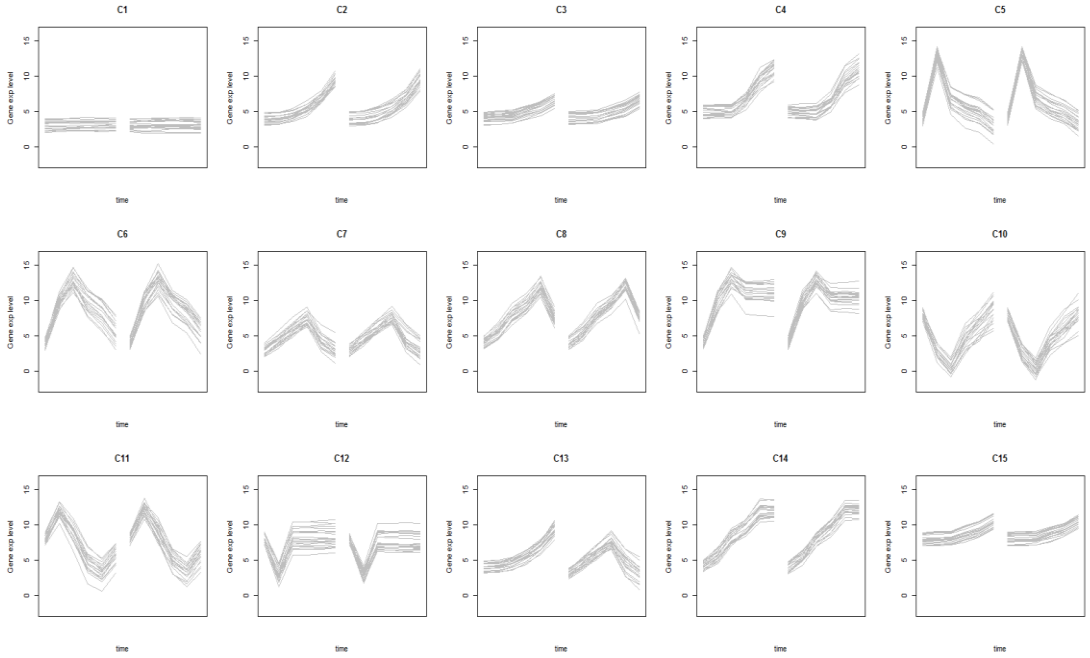


Figure 5.11.Clustering results of **Algorithm CGR** on the expanded simulation data with $w = 0.5$

Next, in order to see general results, this expanded simulation data set was generated 1000 times and the number of misclustered genes were calculated. The misclustered genes were found as the previous procedure. Suppose that $M_{k,l}^m$ shows the number of genes from group *k*, misclustered in the cluster whose majority is constituted by the genes from group *l*, at the m^{th} iteration. Therefore the average misclustering rate, over 1000 iterations, for group *k*, which is shown as MC_k, can be calculated as follows:

$$MC_k = \frac{\sum_{m=1}^{1000} \sum_{l=1, l \neq k}^{15} M_{k,l}^m}{\sum_{m=1}^{1000} n_k^m} \qquad (\mathbf{5.1})$$

where, n_k^m shows the number of genes generated for group *k* at the m^{th} iteration. For example, n_k^m was 20 for this simulation study since 20 genes were generated for each group at each iteration. The results can be seen in Table 5.2.

Table 5.2. Average misclustering rates over 1000 iterations under different weight selections

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.07
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.03	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
D*1 + S*0	0.00	0.06	0.00	0.09	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01

The first column in Table 5.2 shows the weight selections for the clustering process. The columns between G1 and G15 displays the average number of misclustered genes, MC_k, over 1000 iterations for each group, whereas the last column gives the average rate of misclustered genes to the total number of genes over all groups and iterations which is calculated as in Eq. 5.2.

$$MR = \frac{\sum_{k=1}^{15} \sum_{m=1}^{1000} \sum_{l=1, l \neq k}^{15} M_{k,l}^m}{\sum_{k=1}^{15} \sum_{m=1}^{1000} n_k^m} \qquad (\mathbf{5.2})$$

Remember that, Table 5.1 revealed that when only the distance metrics was used ($\bar{\mathbf{D}} = \mathbf{D}$), the algorithm only failed to separate G2 and G4. These two groups were similar with respect to their profiles: G2 had an exponential increase whereas G4 presented a monotonic increase with a constant pattern at the first two time intervals. In Table 5.2, this led to 0.06 * 100 = 6 and 0.09 * 100 =9 percent of the genes misclustered from G2 and G4, respectively. However, the average misclustering rate of all groups was very small, 0.01, with $w = 1$. This rate remained similar with the selection of $\bar{\mathbf{D}} = 0.75 * \mathbf{D} + 0.25 * \mathbf{S}$. However, the misclustering rates in both G2 and G4 decreased when the shape metric was involved.

Moreover, the average misclustering rate decreased as more emphasis was given on the STS distance. The average misclustering rate were obtained smaller than 0.01 for both weight selections $w = 0.50$ and $w = 0.25$. Hence, the shape metric might be accepted as more useful to decompose the patterns, especially when the profiles are close to each other based on the magnitude levels. Shape metric, nevertheless, was not to be used alone on such experiments. Table 5.2 shows that when only the shape metric was used, the algorithm failed the most and gave the highest average misclustering rate. The reason for this result was G15 which had the same shape with G3 at a different magnitude level. **Algorithm CGR** failed to separate these two groups in all iterations and resulted with 100 percent misclustered genes in G15. With equal weights, nonetheless, the algorithm worked very well with a very low average misclustering rate smaller than 0.01. In that case, the algorithm failed to divide G2 and G4 the most, again, with average misclustering rate smaller than 0.01 and 0.01, respectively.

Finally, the expanded simulated data set was modified in several features in order to make it closer to the real life cases. Three features of this data set was categorized into two levels and 8 scenarios were created as the combinations of the levels of THESE three features. The first feature was the sample size. 300 genes as in this study may not be a suitable projection for real life cases. Therefore, the levels of the first feature was declared as small and big data sets. The sample sizes were directly related to the second feature. The second feature defined the equality of the number of genes within each group. It is likely to have varying number of genes from each profile in real data sets. Hence, the second feature defined whether the number of genes from different profiles were equal to each other. In the first level of this feature, same number of genes were generated for each level. For the small data sets, 20 genes were generated for each group while 50 genes were used for every profiles for big data sets. Next, Discrete Uniform distribution was used to select the number of genes in each group. For small data sets, a random number was selected from 5 to 35 for each group. However, for big data sets a random number between 35 and 65 was generated to state the number of genes for each group. Note that, n_k^m defined the number of genes in group k at the m^{th} iteration in Eq. 5.1. Therefore, n_k^m was distributed with Discrete Uniform with parameters 5 and 35 or 35 and 65 depending on the level of this feature. Finally, the last feature defined the time spaces in time-series. First level stated the equal time points. Six consecutive successive time points, 1^{st}, 2^{nd}, 3^{rd}, 4^{th}, 5^{th} and 6^{th} units were used for this selection. However, the successive follow-up points were selected as 1^{st}, 2^{nd}, 4^{th}, 8^{th}, 16^{th} and 32^{nd} units for the unequal time spaces which was the second level of this third feature.

1000 data sets were generated from each of the 8 scenarios, and **Algorithm CGR** was tested on these data sets with five different weights, 0, 0.25, 0.50, 0.75 and 1. The simulation study resulted in Table 5.2 was the first scenario with a small data set where 20 genes generated for each profile and equal time spaces were used. The same misclustering rates were calculated for the remaining scenarios and the results are given in the Appendix A. The results showed that the algorithm failed to separate more with the big sample sizes than the small sample sizes. When big sample sizes were used, the algorithm especially had challenge to separate the genes in G3 from the constant genes. The second challenge for the algorithm was seen with the unequal time points. The longer time points disappeared the slope information between the time points and led to display them as constant shapes.

Therefore, these cases also became problem for the algorithm to separate the constant genes. Such problems occurred especially when only one of the metrics was used. However, **Algorithm CGR** was able to detect the most of the genes when both of the metrics were included in the algorithm and resulted in very small average misclustering rates. The highest misclustering rate was 0.03 among the simulations with $w = 0.5$. This rate was obtained with the experiments where small and unequal groups sizes were used with unequal time points.

5. 1. 3. Simulation Study 3

As the third and final simulation study a new set of profiles were generated. For this simulation set, three replications were used for each gene. Furthermore, each time-series included 4 follow-ups. For this simulation study 23 different groups of genes were used. Figure 5.12 shows the profiles for the genes in these 23 groups.

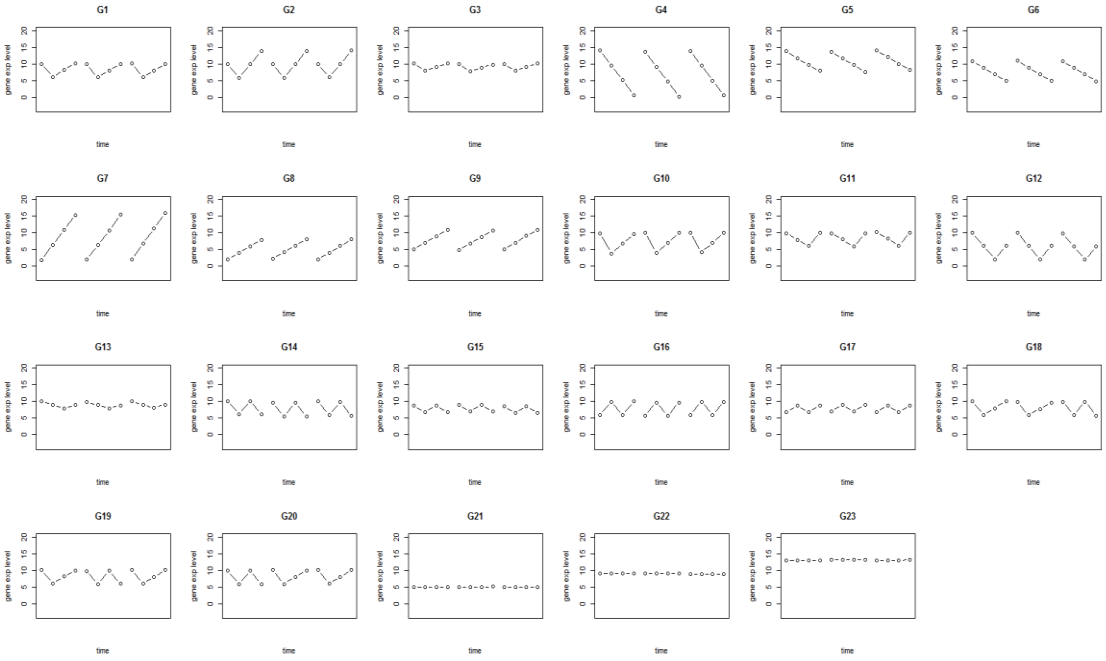


Figure 5.12. Twenty three groups simulated for simulation study 3

The first 3 groups showed a different down-up profile with the lowest expression level at the second time points. Each of these groups had different features. G1 displayed a profile with close first and last expression levels. The genes in G2 expressed themselves at the end higher than the baseline and genes in G3 showed a similar profile as G1 but with a higher minimum value. The groups between 4 and 6 displayed a monotonic decrease pattern where G4 showed a faster decrease than the others. G5 and G6 demonstrated a similar shape with similar slope decreases. However, there were a magnitude difference between them. The next three groups showed monotonic increases and the differences between them were

similar to the dissimilarities between G4, G5 and G6. G10 again displayed a down-up pattern with a lower minimum expression level than G1 at the second time point. Furthermore, the genes in groups G11 to G13 showed down-up patterns with the minima at the second follow-up point. The difference between these three groups were similar to the groups between G1 and G3. The next four groups (from G14 to G17) showed sinusoidal patterns. G14 and G15 started to the profile with a decrease. However the change level in the expression levels were different even though their magnitude levels were very similar. The other sinusoidal groups, G16 and G17, displayed the similar patterns with G14 and G15 except their profiles started with an increase. Next, the groups from 18 to 20 contained the genes with differences between the replicates. Finally, the last three groups included the constant genes. In real data sets, it is very unusual to have constant genes at a single magnitude level as in the previous simulation data set. Therefore, for this simulation study, constant genes were generated at three different magnitude levels.

The simulated data set presented in Figure 5.12 was used to evaluate the success of **Algorithm CGR**. Similar to the previous simulation study, different scenarios were created from this simulated data set by changing five features of it. Those different features were equality of group sizes, number of genes per groups, rate of genes with variations among the replications, rate of constant genes, and lengths of the time spaces. A summary of the levels of these parameters and their abbreviations are seen in Table 5.3.

Table 5.3. Different conditions for the simulation study

Feature	Levels	Abbreviations
group size	equal	ES
	unequal	US
number of genes per group	~20	L
	~50	H
rate of genes with different replications	same	VS
	decreased	VD
rate of constant genes	same	CS
	expanded	CE
time space	equal	ET
	unequal	UT

First one of these features states the equality of the number of genes in different profiles. To test the algorithm, equal number of genes for each profile was used in half of the simulations. On the other hand, random number of genes for each profile was used for the remaining simulations. The second feature showed the sizes of the profiles. The algorithm

proposed in this study was tested on both small and big data sets. For small data sets, 20 time-series were generated for each profile in the equal group size case, while 50 time-series were used for the big data sets. With the simulations which have unequal group sizes, a random discrete number was generated between 10 and 30 for the small data sets. On the other side, a discrete random number was generated between 40 and 60 for the big data sets with unequal group sizes. Next, different scenarios were considered for the rate of genes with variations among the replicates. Such genes may exist in data sets rarely, which create a challenge to detect them. Due to this challenge, in half of the studies, the total number of genes with variations among the replicates was decreased to a value, specifically 10 % of the total number of genes without any differences among the replications. Furthermore, two different cases on the number of constant genes were tested during the simulations. As aforementioned, the number of constant genes might be high in a real data set. For half of the simulations, the same rules for the second feature were used to declare the number of genes in constant groups. However, the number of constant genes was expanded to the total number of genes in the non-constant groups which increases the data set sizes twice. Finally, two cases on the time spaces, equal and unequally spaced time points, were tested during simulations. Half of the studies were tested on equal time spaces with 1^{st}, 2^{nd}, 3^{rd} and 4^{th} unit time points, whereas others were tested by using unequal time spaces with 1^{st}, 4^{th}, 16^{th} and 48^{th} unit time points.

From the five two-level features, 32 different scenarios were created to measure the success of **Algorithm CGR**. These different scenarios were shortened by using the abbreviations of the features stated in Table 5.3. For example, "ES_L_VS_CS_ET" defines the simulation study where each group has exactly 20 time-series, including the genes with variations among the replications and constant genes, with equal time spaces. For simplicity, those scenarios were shown with another notation "S#". The order of the scenarios and their explanations can be seen in Table 5.4.

In this simulation study, **Algorithm CGR** was tested on these 32 scenarios. 1000 data sets generated for each of the 32 scenarios, and several accuracy measures and computational times were calculated for different situations. For each scenario, minimum, maximum, average and standard deviation of computational times were collected over 1000 iterations. Moreover, misclustering rates on each group and average of them over 1000 iterations were calculated. Finally, two accuracy measure studies were held on these simulations. The first accuracy measure study showed the accuracy of detecting the constant genes. This study was hold to see whether the algorithm could detect the constant genes among the data set. As noted before, if the algorithm is able to assign the constant genes into separate clusters, it would rule out the need for a filtering on the constant genes. The second accuracy measure study showed these accuracy values while exposing the genes with variations among the replicates. This study tested the ability of identifying the genes with different replicates.

Table 5.4. Scenario orders and explanations of abbreviations of the scenarios

Scenario Number	Scenarios with Abbreviations	Group Size	Number of Genes	Rate of Genes with Variation in Replicates	Rate of Constant Genes	Time Spaces
S1	ES_L_VS_CS_ET	Equal	20	Same	Same	Equally Spaced
S2	ES_L_VS_CE_ET	Equal	20	Same	Expanded	Equally Spaced
S3	ES_L_VD_CS_ET	Equal	20	Decreased	Same	Equally Spaced
S4	ES_L_VD_CE_ET	Equal	20	Decreased	Expanded	Equally Spaced
S5	US_L_VS_CS_ET	Unequal	Disc. Unif. (10, 30)	Same	Same	Equally Spaced
S6	US_L_VS_CE_ET	Unequal	Disc. Unif. (10, 30)	Same	Expanded	Equally Spaced
S7	US_L_VD_CS_ET	Unequal	Disc. Unif. (10, 30)	Decreased	Same	Equally Spaced
S8	US_L_VD_CE_ET	Unequal	Disc. Unif. (10, 30)	Decreased	Expanded	Equally Spaced
S9	ES_L_VS_CS_UT	Equal	20	Same	Same	Unequally Spaced
S10	ES_L_VS_CE_UT	Equal	20	Same	Expanded	Unequally Spaced
S11	ES_L_VD_CS_UT	Equal	20	Decreased	Same	Unequally Spaced
S12	ES_L_VD_CE_UT	Equal	20	Decreased	Expanded	Unequally Spaced
S13	US_L_VS_CS_UT	Unequal	Disc. Unif. (10, 30)	Same	Same	Unequally Spaced
S14	US_L_VS_CE_UT	Unequal	Disc. Unif. (10, 30)	Same	Expanded	Unequally Spaced
S15	US_L_VD_CS_UT	Unequal	Disc. Unif. (10, 30)	Decreased	Same	Unequally Spaced
S16	US_L_VD_CE_UT	Unequal	Disc. Unif. (10, 30)	Decreased	Expanded	Unequally Spaced
S17	ES_H_VS_CS_ET	Equal	50	Same	Same	Equally Spaced
S18	ES_H_VS_CE_ET	Equal	50	Same	Expanded	Equally Spaced
S19	ES_H_VD_CS_ET	Equal	50	Decreased	Same	Equally Spaced
S20	ES_H_VD_CE_ET	Equal	50	Decreased	Expanded	Equally Spaced
S21	US_H_VS_CS_ET	Unequal	Disc. Unif. (40, 60)	Same	Same	Equally Spaced
S22	US_H_VS_CE_ET	Unequal	Disc. Unif. (40, 60)	Same	Expanded	Equally Spaced
S23	US_H_VD_CS_ET	Unequal	Disc. Unif. (40, 60)	Decreased	Same	Equally Spaced
S24	US_H_VD_CE_ET	Unequal	Disc. Unif. (40, 60)	Decreased	Expanded	Equally Spaced
S25	ES_H_VS_CS_UT	Equal	50	Same	Same	Unequally Spaced
S26	ES_H_VS_CE_UT	Equal	50	Same	Expanded	Unequally Spaced
S27	ES_H_VD_CS_UT	Equal	50	Decreased	Same	Unequally Spaced
S28	ES_H_VD_CE_UT	Equal	50	Decreased	Expanded	Unequally Spaced
S29	US_H_VS_CS_UT	Unequal	Disc. Unif. (40, 60)	Same	Same	Unequally Spaced
S30	US_H_VS_CE_UT	Unequal	Disc. Unif. (40, 60)	Same	Expanded	Unequally Spaced
S31	US_H_VD_CS_UT	Unequal	Disc. Unif. (40, 60)	Decreased	Same	Unequally Spaced
S32	US_H_VD_CE_UT	Unequal	Disc. Unif. (40, 60)	Decreased	Expanded	Unequally Spaced

For all of these simulations, five different weight pairs were used for the two metrics used in the clustering algorithm. Those five weights were 0, 0.25, 0.50, 0.75 and 1 for the Euclidean distance and their complementary to 1 for the slope distance. In order to show these five

weight levels show in the tables, weights were added to the shortened scenario notations. This approach is shown in Table 5.5.

Table 5.5. Shortened scenario notations with weight selections

Weight	Scenario Notation
w = 0	S#.1
w = 0.25	S#.2
w = 0.50	S#.3
w = 0.75	S#.4
w = 1	S#.5

According to Table 5.5, for example, S1.1 shows the results of the clustering algorithm with 0 weight assigned to Euclidean distance metric on "ES_L_VS_CS_ET" simulation set in the following tables. The first table (see Table 5.6) shows the results of the clustering times on the 32 scenarios over 1000 iterations separately for each weight selection. The results showed that computational times did not vary depending on different weight selections. Computations took the longest time with S22.1. It would be expected, since the number of sample size took the highest values in 22nd scenario. Furthermore, clustering the genes with expanded number of constant genes and big data sets took more time to reach the dendrogram. On the other hand, the minimum computational time was seen on a trial with S15.3 which took 0.15 seconds to reach the dendrogram. Furthermore, the computational time did not vary between the experiments with equal or unequal time spaces. Figure 5.13 displays the mean computational time for each of the 32 scenarios. It shows that after S17, which contained high sample sizes, the algorithm started to took more time.

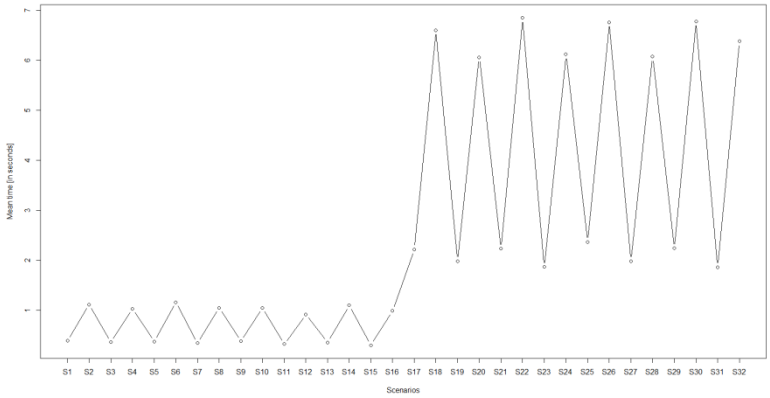


Figure 5.13. Mean computational times under 32 scenarios

Table 5.6. Clustering times over 1000 iterations

	* = 1				* = 2				* = 3				* = 4				* = 5			
Scenario	min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std
S1.*	0.33	0.55	0.39	0.04	0.31	0.57	0.39	0.05	0.31	0.57	0.38	0.05	0.31	0.62	0.39	0.05	0.30	0.59	0.40	0.04
S2.*	1.05	1.32	1.12	0.03	1.03	1.32	1.12	0.03	1.07	1.37	1.11	0.03	1.06	1.37	1.11	0.04	1.07	1.45	1.12	0.04
S3.*	0.29	0.51	0.37	0.03	0.26	0.48	0.36	0.03	0.28	0.51	0.36	0.03	0.28	0.51	0.36	0.03	0.28	0.53	0.37	0.03
S4.*	0.90	1.51	1.03	0.12	0.90	1.50	1.03	0.12	0.87	1.49	1.03	0.12	0.90	1.67	1.03	0.12	0.90	2.86	1.03	0.13
S5.*	0.20	0.66	0.38	0.07	0.21	0.61	0.38	0.06	0.22	0.65	0.37	0.06	0.21	0.54	0.37	0.06	0.22	0.56	0.38	0.06
S6.*	0.48	2.50	1.17	0.27	0.53	2.29	1.16	0.26	0.52	2.91	1.16	0.26	0.50	2.23	1.16	0.25	0.52	1.99	1.16	0.25
S7.*	0.18	0.56	0.34	0.07	0.19	0.52	0.34	0.07	0.19	0.87	0.35	0.07	0.18	0.53	0.34	0.06	0.19	0.50	0.35	0.06
S8.*	0.50	2.06	1.06	0.25	0.48	1.93	1.04	0.24	0.49	2.12	1.04	0.24	0.50	1.92	1.04	0.23	0.51	7.56	1.05	0.31
S9.*	0.31	0.84	0.39	0.03	0.30	0.86	0.38	0.04	0.31	0.77	0.38	0.03	0.31	0.67	0.38	0.03	0.31	0.73	0.39	0.03
S10.*	1.00	1.63	1.05	0.04	0.98	1.61	1.05	0.04	1.00	1.62	1.04	0.04	1.00	1.64	1.04	0.04	0.98	1.60	1.05	0.04
S11.*	0.27	0.49	0.33	0.02	0.27	0.59	0.32	0.02	0.26	0.73	0.31	0.03	0.27	0.66	0.32	0.03	0.26	0.67	0.34	0.02
S12.*	0.87	0.97	0.92	0.02	0.87	0.97	0.92	0.01	0.86	0.96	0.92	0.01	0.88	0.96	0.92	0.01	0.88	1.02	0.91	0.01
S13.*	0.19	0.53	0.36	0.06	0.19	0.53	0.36	0.06	0.20	0.55	0.36	0.06	0.20	0.54	0.36	0.06	0.21	0.53	0.36	0.05
S14.*	0.47	3.02	1.11	0.25	0.50	1.91	1.10	0.23	0.51	2.60	1.10	0.23	0.53	1.80	1.09	0.22	0.53	1.81	1.10	0.22
S15.*	0.17	0.47	0.30	0.05	0.17	0.46	0.30	0.05	0.15	0.47	0.30	0.05	0.18	0.46	0.30	0.05	0.17	0.44	0.30	0.05
S16.*	0.45	1.78	0.99	0.22	0.45	1.67	0.98	0.21	0.46	2.14	0.98	0.21	0.45	3.65	0.99	0.22	0.45	2.41	0.99	0.21

Table 5.6 (cont'd). Clustering times over 1000 iterations

	* = 1				* = 2				* = 3				* = 4				* = 5			
Scenario	min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std
S17.*	2.15	3.37	2.22	0.12	2.15	3.65	2.22	0.12	2.15	3.18	2.22	0.11	2.12	4.01	2.24	0.12	2.14	3.18	2.19	0.11
S18.*	6.41	10.83	6.61	0.45	6.39	10.90	6.61	0.46	6.39	10.76	6.58	0.44	6.39	10.33	6.59	0.45	6.42	10.84	6.61	0.45
S19.*	1.92	2.10	1.99	0.03	1.91	2.09	1.99	0.03	1.93	2.08	1.98	0.04	1.93	2.11	1.98	0.04	1.91	2.08	1.98	0.04
S20.*	5.67	9.03	6.06	0.38	5.65	8.68	6.06	0.37	5.66	10.88	6.06	0.40	5.68	11.45	6.08	0.46	5.69	10.69	6.04	0.39
S21.*	1.89	2.62	2.24	0.13	1.89	2.64	2.23	0.12	1.89	2.59	2.23	0.11	1.87	2.61	2.23	0.12	1.84	2.62	2.23	0.12
S22.*	5.29	22.31	6.88	0.75	5.44	10.66	6.85	0.56	5.35	10.59	6.84	0.55	5.37	10.59	6.84	0.55	5.43	10.58	6.84	0.55
S23.*	1.52	2.23	1.87	0.12	1.52	2.20	1.87	0.12	1.52	2.25	1.86	0.11	1.56	2.21	1.87	0.11	1.53	2.21	1.86	0.11
S24.*	4.95	10.52	6.14	0.54	4.96	10.55	6.13	0.53	4.96	10.55	6.12	0.53	4.96	10.58	6.12	0.53	5.01	10.50	6.11	0.52
S25.*	2.15	3.88	2.37	0.35	2.15	4.62	2.37	0.36	2.15	4.86	2.37	0.36	2.14	4.59	2.40	0.38	2.14	3.74	2.33	0.34
S26.*	6.34	9.27	6.78	0.44	6.37	9.75	6.74	0.46	6.37	9.51	6.77	0.46	6.33	9.47	6.76	0.45	6.34	9.86	6.76	0.46
S27.*	1.92	3.11	2.01	0.10	1.90	2.92	2.00	0.09	1.88	3.54	1.98	0.11	1.88	2.84	1.97	0.11	1.87	2.98	1.97	0.10
S28.*	5.72	8.46	6.08	0.39	5.70	9.90	6.08	0.39	5.71	8.52	6.08	0.39	5.71	8.40	6.08	0.39	5.71	8.47	6.07	0.39
S29.*	1.89	2.72	2.25	0.13	1.87	2.68	2.24	0.12	1.82	2.72	2.24	0.12	1.88	2.70	2.24	0.12	1.89	2.64	2.24	0.12
S30.*	5.26	10.52	6.79	0.57	5.29	10.44	6.78	0.55	5.30	10.43	6.78	0.55	5.35	10.53	6.78	0.55	5.29	10.48	6.77	0.55
S31.*	1.52	2.41	1.87	0.13	1.54	2.33	1.86	0.13	1.54	2.49	1.86	0.13	1.54	2.25	1.86	0.12	1.54	2.24	1.86	0.12
S32.*	4.94	11.42	6.39	0.66	4.90	10.81	6.39	0.66	4.97	10.64	6.39	0.66	4.96	10.61	6.38	0.65	4.97	11.31	6.37	0.65

Table 5.7 shows the overall misclustering rates for each scenario with five weight selections. The average number of misclustered genes were again calculated with Equations 5.1 and 5.2.

Table 5.7. Average rate of misclustering

	* = 1	* = 2	* = 3	* = 4	* = 5
S1.*	0.17	0.00	0.00	0.00	0.03
S2.*	0.17	0.00	0.00	0.00	0.01
S3.*	0.17	0.00	0.00	0.00	0.03
S4.*	0.17	0.00	0.00	0.00	0.01
S5.*	0.15	0.00	0.00	0.00	0.03
S6.*	0.34	0.00	0.00	0.01	0.05
S7.*	0.16	0.00	0.00	0.00	0.03
S8.*	0.36	0.00	0.00	0.01	0.05
S9.*	0.18	0.00	0.00	0.00	0.03
S10.*	0.18	0.00	0.00	0.00	0.01
S11.*	0.18	0.00	0.00	0.00	0.03
S12.*	0.18	0.00	0.00	0.00	0.01
S13.*	0.16	0.00	0.00	0.00	0.03
S14.*	0.35	0.00	0.01	0.02	0.05
S15.*	0.18	0.01	0.00	0.01	0.03
S16.*	0.37	0.01	0.01	0.02	0.05
S17.*	0.17	0.00	0.00	0.00	0.00
S18.*	0.17	0.00	0.00	0.00	0.00
S19.*	0.17	0.00	0.00	0.00	0.00
S20.*	0.17	0.00	0.00	0.00	0.00
S21.*	0.16	0.00	0.00	0.00	0.01
S22.*	0.37	0.00	0.00	0.00	0.05
S23.*	0.18	0.00	0.00	0.00	0.01
S24.*	0.39	0.00	0.00	0.00	0.04
S25.*	0.17	0.00	0.00	0.00	0.00
S26.*	0.52	0.00	0.00	0.00	0.00
S27.*	0.52	0.00	0.00	0.00	0.00
S28.*	0.52	0.00	0.00	0.00	0.00
S29.*	0.51	0.00	0.00	0.00	0.01
S30.*	0.56	0.00	0.00	0.02	0.05
S31.*	0.46	0.01	0.00	0.00	0.01
S32.*	0.54	0.02	0.02	0.02	0.04

The detailed tables which also show the number of misclustered genes for each group separately are given in the Appendix C. Table 5.6 showed that the highest misclustering rates were seen when only the shape metric was used. However, the reason for this was the existence of constant genes in different levels. Since there was no difference in shape between those constant genes, they were grouped together in clusters no matter what their magnitude levels were which exaggerated the misclustering rate. Further, it was seen that, increasing the sample size slightly increased the misclustering rate. The average misclustering rate was 0.074 for big data sets while it was 0.051 for small data sets. Moreover, the unequal time spaces led to higher misclustering rates. This appeared especially with unequal group sizes. Previous simulation studies showed that using only one of the metrics may fail clustering the time-series. These results suggested that both metrics should be included in the algorithm. Among all the scenarios, when both metrics were included in the algorithm, the highest misclustering rate was calculated as 0.02. This concluded that using **Algorithm CGR** was a useful methodology to cluster the time-series when both metrics were included. Finally, increasing the rate of constant genes in unequally sized groups increased the misclustering rate for the cases where only the squared Euclidean distance was used.

The ability of **Algorithm CGR** to separate the constant genes and the genes with differences among their replications from the rest of the genes were evaluated with two further analysis. In the first analysis, the data were accepted as two groups which contained the constant and non-constant genes. Therefore the first group included G21 to G23 whereas the second group had the remaining profiles presented in Figure 5.12. In order to test the accuracy of **Algorithm CGR** on dividing the constant genes from the others, several accuracy measures were also calculated on every 32 scenarios with 5 weight levels. The formulations of these accuracy measures are given in Appendix B.

Table 5.8. Results of the accuracy measures for detecting the constant genes for the first scenario

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S1.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.5	0.98	0.99	0.90	0.10	0.01	0.98	0.96

The accuracy measures showed that **Algorithm CGR** was very successful on separating the constant genes. Most of the accuracy measures gave perfect results for all seven accuracy measures. The most failures were obtained when only the squared Euclidean distance was

used. In most of the scenarios, the accuracy measures showed that there were little departures from the perfect results with the selection of $w = 1$. For exemplary, the results of the first scenario iss given below (see Table 5.8), whereas the whole table is given in the Appendix B. The most dispersion was observed within S3.5 and S11.5. In both situations the rate of constant genes was not expanded and the rate of genes with dissimilar replicates was decreased. In those conditions, specificity was calculated as 0.89 which was smaller compared to the most of the other scenarios. Furthermore, both scenarios resulted with higher false negative rate than false positive rate. This means that the algorithm failed more by misclustering the constant genes into the non-constant genes when only the distance metrics was used than the otherwise.

The second accuracy measure tests were hold to see the ability of the algorithm in detecting the genes with dissimilar replicates. **Algorithm CGR** seemed to be more successful on detecting such genes. Again, the whole table is given in the Appendix B. The results showed that, in most of the trials the algorithm perfectly separated the genes with dissimilar replicates. Algorithm failed to detect those genes the most when only one of the metrics was used. There were several cases when the algorithm could not catch such genes. S6 was one of the scenarios where the algorithm failed in separating the genes with dissimilar replications (see Table 5.9). It shows that, positive predictive power was very low with which means that the algorithm misclustered the genes with dissimilar replications into the groups of the genes without different replications when the rate of constant genes were expanded.

Table 5.9. Results of the accuracy measures for detecting the genes with dissimilar replicates for the sixth scenario

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S6.1	0.93	0.49	0.97	0.03	0.51	0.55	0.96
S6.2	0.91	0.06	0.99	0.01	0.94	0.49	0.92
S6.3	0.95	0.04	1.00	0.00	0.96	0.49	0.95
S6.4	0.88	0.49	0.94	0.06	0.51	0.55	0.93
S6.5	0.93	0.49	0.97	0.03	0.51	0.55	0.96

5. 1. 4. Real Data Study

A real data set from Tomancak et al. (2002) was used to test the **Algorithm CGR**. The data set contains the expression levels of *Drosophila Melanogaster*. Thirty six Affymetrix Arrays were used in the experiment which included 14010 genes. The experiment was followed every hour for 12 consecutive hours. Therefore there were 12 time points for each time-

series. Moreover, 3 replications were used for each gene. This data set was used in the study of Irigoien et al. (2011). Their aim was to find and cluster the gene expression profiles as well as detecting the genes which show differences among the replications.

As explained before, the methodology proposed by Irigoien et al. (2011) filtered the constant shape genes in the first step. In the next step, 700 genes were found to be differentially expressed. Thus, they continued the analysis with these 700 genes. In the second step, among those DE genes, the ones with differences among the replications were filtered out to be analyzed later. Their study showed that 71 genes displayed variations among their replicates. Thus, the expression profiles were clustered by using the remaining 629 genes.

Algorithm CGR was tested on this data set in two parts. In the first part, it was tested on 629 genes which did not have differences among the replications. The results of this clustering study were compared with the results in Irigoien et al. (2011). Then, in the second part, all the 700 genes were inputted in the clustering algorithm and the algorithm was tested if it can detect the genes with different replications.

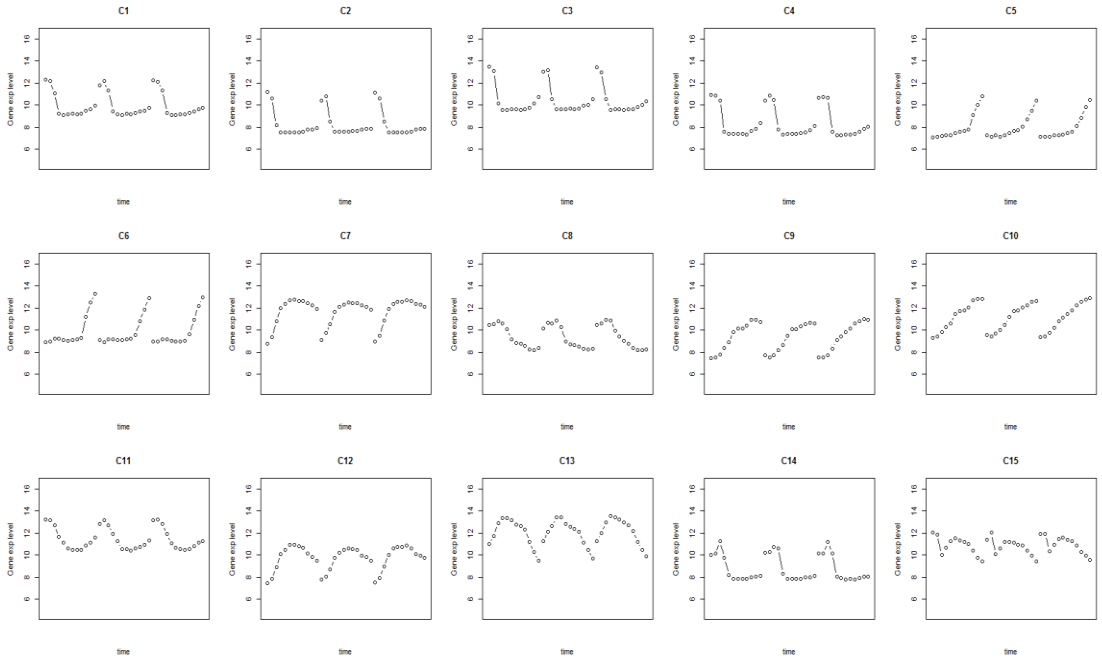


Figure 5.14. Clustering results on the real data for which the genes with different replications were put aside

When Irigoien et al. (2011) studied the 629 genes, they found 15 clusters. These 15 clusters, later, classified in higher levels as “down-constant”, “constant-up”, “up-constant”,

"monotonic decrease", "monotonic increase", "down-up" and "up-down" profiles. Therefore, **Algorithm CGR** was tested on these 629 genes and 15 clusters were searched. While using the clustering to search for the patterns, equal weights were assigned to both distance and shape metrics. Figure 5.14 shows the profiles obtained from the 15 clusters. Each panel in Figure 5.14 shows the average expression levels of the genes in that cluster at each time point. Moreover, the three replications of the genes are represented consecutively, as explained before.

Algorithm CGR detected most of the patterns found by Irigoien et al. (2011) with small differences. For example, when their algorithm found three "down-constant" patterns, **Algorithm CGR** found four such patterns. However, the patterns within those four clusters displayed slight differences. For example, Both of C2 and C3 showed the genes which decrease in their expression levels at the second time interval and kept their levels until the end of the experiment. Although their shape was very similar, there was a magnitude difference between these two profiles. Therefore, the algorithm divided this pattern into two clusters. On the other hand, C1 and C3 had a "down-constant" pattern with close magnitude levels. However, they were separated due to different time intervals that the genes experienced the decrease in the expression level. The genes in C1 decreased their expression levels right before the 4^{th} time point while the genes in C3 had this phenomena before the third time point.

There were also slight differences for other patterns. One of the most significant differences between the two algorithms was observed with the "monotonic decrease" pattern. The algorithm proposed in Irigoien et al. (2011) found a pattern in which the genes experienced a decrease after the 2^{nd} time point. On the other hand, **Algorithm CGR** did not detect such a pattern. However, it detected a profile which displayed a monotonic decrease after the fourth time point and assigned them into C8. Moreover, the algorithm could detect a new pattern which was demonstrated in C15. The genes in that cluster decreased their expression levels until the third time point. Then their expression level increased between the third and sixth time points and then decreased monotonically until the end of the experiments. This pattern could not be detected in the previous study.

It was shown that **Algorithm CGR** worked well and detected the profiles in the data set. Next, the algorithm was tested to evaluate its ability to catch the genes with dissimilar replicates. It was mentioned earlier that 71 genes had variations among their replications. Further, it was found that, some of these genes showed a specific variation. Some of the genes which form "up-constant" pattern experienced a dramatic decrease at the 10^{th} follow-up and then returned back to their previous expression level at the next follow-up. Other than these ones, none of the genes with variations among the replications showed significantly different profiles than the previously found 15 profiles. Hence, each of these genes were assigned to the cluster with the most similar profile. 700 genes were inputted in **Algorithm CGR** to test if it can catch these genes. Again, equal weights were used for both metrics in this study ($w = 0.5$). However, the number of profiles was not known with the addition of 71 genes to the previous 15 clusters. Twenty clusters were used in order to see the profiles within all 700 genes. The results were displayed in Figure 5.15.

Algorithm CGR was successful in dividing the similar profiles with slight differences. For example, it divided two "down-constant" patterns with similar shape and dissimilar magnitude levels into different clusters, C2 and C5. However, the most important result of this experiment was that **Algorithm CGR** was able to separate the genes with differences among the replications. The genes in C9 showed a "up-constant" pattern. Further, the genes in G10 also showed that pattern, however, their second replicates experienced a difference. Their second replicates displayed a variation which was compatible with findings of Irigoien et al. (2011). The second replicates experienced a dive at the 10^{th} time point which returned back to its previous level at the next follow-up. These two clusters showed the success of the algorithm in separating the genes with dissimilar replicates. Furthermore, it could separate these genes without any need of pre-filtering which helps the user to reach to the results in a shorter time. As one of the strongest features of this algorithm, it took very short computational time to obtain the results. Reaching the 20 clusters from the 700 genes took 0.09 seconds in terms of system time and 1.26 seconds in real time.

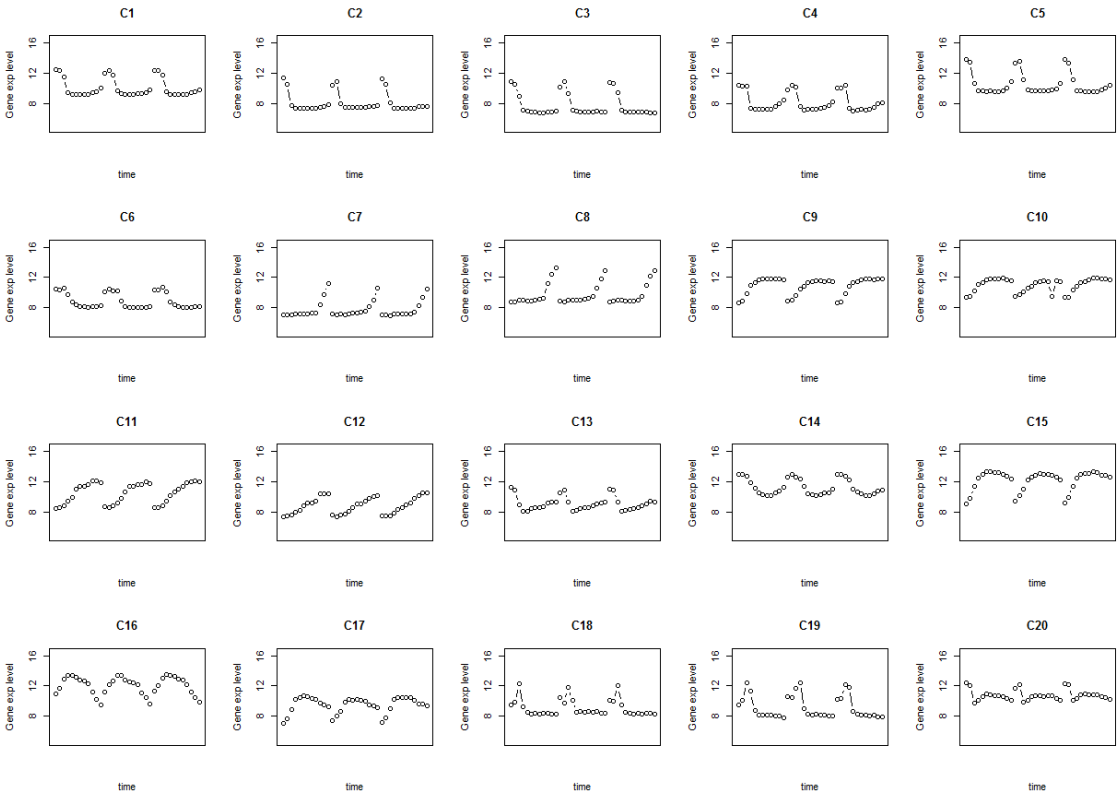


Figure 5.15. Twenty clusters from 700 genes by using both metrics with $w = 0.5$

As consequent, the last two studies showed that **Algorithm CGR** was able to find the profiles among the genes. Furthermore, the second study showed that, it was also able to detect the genes with dissimilar replications and separate them from the others by assigning them into a different cluster. However, as mentioned before, one of the most important problems about clustering studies is deciding on the number of clusters when it is unknown. For example, for the clustering study on all of the 700 genes in this subsection, 20 clusters were used without any prior information. Therefore, some methods should be used to decide on the number of clusters. The second section in this chapter displays the applications of the cluster validation techniques proposed in Section 4. 4.

5. 2. Finding the Number of Clusters

This section displays the results of the cluster validation techniques presented in this thesis. The next two subsections will use two of the simulation sets while the last subsection uses the real data from the previous section.

5. 2. 1. Simulation Study 1

For the first simulation study, the data set generated in Subsection 5. 1. 1. was used. This data set contained 90 hypothetical time-series which should be divided into 9 clusters. There were genes at three different magnitude levels in three different shapes which created nine different patterns. Therefore, the validation score graphs were expected to highlight the validation scores with 9 cluster set. Validation scores could be calculated for different number of clusters. The number of clusters can be changed between one and the number of objects in the data set. The set of clusters with small validation scores or with significant decreases could be selected.

The possible number of clusters for the first simulated data set may be between 1 and 90 since there were 90 time-series totally. However, searching the cluster set in all possible numbers might be inefficient. For example, dividing the data set which contains 90 objects into 60 or 70 clusters may be inappropriate, since it would not reduce the complexity significantly. Therefore, the number of clusters may be searched within a smaller range. For this study, validation scores were calculated for the set of clusters where the number of clusters were between 2 and 40. Figure 5.16 shows the validation scores for all of these cluster sets. Furthermore, validation scores for the cluster sets between 5 and 14 were zoomed in.

Figure 5.15 showed that both validation scores presented very small values at 9 clusters compared to the scores around 9 clusters. Further, there seemed a significant decrease in the validation scores between 8 and 9 clusters. As a consequence, the validation scores gave the hint to use 9 clusters for this data set. In the further cluster numbers, there were also very small validation scores. For example after 27 clusters, low validation scores were observed. However, separating a data set with 90 time-series into 27 clusters may create redundancy. Therefore, 9 clusters could be more useful against the number of clusters higher than 27 clusters.

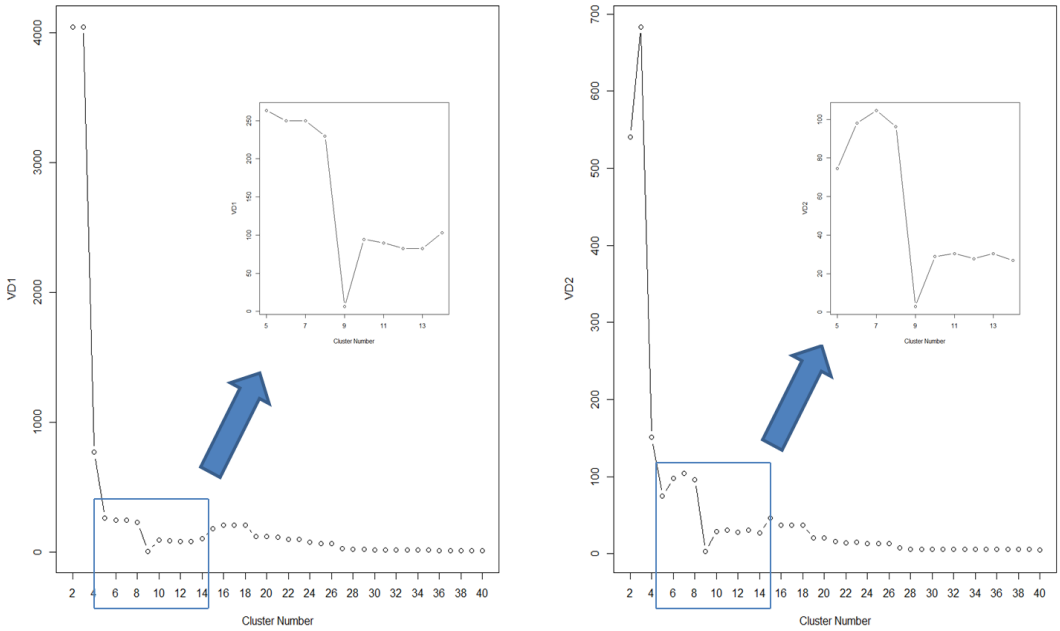


Figure 5.16. Two validation score graphs for the first simulation study

5. 2. 2. Simulation Study 2

In this subsection, these validation scores were tested by using the simulation sets from Subsection 5. 1. 3. This data set contained 23 clusters, therefore, the validation scores were expected to highlight the use of 23 clusters. The genes were followed in four time points and three replications were used for each gene. There were 32 different scenarios generated from this data set. To test the success of the validation scores, the first scenario was used firstly. In this scenario, there were 20 genes generated for each of the 23 groups and equal time spaces were used.

First, the dendrogram was built for the with using both metrics equally in hierarchical clustering. After this dendrogram was built, the validation scores for different set of clusters were calculated. For this simulated data, the correct number of clusters was searched between 7 and 35. The validation score graphs are displayed in Figure 5.17.

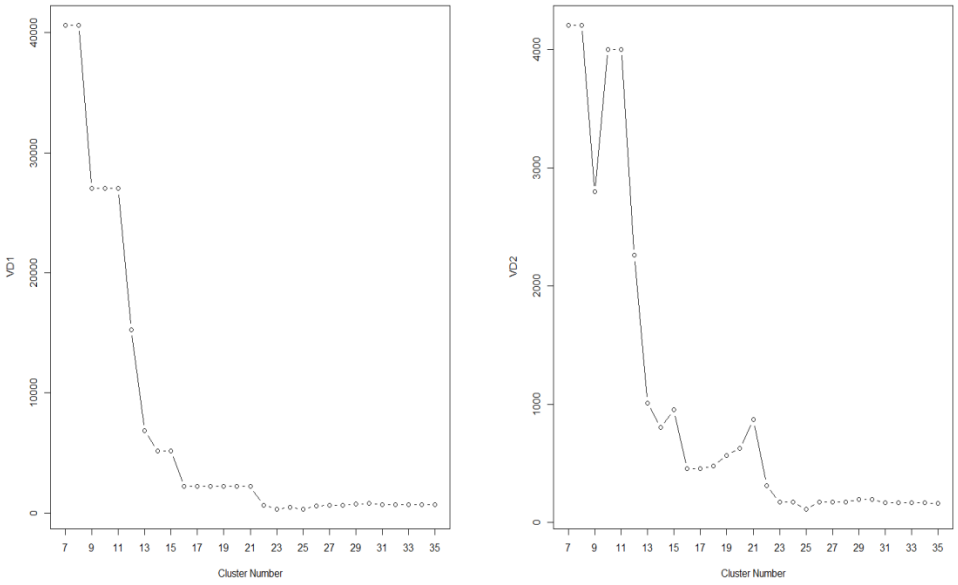


Figure 5.17. Two validation score graphs for the simulated data set

Two validation scores generally showed very small validation scores after 16 clusters. Moreover, there were decreases after 21 cluster in both graphs. *VD2* displayed a decrease between 22 and 23 clusters which was followed by convergence in the validation scores. Finally, the smallest validation score was obtained when the data set was divided into 25 groups in the second graph. Therefore, 23 and 25 cluster sets were used for this data set. The results for 23 clusters are displayed in Figure 5.18.

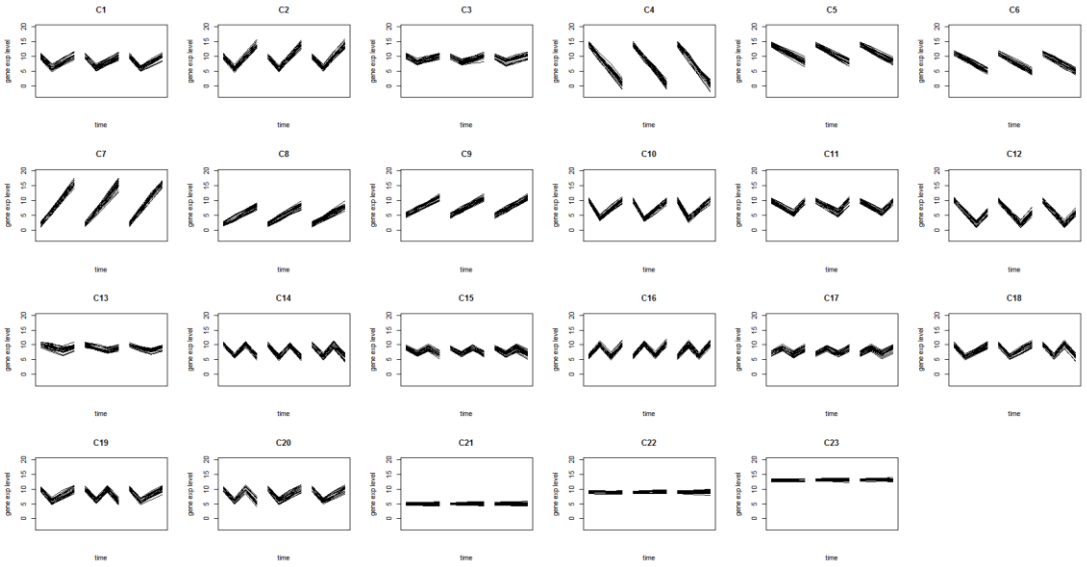


Figure 5.18. Twenty three clusters obtained from the simulated data set

Figure 5.18 showed that when the data set was divided into 23 clusters, the algorithm was able to detect all the profiles in separate groups. It can be seen that the first 17 clusters contained the genes which show different patterns. However, their replicates showed the similar profiles. Further, the clusters between 18 and 20 included the genes with dissimilar replicates. Finally, the last three clusters contained the constant genes at different magnitude levels. This showed that the algorithm successfully detected all the profiles in the data set.

Moreover, *VD2* highlighted that the data set might be divided into 25 clusters, since it gave the smallest validation score. The clusters are shown in Figure 5.19.

When the data set was divided into 25 clusters, the algorithm was again able to show different profiles in different clusters. Each cluster presented a single profile. Two new clusters that were added to Figure 5.18 showed two profiles separately in two groups. The genes in C4 and C5 demonstrated the genes in G4 while the genes in C8 and C9 displayed the profile in G7 from Figure 5.12. However, none of the dissimilar profiles were grouped together into a cluster and all the profiles with dissimilar replicates were separated into different clusters. Finally, the constant genes were also divided into different clusters with respect to their magnitude levels. Although the last set with 25 clusters had redundancy it may not be chosen by the user, while 23 clusters did not have such problem. However, 23 clusters was also highlighted, therefore, the algorithm can lead the user to the most appropriate result.

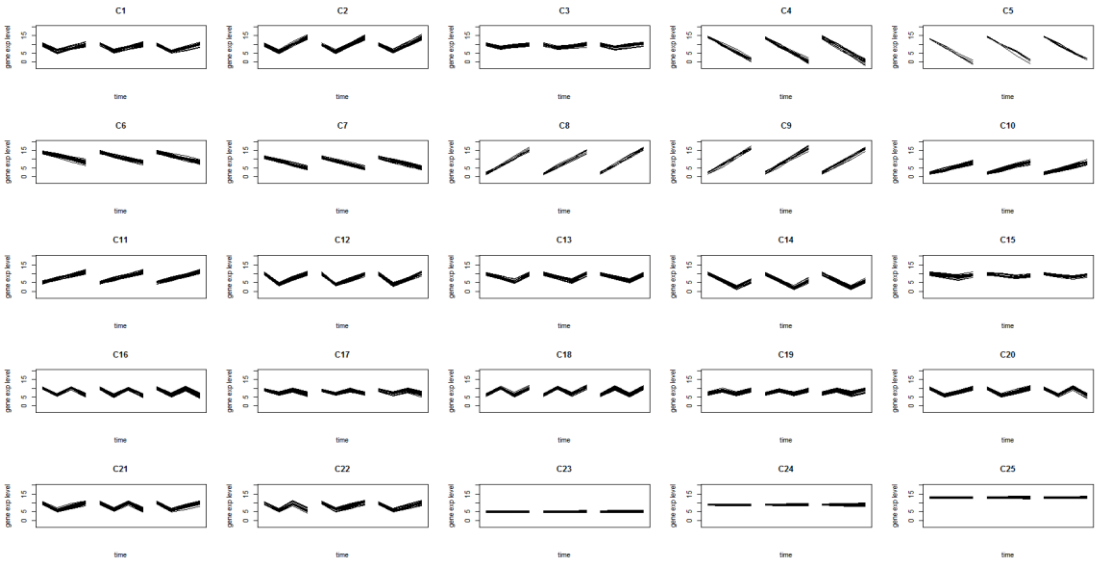


Figure 5.19. Twenty five clusters obtained from the simulated data set

Finally, another scenario was. In real life cases, the number of genes with different profiles would be expected to be different. Furthermore, as noted before, the number of genes with dissimilar replicates would be smaller compared to the number of genes with similar

replicates. Finally, the constant genes may appear at any magnitude level and their number would be very high compared to the other genes. Therefore, S8 explained in Table 5.4 was used with only one modification on the constant genes. Constant genes generated were scattered through all the range of data set. The profiles for this simulation study is shown in Figure 5.20.

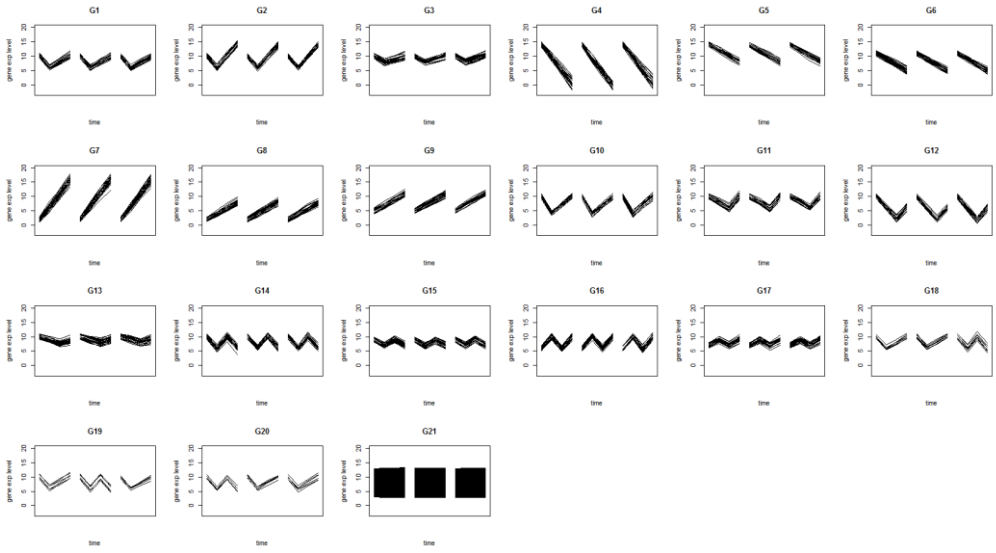


Figure 5.20. Genes generated for the new simulation data

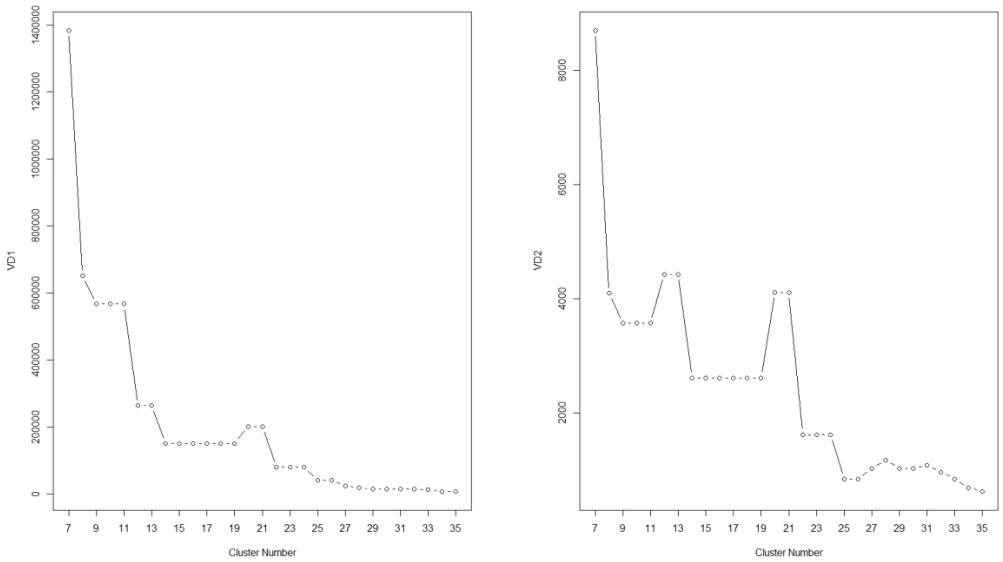


Figure 5.21. The validation score graphs for the data set shown in Figure 5.19

The validation scores from *VD2* in Figure 5.21 displayed small validation scores at cluster numbers 25 and 26. Furthermore, there was a high difference between the validation scores for cluster numbers 24 and 25. According to these observations the data set was divided into 25 clusters. The genes in a clustered form with 25 clusters are displayed in Figure 5.22.

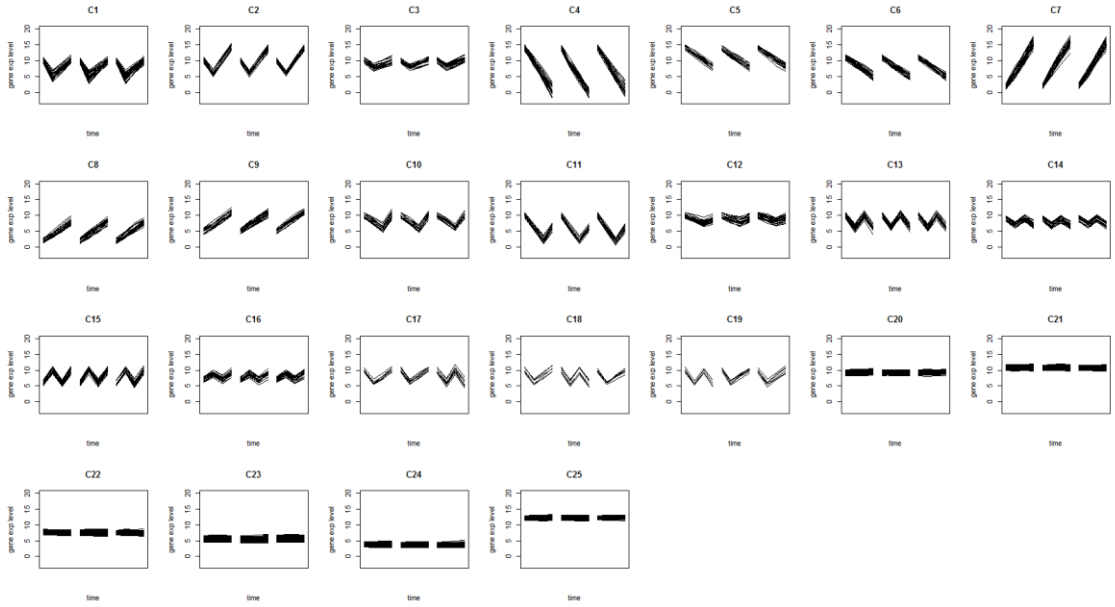


Figure 5.22. Twenty five clusters obtained from the data set in Figure 5.19

Figure 5.22 showed that the genes with different profiles were separated into different clusters. There were only one exception which occurred in C1. When the data set was divided into 25 clusters, C1 contained the genes from G1 and G10 of Figure 5.20. The genes in both of these groups displayed a "down-up" profile with minima at the 2nd time points. The only difference between them was that the minimum level of the genes in G10 was lower than the genes in G1. However, the other patterns were clustered successfully. Furthermore, the genes with different replicates were clustered in separate clusters even though their numbers were very small with respect to the other profiles. Finally, the constant genes were clustered into different clusters. It was mentioned that the constant genes were dispersed around all the range in this data set. Therefore, there was not a clear separation between them. However, the algorithm could assign these constant genes into 6 different clusters with respect to their magnitude levels. However, none of the constant genes were grouped with DE genes.

Since, the validation score graphs also highlighted the 26 number of clusters, finally, the data set was divided into 26 clusters. The results are displayed in Figure 5.23.

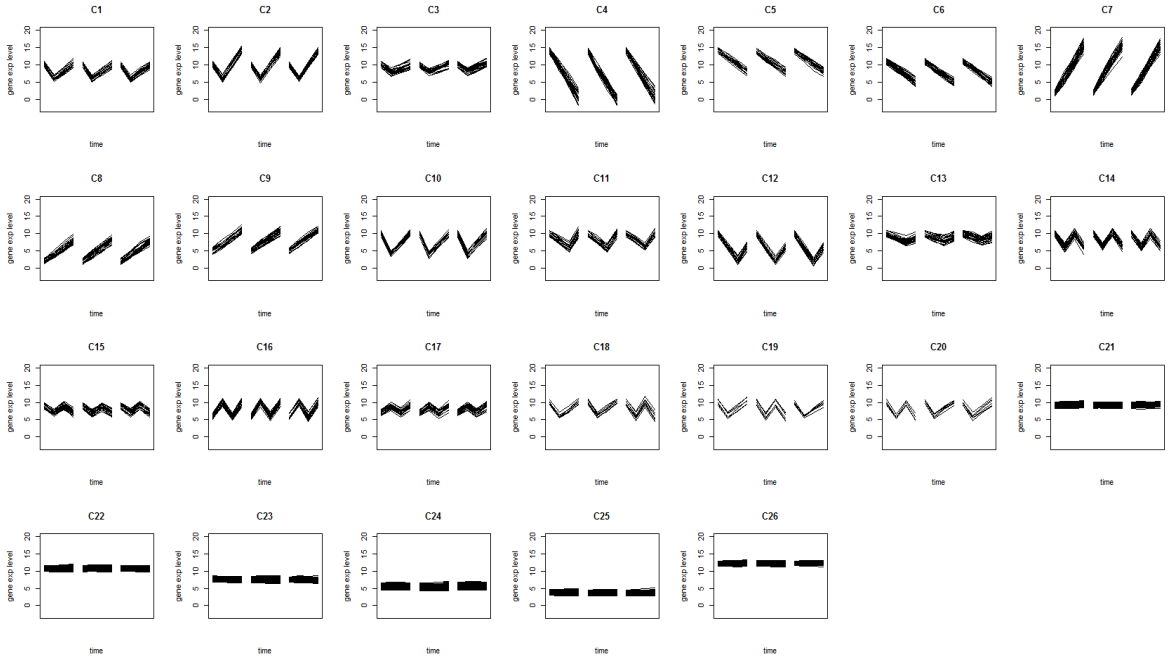


Figure 5.23. Twenty six clusters obtained from the data set displayed in Figure 5.19

When the data set was divided into 26 clusters, each group showed a unique pattern. On the contrary to the previous set, this time the genes from G1 and G10 were separated into different clusters, C1 and C10, respectively. Furthermore, there were not any problems with the genes with dissimilar replicates. They were again separated into different clusters. Finally, the constant genes were divided into 6 different clusters around different magnitude levels.

As consequence, the simulation studies in the last two subsections showed that the validation techniques proposed in this thesis was successful to find the number of clusters in a data set. The last simulation data set showed that these validation scores together with **Algorithm CGR** were able to find the profiles in a data set even with the existence of many constant genes. The algorithm could detect the profiles by assigning the constant patterns into different clusters than the other ones and it could find the number of clusters which should be used for the constant genes. Finally, these cluster validation techniques are used on the real data set used in Subsection 5. 1. 4. The results are displayed in the next subsection.

5. 2. 3. Real Data Study

The real data set in the Subsection 5. 1. 4. contained 700 genes none of which displayed a constant shape. In order to apply the validation techniques on this data set, **Algorithm CGR** was applied with equal weights for both metrics. After that, the correct number of clusters

was searched on sets which had clusters between 6 and 30 (see Figure 5.24). In real data sets, the profiles were not separated from each other as clearly as in the simulation sets. Therefore, in the real data sets, it is very hard to detect a unique number of cluster. Instead of specifying a single cluster number, the validation techniques may highlight several set of clusters on real data sets.

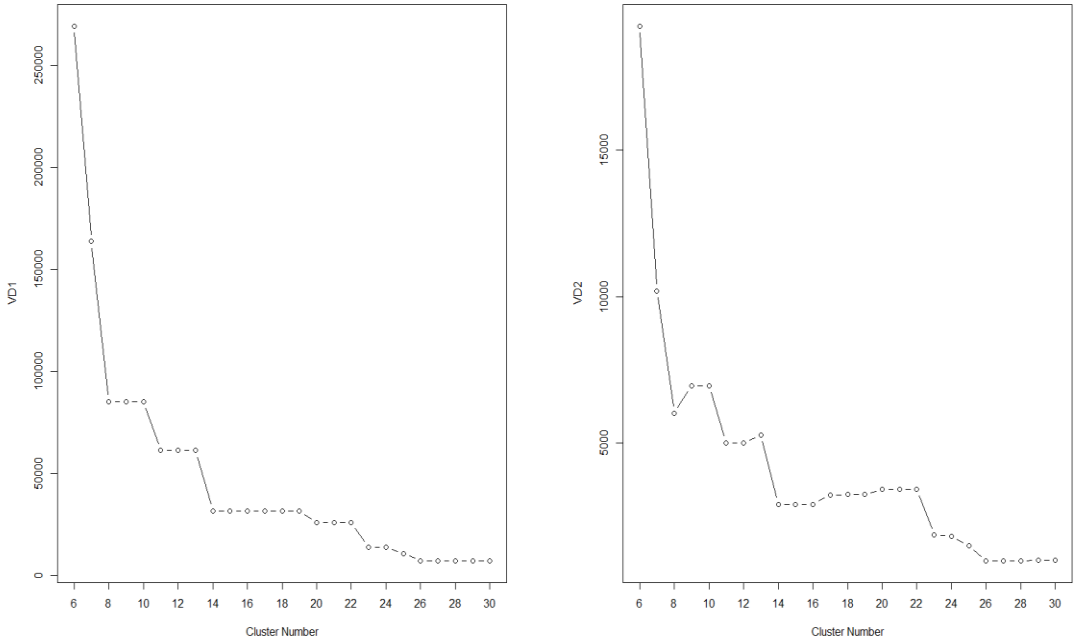


Figure 5.24. Validation graphs on the real data set

The two validation scores in Figure 5.24 experienced decreases at the same cluster numbers. Both graphs suggested 8, 14 and 23 clusters. Therefore, the data set was investigated with these three cases.

Figure 5.25 shows the profiles obtained from 8 clusters by taking the averages of the expression levels of the genes in the clusters. It shows that when the data was divided into 8 groups, all of the profiles could not be detected. For example, the "constant-up" profiles could not be displayed. More importantly, the algorithm failed to divide the genes with dissimilar replications from the others. It was shown before that some of the genes within "up-constant" groups experienced a variation among their second replicates. When the data set was divided into 8 clusters, the algorithm was able to show the "up-constant" profiles, however, it failed to divide the genes with different replicates from the genes with similar replicates. A slight difference, nevertheless, was shown within the second replicates of the genes in C3 which demonstrate "up-constant" pattern. We might suspect that the genes in that cluster may have dissimilar replications. This can be investigated with higher number of clusters. The next set of clusters suggested by the validation scores was 14 clusters. Thus, the data set was divided into 14 clusters in this step. The profiles are shown in Figure 5.26.

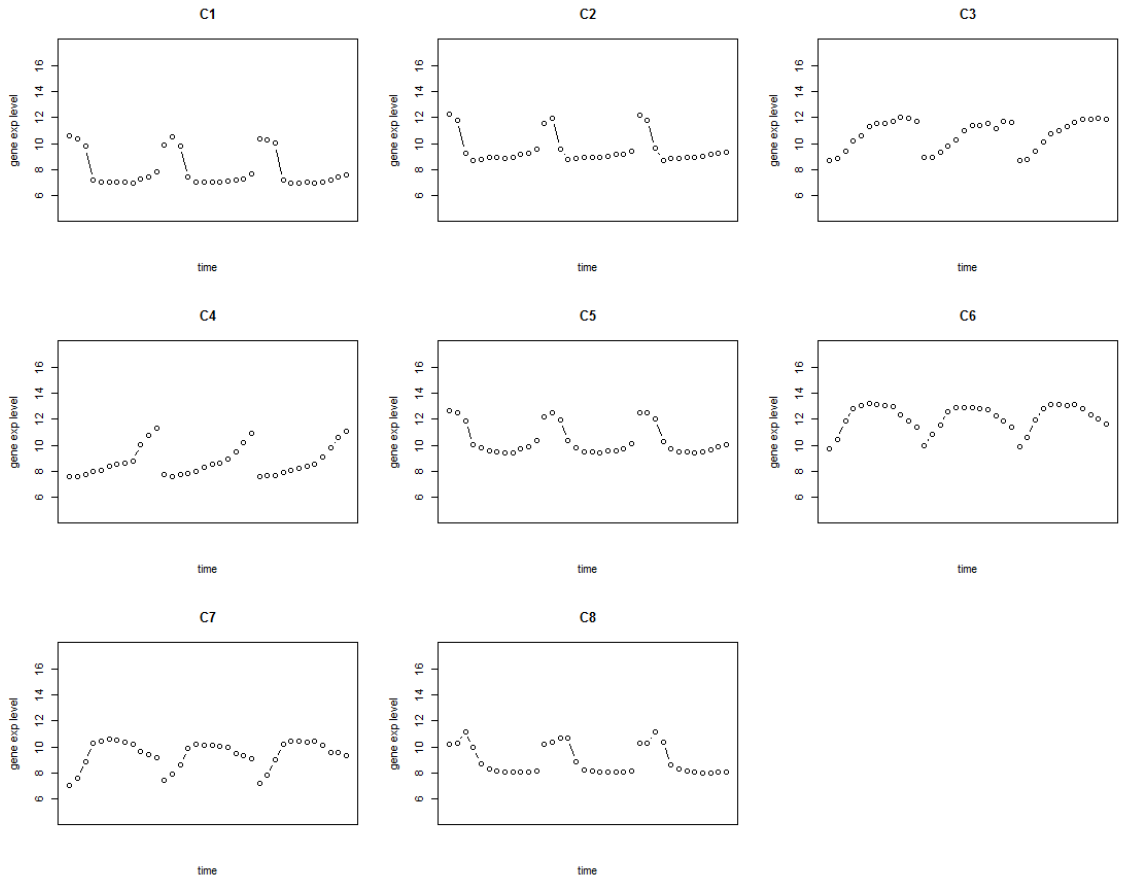


Figure 5.25. Eight clusters obtained from the real data set

When the data set was divided into 14 groups, the clusters showed the different profiles which can not be seen with the 8 clusters. For example, C5 showed the “constant-up” profile which was not seen within the 8 cluster set. Furthermore, the algorithm was able to divide the genes with dissimilar replicates from the others this time. Both of C7 and C8 represented the “up-constant” patterns. However, the genes in C8 showed a variation among their second replicates. **Algorithm CGR** was successful in dividing these two groups of genes into different clusters. Moreover, it detected several profiles which was stated to be in the data set. Finally, the validation scores highlighted that the data set might be divided into 23 groups. The profiles obtained from 23 groups are shown in Figure 5.27.

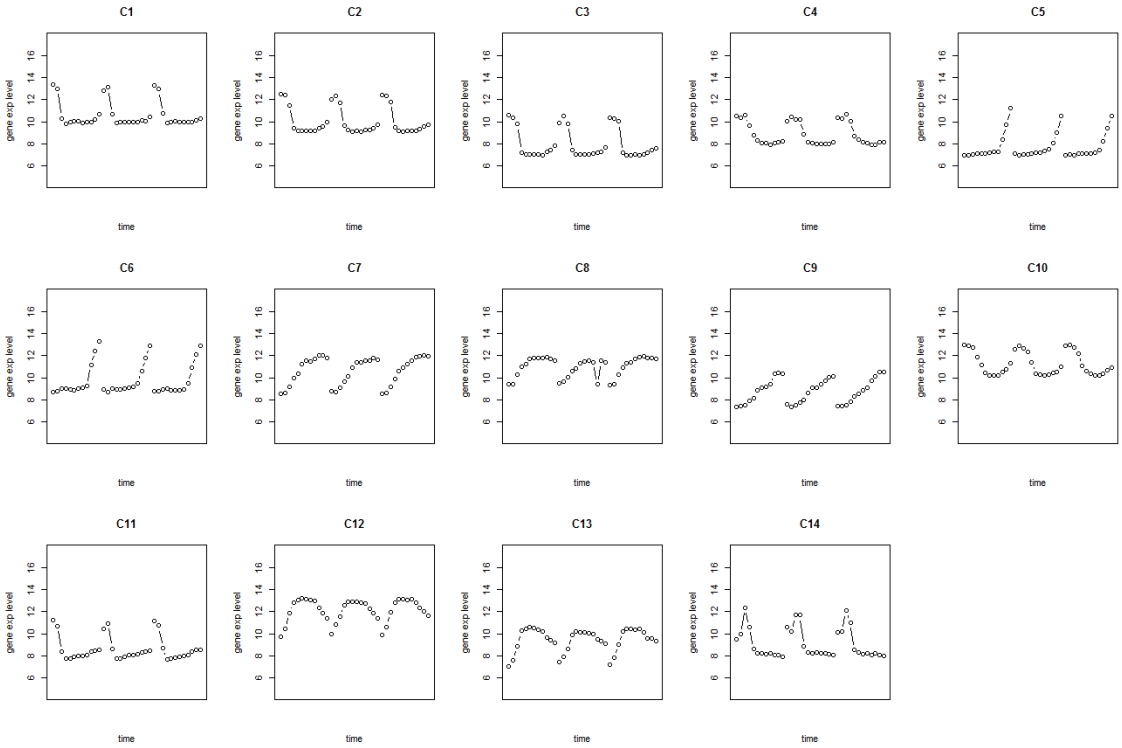


Figure 5.26. Fourteen clusters obtained from the real data set

When the data set was divided into 23 groups, all of the profiles mentioned in Subsection 5. 1. 4. were detected. Moreover, the algorithm was again able to separate the genes with dissimilar replicates from the others. C9 and C10 represents the “constant-up” profile genes with similar and dissimilar replicates, respectively. Next, the “monotonic decrease” genes which were not detected in the previous examples could be revealed with this set of clusters. Finally, some profiles were able to be investigated in more details. For example, there were slight differences between the “up-down” profiles shown between C18 and C20. The genes in C18 showed that pattern in a more curve shape whereas the genes in C20 had a clear peak at the 4^{th} time point.

As a result, the cluster validation techniques suggests several possible set of clusters in a real data set. The algorithm can suggest some small number of clusters which provide a more general display of the profiles in the data set as in 8 cluster set in this subsection. However, this may result in not being able to see the genes with different replicates. On the other hand, the algorithm may also suggest some large number of clusters. With this case, the user had the chance to see the profiles in more details besides being able to see the profiles with differences among the replications.

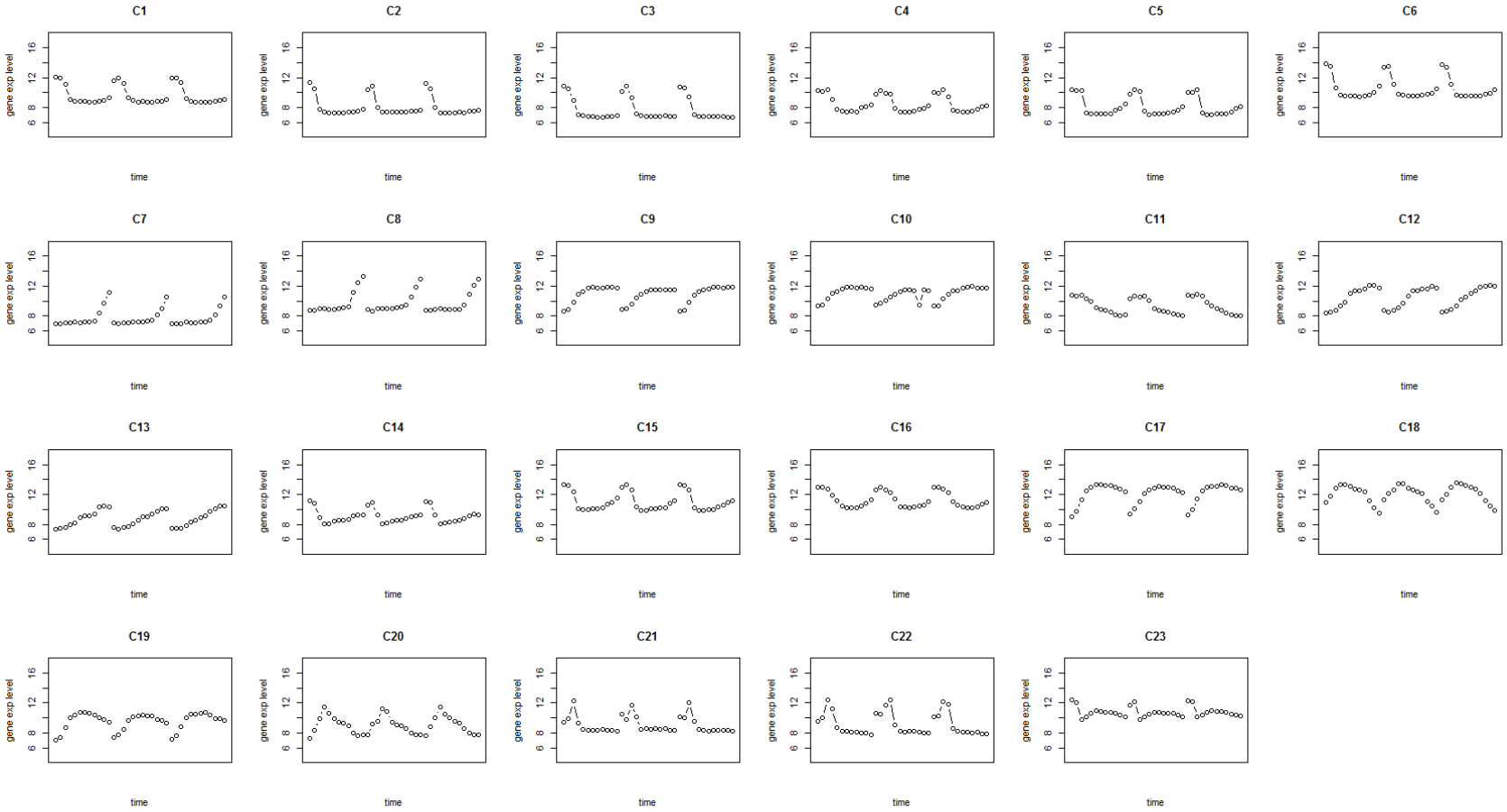


Figure 5.27. Twenty three clusters obtained from the real data set

CHAPTER 6

CONCLUSION

This thesis proposed a methodology to cluster the time-course gene expression profiles with dissimilar replicates. As mentioned in previous chapters, there are several model based clustering algorithms. However, there are several disadvantages of these methods such as their dependencies on sufficient time points or distributional assumptions. Therefore, the methodology proposed in this study is a non-model based clustering algorithm.

An important problem on clustering methodologies on the time-series is that the distance measures they use calculates the dissimilarities based on only one characteristic. For example, the metrics such as Euclidean distance considers only the magnitude differences between the profiles. Furthermore, such metrics use every observation in different time points separately which ignores the time dependencies. On the other hand, the shape based metrics such as Short Time Series or Pearson correlation distances consider only the similarities between the shapes of the profiles. In this study, two metrics which measure dissimilarities based on two different characteristics were emerged together to define the distances between the time-series.

The differences among the profiles of the replicates of the genes may carry important information for the biologists. Thus, detecting such genes is very beneficial for biological purposes. However, highlighting such genes may involve computational burden. A simple but novel method was proposed in this thesis to identify those genes. For this method, replications of each gene were joined consecutively in order to reflect possible dissimilarities among the replications of a gene. Next, a hierarchical clustering approach was applied to the genes by using information from both magnitude and shape characteristics of all replicates of the genes.

Studies on several simulations and a real data set showed that the algorithm was able to detect the profiles in a time-course gene expression data set in a very short time. Furthermore, accuracy measures showed that the algorithm could detect the genes with different replicates successively. Moreover, it was shown by the simulation studies, the algorithm was also able to detect the constant shape genes. High number of constant shape genes are observed in real data sets, which make the analysis very challenging. However, **Algorithm CGR** was shown to detect these constant shape genes in separate clusters, and provides the user flexibility of filtering them easily.

The algorithm failed in some cases, nonetheless. The simulations exposed that the algorithm may fail to recognize the particular profiles with the unequal time spaces. It especially had problems when the time lengths were too big, since the slope information started to disappear in such cases. In those cases, specificity value decreased until 0.90 which was still an acceptable result. Further, it may take longer computational time with bigger sample sizes to reach the results. However, the longest computational time observed was 22.31 seconds among the simulation studies.

Finally, two approaches were proposed to detect the number of clusters in a data set when it is not known a priori. Two simulation studies showed that these approaches were useful to find the correct number of clusters in the data set. Moreover, the study on the real data set revealed that the approaches led to several set of clusters which might be useful. Different number of clusters suggested by our approaches might be evaluated to investigate the data set with several degrees of details.

There are several features of the algorithm proposed in this study to be extended in future studies. First, a different approach can be proposed to handle the replications by keeping the replications as individuals. However, although, it may lead to more detailed results, new challenges may arise, such as over clustering, since the orders of the replications will be mixed with that way. Moreover, several methodologies can be proposed to filter numerous constant-shape genes before starting the clustering. This may result in reaching the clusters even in shorter time possibly with higher accuracy. However, this will increase the number of steps; hence it may be less practical. Next, since this clustering algorithm can be hold for any condition, such as cancer and control, this methodology can also be extended to be used in biclustering studies. Such studies cluster the genes in two directions, within and between conditions, and may produce more important information for the bioinformaticians. In the first direction, within clusters, the profiles are detected within each condition. In the second step, the profiles from different conditions can be clustered and the genes which show dissimilar profiles in different conditions can be detected with this way. Such genes can be considered to be associated with the disease in interest. Finally, several approaches can be added to our algorithm. First, the algorithm can be modified to handle the possible variations among the successive time points in different time-series. For this modification, the union of the distinct time points over all time-series should be identified. These joint time points can be adapted to every time-series. Expression levels for the non-observed time points can be imputed with interpolation depending on the linear changes assumption between the time points. Next, Short Time Series distance can be substituted with a distance metric based on autocorrelation to allow the shape metric consider all the previous time points instead of just the prior time point. Moreover, a third metric can be added to include a third characteristic besides the magnitude and shape dissimilarities into the general distance matrix. Finally, a posterior step can be added to the algorithm to build the network of the genes in the organisms. For this step, it can be assumed that the genes can activate the other genes which start to show reactions in later time intervals. Cross-correlation measures can be used for this analysis. All these modifications can be applied on **Algorithm CGR** and it can be evaluated on simulation studies if they make the algorithm a more powerful tool.

REFERENCES

Baldi, P., Hatfield, G. W. (2002). DNA Microarrays and Gene Expression From Experiments to Data Analysis and Modeling. Cambridge University Press.

Bar-Joseph, Z. (2004). Analyzing Time Series Gene Expression Data. *Bioinformatics*, Volume: 20, Issue: 16, pp. 2493-2503.

Bar-Joseph, Z., Gerber, G., Jaakkola, T. S., Gifford, D. K., and Simon, I. (2003). Continuous Representations of Time Series Gene Expression Data. *J. Computational Biology*, Volume: 34, pp. 341-356.

Bolshakova, N., Azuaje, F. (2003). Cluster Validation Techniques for Genome Expression Data. *Signal Processing*, Volume: 83, Issue: 4, pp. 825 – 833.

Calza, S., Raffelsberger, W., Ploner, A., Sahel, J., Leveillard, T., and Pawitan, Y. (2007). Filtering Genes to Improve Sensitivity in Oligonucleotide Microarray Data Analysis. *Nucleid Acid Research*, Volume: 35, Issue: 16, Article Number: e102

Celeux, G., Martin, O., and Lavargne, C. (2005). Mixture of Linear Mixed Models for Clustering Gene Expression Profiles from Repeated Microarray Experiments. *Statistical Modelling*, Volume: 5, pp. 243-267.

Cho, R. J, Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. (1998). A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle. *Molecular Cell*, Volume: 2, pp. 65-73.

Chu, S., Derisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P. O., and Herskowitz, I. (1998). The Transcription Program of Sporulation in Budding Yeast. *Science*, Volume: 282, pp. 699-705.

Do, J. H., and Choi, D.-K. (2007). Clustering Approaches to Identifying Gene Expression Patterns from DNA Microarray Data. *Molecules and Cells*, Volume: 25, pp. 279-288.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998) Cluster Analysis and Display of Genome-Wide Expression Patterns. *Proceedings of the National Academy of Sciences of the United States of America*, Volume: 95, pp. 14863-14868

Ernst, J., Nau, G. J., and Bar-Joseph, Z. (2005). Clustering Short Time Series Gene Expression Data. *Bioinformatics*, Volume: 21, pp. i159-i168.

Hakamada, K., Okamoto, M., and Hanai, T. (2006). Novel Technique for Preprocessing High Dimensional Time-Course Data from DNA Microarray: Mathematical Model-Based Clustering. *Bioinformatics,*. Volume: 22, pp. 843-848.

Heard, N. A., Holmes, C. C., Stephens, D. A., Hand, D. J., and Dimopoulos, G. (2005). Bayesian Co-clustering of Anopheles Gene Expression Time Series: Study of Immune Defense Response to Multiple Experimental Challenges. *Proceedings of the National Academy of Sciences of the United States of America*, Volume: 102, pp. 16939-16944.

Irigoien I., Vives, S., and Arenas, C. (2011). Microarray Time Course Experiments: Finding Profiles. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Volume: 8, pp.464 – 475.

Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, Volume: 43, pp. 59-69.

Khan, J., Simon, R., Bittner, M., Chen, Y., Leighton, S. B., Pohida, T., Smith, P. D., Jiang, Y., Gooden, G. C., Trent, J. M., and Meltzer, P. S. (1998). Gene Expression Profiling of Alveolar Rhabdomyosarcoma with cDNA Microarrays. *Cancer Research*, Volume: 58, Issue: 22, pp. 5009 – 5013.

Luan, Y., Li, H. (2004). Model-Based Methods for Identifying Periodically Regulated Genes Based on the Time Course Microarray Gene Expression Data. *Bioinformatics*, Volume: 20, pp. 332-339

Maki, Y., Takanashi, Y., Arikawa, Y., Watanabe, S., Aoshima, K., Eguchi, Y., Ueda, T., Aburatani, S., Kuhara, S., and, Okamoto, M. (2004). An Integrated Comprehensive Workbench for Inferring Genetic Networks: VOYAGENE. *Journal of Bioinformatics and Computational Biology,* Volume: 2, pp. 533-550

Möller-Levet, C. S., Klawonn, F., Cho, K.-H., Yin, H., Wolkenhauer, O. (2005). Clustering of Unevenly Sampled Gene Expression Time-Series Data. *Fuzzy Sets and Systems*, Volume: 152, pp. 49-66.

Nguyen, T. T., Almon, R. R., DuBois, D. C., Jusko, W. J., Androulakis, I. P. (2010). Importance of Replication in Analyzing Time-Series Gene Expression Data: Corticosteroid Dynamics and Circadian Patterns in Rat Liver. *BMC Bioinformatics*, Volume: 11, Article Number: 279

Peddada, S., Harris, S., Zajd, J., and Harvey, E. (2005). ORIOGEN: Order Restricted Inference for Ordered Gene Expression Data. *Bioinformatics*, Volume: 21, pp. 3933-3934.

Ramoni, M. F., Sebastiani, P., and Kohane, I. S. (2002). Cluster Analysis of Gene Expression Dynamics. *Proceedings of the National Academy of Sciences of the United States of America*, Volume: 99, pp. 9121-9126

Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, Volume: 66, pp. 846-850.

Spellman, T. P., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive Identification of Cell Cycle-Regulated Genes of the Yeast Saccharomyces Cerevisiae by Microarray Hybridization. *Molecular Biology*, Volume: 9, pp. 3273-3297.

Szekely, G. J., Rizzo, M. L. (2005). Hierarchical Clustering via Joint Between-Within Distances: Extending Ward's Minimum Variance Method. *Journal of Classification*, Volume: 22, pp. 151-183.

Tan, P. N., Steinbach, M., Kumar, V. (2006). Introduction to Data Mining. Pearson Education, Inc.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., and Golub, T. R. (1999). Interpreting Patterns of Gene Expression with Self Organizing Maps and Applications to Hematopoietic Differentiation. *Proceedings of the National Academy of Sciences of the United States of America*, Volume: 96, pp. 2907-2912.

Tomancak, P., Beaton, A., Weiszmann, R., Kwan, E., Shu, S., Lewis, S. E., Richards, S., Ashburner, M., Hartenstein, V., Celniker, S. E., and Rubin, G. M. (2002). Systematic Determination of Patterns of Gene Expression During *Drosopohila* Embryogenesis. *Genome Biology*, Volume: 3, pp. 1-14.

Yang, Y. H., and Speed, T. P. (2003). Design and Analysis of Comparative Microarray Experiments in Statistical Analysis of Gene Expression Microarray Data. Ed. T. Speed. Chapman & Hall/CRC Press.

Yeung, K. Y, Fraley, C, Murua, A., Raftery, A. E., and Ruzzo, W. L. (2001). Model-Based Clustering and Data Transformations for Gene Expression Data. *Bioinformatics*, Volume: 17, pp. 977-987.

APPENDIX A

RESULTS OF MISCLUSTERING RATES OF THE SIMULATION STUDIES IN SUBSECTION 5.1.2.

Table A.1. Average misclustering rates over 1000 iterations under different weight selections for the simulation study with 50 genes per group with equal time points

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.01	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.00	0.04	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.2. Average misclustering rates over 1000 iterations under different weight selections for the simulation study with 20 genes per group with unequal time points

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	Mis. Rate
D*0 + S*1	0.00	1.00	1.00	1.00	0.00	0.34	0.00	0.01	0.46	0.00	0.00	0.00	0.00	1.00	1.00	0.02
D*0.25 + S*0.75	0.00	0.06	0.00	0.09	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.06	0.00	0.09	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.06	0.00	0.09	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.00	0.06	0.00	0.09	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.3. Average misclustering rates over 1000 iterations under different weight selections for the simulation study with 50 genes per group with unequal time points

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	Mis. Rate
D*0 + S*1	0.00	0.16	1.00	0.93	0.00	0.40	0.00	0.00	0.45	0.00	0.00	0.00	0.00	1.00	1.00	0.01
D*0.25 + S*0.75	0.00	0.03	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.04	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.04	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.00	0.04	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.4. Average misclustering rates over 1000 iterations under different weight selections for the simulation study where the number of genes for each groups was generated from Disc. Unif (5, 35) with equal time points

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	Mis. Rate
D*0 + S*1	0.00	0.00	0.49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.52	0.07
D*0.25 + S*0.75	0.00	0.02	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.07	0.05	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
D*0.75 + S*0.25	0.01	0.13	0.07	0.13	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02
D*1 + S*0	0.01	0.19	0.08	0.17	0.00	0.01	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.03

Table A.5. Average misclustering rates over 1000 iterations under different weight selections for the simulation study where the number of genes for each groups was generated from Disc. Unif (35, 65) with equal time points

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	Mis. Rate
D*0 + S*1	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.07
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.02	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.00	0.05	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01

Table A.6. Average misclustering rates over 1000 iterations under different weight selections for the simulation study where the number of genes for each groups was generated from Disc. Unif (5, 35) with unequal time points

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	Mis. Rate
D*0 + S*1	0.64	0.67	0.75	0.53	0.00	0.41	0.00	0.48	0.47	0.00	0.00	0.00	0.00	0.51	0.74	0.35
D*0.25 + S*0.75	0.01	0.19	0.11	0.18	0.00	0.01	0.00	0.03	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.04
D*0.5 + S*0.5	0.01	0.19	0.09	0.17	0.00	0.01	0.00	0.02	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.03
D*0.75 + S*0.25	0.01	0.19	0.08	0.17	0.00	0.01	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.03
D*1 + S*0	0.01	0.19	0.08	0.17	0.00	0.01	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.03

Table A.7. Average misclustering rates over 1000 iterations under different weight selections for the simulation study where the number of genes for each groups was generated from Disc. Unif (35, 65) with unequal time points

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	Mis. Rate
D*0 + S*1	0.63	0.60	0.75	0.51	0.00	0.41	0.00	0.50	0.47	0.00	0.00	0.00	0.00	0.50	0.70	0.34
D*0.25 + S*0.75	0.00	0.04	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
D*0.5 + S*0.5	0.00	0.05	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
D*0.75 + S*0.25	0.00	0.05	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
D*1 + S*0	0.00	0.05	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01

APPENDIX B

RESULTS OF ACCURACY MEASURES FOR THE SIMULATION STUDIES IN SUBSECTION 5.1.3.

		Real	
		Constant	Non-Constant
Predicted	Constant	a	b
	Non-Constant	c	d

		Real	
		With Variation Among Replications	Without Variation Among Replications
Predicted	With Variation Among Replications	a	b
	Without Variation Among Replications	c	d

$$Correct\ Classification\ Rate = \frac{a+d}{a+b+c+d}$$

$$Sensitivity = \frac{a}{a+c}$$

$$Specificity = \frac{d}{b+d}$$

$$False\ Positive\ Rate = \frac{b}{b+d}$$

$$False\ Negative\ Rate = \frac{c}{a+c}$$

$$Positive\ Predictive\ Power = \frac{a}{a+b}$$

$$Negative\ Predictive\ Power = \frac{d}{c+d}$$

Table B.1. Results of the accuracy measures for detecting the constant genes

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S1.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.5	0.98	0.99	0.90	0.10	0.01	0.98	0.96
S2.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S2.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S2.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S2.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S2.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S3.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S3.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S3.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S3.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S3.5	0.98	0.99	0.89	0.11	0.01	0.98	0.96
S4.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S4.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S4.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S4.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S4.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S5.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S5.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S5.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S5.4	1.00	1.00	0.99	0.01	0.00	1.00	1.00
S5.5	0.98	0.99	0.94	0.06	0.01	0.99	0.92
S6.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S6.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S6.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S6.4	1.00	0.99	1.00	0.00	0.01	1.00	0.99
S6.5	0.98	0.96	1.00	0.00	0.04	1.00	0.96
S7.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S7.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S7.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S7.4	1.00	1.00	0.99	0.01	0.00	1.00	1.00
S7.5	0.98	0.99	0.94	0.06	0.01	0.99	0.93

Table B.1 (cont'd). Results of the accuracy measures for detecting the constant genes

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S8.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S8.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S8.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S8.4	1.00	0.99	1.00	0.00	0.01	1.00	0.99
S8.5	0.98	0.96	1.00	0.00	0.04	1.00	0.97
S9.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S9.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S9.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S9.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S9.5	0.98	0.99	0.90	0.10	0.01	0.98	0.96
S10.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S10.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S10.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S10.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S10.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S11.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S11.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S11.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S11.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S11.5	0.98	0.99	0.89	0.11	0.01	0.98	0.96
S12.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S12.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S12.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S12.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S12.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S13.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S13.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S13.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S13.4	1.00	1.00	0.99	0.01	0.00	1.00	0.99
S13.5	0.98	0.99	0.94	0.06	0.01	0.99	0.92
S14.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S14.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S14.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S14.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S14.5	0.98	0.96	1.00	0.00	0.04	1.00	0.96

Table B.1 (cont'd). Results of the accuracy measures for detecting the constant genes

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S15.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S15.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S15.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S15.4	1.00	1.00	0.99	0.01	0.00	1.00	1.00
S15.5	0.98	0.99	0.94	0.06	0.01	0.99	0.93
S16.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S16.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S16.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S16.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S16.5	0.98	0.96	1.00	0.00	0.04	1.00	0.97
S17.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S17.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S17.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S17.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S17.5	1.00	1.00	0.99	0.01	0.00	1.00	1.00
S18.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S18.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S18.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S18.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S18.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S19.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S19.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S19.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S19.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S19.5	1.00	1.00	0.99	0.01	0.00	1.00	1.00
S20.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S20.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S20.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S20.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S20.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S21.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S21.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S21.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S21.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S21.5	1.00	1.00	0.98	0.02	0.00	1.00	0.99

Table B.1. (cont'd). Results of the accuracy measures for detecting the constant genes

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S22.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S22.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S22.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S22.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S22.5	0.99	0.98	1.00	0.00	0.02	1.00	0.98
S23.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S23.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S23.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S23.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S23.5	1.00	1.00	0.98	0.02	0.00	1.00	0.99
S24.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S24.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S24.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S24.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S24.5	0.99	0.98	1.00	0.00	0.02	1.00	0.99
S25.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S25.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S25.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S25.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S25.5	1.00	1.00	0.99	0.01	0.00	1.00	1.00
S26.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S26.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S26.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S26.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S26.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S27.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S27.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S27.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S27.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S27.5	1.00	1.00	0.99	0.01	0.00	1.00	1.00
S28.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S28.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S28.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S28.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S28.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00

Table B.1. (Cont'd). Results of the accuracy measures for detecting the constant genes

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S29.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S29.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S29.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S29.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S29.5	1.00	1.00	0.98	0.02	0.00	1.00	0.99
S30.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S30.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S30.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S30.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S30.5	0.99	0.98	1.00	0.00	0.02	1.00	0.98
S31.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S31.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S31.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S31.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S31.5	1.00	1.00	0.98	0.02	0.00	1.00	0.99
S32.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S32.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S32.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S32.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S32.5	0.99	0.98	1.00	0.00	0.02	1.00	0.99

Table B.2. Results of the accuracy measures for detecting the genes with variations among replications

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S1.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S1.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S2.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S2.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S2.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S2.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S2.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S3.1	1.00	0.99	1.00	0.00	0.01	1.00	1.00
S3.2	1.00	0.96	1.00	0.00	0.04	1.00	1.00
S3.3	1.00	0.99	1.00	0.00	0.01	0.99	1.00
S3.4	1.00	0.97	1.00	0.00	0.03	0.98	1.00
S3.5	0.98	**0.67**	1.00	0.00	**0.33**	0.99	0.98
S4.1	0.99	**0.52**	1.00	0.00	**0.48**	0.99	0.99
S4.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S4.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S4.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S4.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S5.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S5.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S5.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S5.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S5.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S6.1	0.93	**0.49**	0.97	0.03	**0.51**	**0.55**	0.96
S6.2	0.91	0.06	0.99	0.01	0.94	**0.49**	0.92
S6.3	0.95	0.04	1.00	0.00	0.96	**0.49**	0.95
S6.4	0.88	**0.49**	0.94	0.06	**0.51**	**0.55**	0.93
S6.5	0.93	**0.49**	0.97	0.03	**0.51**	**0.55**	0.96
S7.1	0.95	0.00	1.00	0.00	1.00	**0.45**	0.95
S7.2	0.97	0.00	1.00	0.00	1.00	**0.44**	0.97
S7.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S7.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S7.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00

Table B.2 (cont'd). Results of the accuracy measures for detecting the genes with variations among replications

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S8.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S8.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S8.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S8.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S8.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S9.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S9.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S9.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S9.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S9.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S10.1	1.00	0.97	1.00	0.00	0.03	0.98	1.00
S10.2	0.99	0.81	1.00	0.00	0.19	1.00	0.99
S10.3	0.99	**0.60**	1.00	0.00	**0.40**	1.00	0.99
S10.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S10.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S11.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S11.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S11.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S11.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S11.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S12.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S12.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S12.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S12.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S12.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S13.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S13.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S13.3	0.99	**0.82**	1.00	0.00	**0.18**	1.00	0.99
S13.4	0.98	**0.41**	1.00	0.00	**0.59**	1.00	0.98
S13.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S14.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S14.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S14.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S14.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S14.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00

Table B.2 (cont'd). Results of the accuracy measures for detecting the genes with variations among replications

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S15.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S15.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S15.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S15.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S15.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S16.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S16.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S16.3	0.99	0.94	1.00	0.00	0.06	0.97	1.00
S16.4	1.00	0.90	1.00	0.00	0.10	1.00	1.00
S16.5	0.99	**0.50**	1.00	0.00	**0.50**	1.00	0.99
S17.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S17.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S17.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S17.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S17.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S18.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S18.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S18.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S18.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S18.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S19.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S19.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S19.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S19.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S19.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S20.1	0.98	**0.37**	1.00	0.00	**0.63**	1.00	0.98
S20.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S20.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S20.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S20.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S21.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S21.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S21.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S21.4	1.00	0.88	1.00	0.00	0.12	1.00	1.00
S21.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00

Table B.2 (cont'd). Results of the accuracy measures for detecting the genes with variations among replications

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S22.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S22.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S22.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S22.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S22.5	0.99	0.95	1.00	0.00	0.05	0.98	1.00
S23.1	1.00	0.92	1.00	0.00	0.08	1.00	1.00
S23.2	0.99	**0.50**	1.00	0.00	**0.50**	1.00	0.99
S23.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S23.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S23.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S24.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S24.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S24.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S24.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S24.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S25.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S25.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S25.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S25.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S25.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S26.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S26.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S26.3	0.98	**0.40**	1.00	0.00	**0.60**	1.00	0.98
S26.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S26.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S27.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S27.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S27.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S27.4	1.00	0.99	1.00	0.00	0.01	1.00	1.00
S27.5	1.00	0.95	1.00	0.00	0.05	1.00	1.00
S28.1	0.99	**0.63**	1.00	0.00	**0.37**	1.00	0.99
S28.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S28.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S28.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S28.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00

Table B.2 (cont'd). Results of the accuracy measures for detecting the genes with variations among replications

	Correct Classification Rate	Sensitivity	Specificity	False Negative Rate	False Positive Rate	Positive Predictive Power	Negative Predictive Power
S29.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S29.2	1.00	0.99	1.00	0.00	0.01	1.00	1.00
S29.3	1.00	0.95	1.00	0.00	0.05	1.00	1.00
S29.4	0.99	**0.63**	1.00	0.00	**0.37**	1.00	0.99
S29.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S30.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S30.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S30.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S30.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S30.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S31.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S31.2	0.99	**0.55**	1.00	0.00	**0.45**	1.00	0.99
S31.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S31.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S31.5	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S32.1	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S32.2	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S32.3	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S32.4	1.00	1.00	1.00	0.00	0.00	1.00	1.00
S32.5	0.99	**0.55**	1.00	0.00	**0.45**	1.00	0.99

APPENDIX C

RESULTS OF MISCLUSTERING RATES FOR THE SIMULATION STUDIES IN SUBSECTION 5. 1. 3.

Table C. Misclustering rates for 32 simulation studies where each study is denoted by using the notations stated in Table 5.4

S1	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.17
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.03	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.31	0.00	0.03

S2	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.17
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.03	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S3	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.17
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.03	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.32	0.00	0.03

S4	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.17
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.03	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S5	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.49	0.51	0.00	0.49	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.70	0.71	0.15
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00
D*1 + S*0	0.03	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.36	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.27	0.00	0.03

S6	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.49	0.51	0.00	0.49	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63	0.68	0.69	0.34
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.07	0.00	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
D*1 + S*0	0.19	0.00	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.01	0.00	0.79	0.27	0.30	0.26	0.31	0.01	0.01	0.02	0.00	0.00	0.00	0.05

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S7	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.49	0.51	0.00	0.49	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63	0.69	0.68	0.16
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00
D*1 + S*0	0.03	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.31	0.00	0.00	0.00	0.01	0.10	0.11	0.09	0.00	0.24	0.00	0.03

S8	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.49	0.51	0.00	0.49	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.66	0.68	0.66	0.36
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.01	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.20	0.20	0.23	0.00	0.00	0.00	0.01
D*1 + S*0	0.05	0.00	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.71	0.21	0.20	0.18	0.23	0.49	0.52	0.52	0.00	0.00	0.00	0.05

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S9	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.01	0.00	0.03	0.00	0.06	0.92	0.00	0.06	0.92	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.18
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.03	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.31	0.00	0.03

S10	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.01	0.00	0.02	0.00	0.01	0.99	0.00	0.01	0.99	0.00	0.00	0.00	0.00	0.00	0.23	0.00	0.00	0.00	0.00	0.00	0.60	0.66	0.66	0.18
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.03	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S11	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.01	0.00	0.03	0.00	0.07	0.91	0.00	0.06	0.92	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.01	0.01	0.01	0.00	1.00	1.00	0.18
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.03	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.32	0.00	0.03

S12	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.01	0.00	0.02	0.00	0.01	0.99	0.00	0.01	0.99	0.00	0.00	0.00	0.00	0.00	0.18	0.00	0.00	0.04	0.04	0.05	0.59	0.66	0.67	0.18
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.03	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S13	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.04	0.00	0.13	0.00	0.48	0.50	0.00	0.47	0.52	0.00	0.00	0.00	0.00	0.00	0.15	0.00	0.00	0.02	0.02	0.01	0.60	0.70	0.71	0.16
D*0.25 + S*0.75	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00
D*1 + S*0	0.03	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.36	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.27	0.00	0.03

S14	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.08	0.00	0.22	0.00	0.49	0.51	0.00	0.48	0.51	0.00	0.00	0.00	0.00	0.00	0.24	0.00	0.00	0.05	0.05	0.05	0.63	0.67	0.67	0.35
D*0.25 + S*0.75	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.05	0.05	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.12	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.07	0.00	0.00	0.00	0.00	0.10	0.11	0.11	0.00	0.00	0.00	0.01
D*0.75 + S*0.25	0.12	0.00	0.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.39	0.01	0.01	0.01	0.01	0.08	0.09	0.08	0.00	0.00	0.00	0.02
D*1 + S*0	0.19	0.00	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.01	0.00	0.79	0.27	0.30	0.26	0.31	0.01	0.01	0.02	0.00	0.00	0.00	0.05

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S15	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.01	0.00	0.08	0.00	0.47	0.51	0.00	0.46	0.51	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.45	0.46	0.48	0.63	0.69	0.68	0.18
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.28	0.31	0.31	0.00	0.00	0.00	0.01
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.19	0.18	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.13	0.15	0.15	0.00	0.04	0.00	0.01
D*1 + S*0	0.03	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.31	0.00	0.00	0.00	0.01	0.10	0.11	0.09	0.00	0.24	0.00	0.03

S16	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.15	0.00	0.48	0.51	0.00	0.47	0.52	0.00	0.00	0.00	0.00	0.00	0.16	0.00	0.00	0.60	0.59	0.63	0.64	0.67	0.65	0.37
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.53	0.55	0.55	0.00	0.00	0.00	0.01
D*0.5 + S*0.5	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.60	0.63	0.64	0.00	0.00	0.00	0.01
D*0.75 + S*0.25	0.00	0.00	0.18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.61	0.63	0.65	0.00	0.00	0.00	0.02
D*1 + S*0	0.05	0.00	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.71	0.21	0.20	0.18	0.23	0.49	0.52	0.52	0.00	0.00	0.00	0.05

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S17	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.17
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00

S18	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.17
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table C (Cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S19	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.17
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00

S20	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.17
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S21	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.49	0.51	0.00	0.49	0.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.70	0.69	0.16
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.01

S22	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.49	0.51	0.00	0.49	0.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.65	0.68	0.67	0.37
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.04	0.00	0.41	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.66	0.27	0.28	0.24	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.05

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S23	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.49	0.51	0.00	0.49	0.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.71	0.68	0.18
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.01

S24	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.00	0.00	0.49	0.51	0.00	0.49	0.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.66	0.68	0.66	0.39
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63	0.08	0.11	0.09	0.11	0.50	0.49	0.51	0.00	0.00	0.00	0.04

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S25	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.00	0.00	0.01	0.00	0.03	0.96	0.00	0.04	0.95	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.17
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00

S26	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.69	0.80	0.66	0.19	0.80	0.78	0.00	0.54	0.73	0.00	0.69	0.02	0.00	0.73	0.65	0.00	0.38	0.79	0.80	0.79	0.61	0.67	0.65	0.52
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S27	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.56	0.63	0.63	0.19	0.78	0.77	0.00	0.54	0.64	0.00	0.67	0.02	0.00	0.55	0.64	0.00	0.32	0.97	0.97	0.98	0.00	1.00	1.00	0.52
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00

S28	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.57	0.63	0.65	0.19	0.80	0.79	0.00	0.54	0.71	0.00	0.68	0.02	0.00	0.56	0.65	0.00	0.37	0.98	0.98	0.98	0.62	0.66	0.65	0.52
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S29	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.69	0.79	0.63	0.19	0.78	0.77	0.00	0.58	0.63	0.00	0.69	0.02	0.00	0.73	0.67	0.00	0.36	0.79	0.79	0.79	0.62	0.70	0.69	0.51
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.01

S30	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.70	0.80	0.64	0.20	0.79	0.78	0.00	0.59	0.66	0.00	0.72	0.02	0.00	0.74	0.69	0.00	0.42	0.80	0.80	0.80	0.64	0.66	0.66	0.56
D*0.25 + S*0.75	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.5 + S*0.5	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
D*1 + S*0	0.04	0.00	0.41	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.66	0.27	0.28	0.24	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.05

Table C (cont'd). Misclustering results for 32 simulation studies where each study was denoted by using the notations stated in Table 5.4

S31	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.57	0.62	0.62	0.19	0.78	0.77	0.00	0.56	0.61	0.00	0.68	0.02	0.00	0.55	0.66	0.00	0.34	1.00	1.00	1.00	0.62	0.71	0.68	0.46
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.22	0.21	0.00	0.00	0.00	0.01
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*0.75 + S*0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D*1 + S*0	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.01

S32	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21	G22	G23	Mis. Rate
D*0 + S*1	0.57	0.64	0.63	0.20	0.79	0.78	0.00	0.59	0.65	0.00	0.71	0.02	0.00	0.57	0.69	0.00	0.39	1.00	1.00	1.00	0.62	0.67	0.67	0.54
D*0.25 + S*0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63	0.63	0.65	0.00	0.00	0.00	0.02
D*0.5 + S*0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.68	0.65	0.68	0.00	0.00	0.00	0.02
D*0.75 + S*0.25	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.66	0.63	0.65	0.00	0.00	0.00	0.02
D*1 + S*0	0.01	0.00	0.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63	0.08	0.11	0.09	0.11	0.50	0.49	0.51	0.00	0.00	0.00	0.04

APPENDIX D

R CODES FOR ALGORTIHM CGR AND CLUSTER VALIDATION TECHNIQUES FOR REAL DATA

```
cgr <- function(d, tp, repl, w = 0.5, mincl = 2, maxcl = 50, ...) {
	ptm <- proc.time()

	### Converting the data set into a 3D array where the dimensions are
	### n for the number of genes, t for the number of time points, r for the number of replications

	data <- array(0, dim=c(dim(d)[1], (length(tp)/max(repl)), max(repl)))
	data_temp <- mat.or.vec(dim(d)[1], (length(tp)/max(repl)))
	for(i in 1:max(repl)) {
		data_temp <- d[, which(repl==i)]
		time <- tp[which(repl==i)]
		for(j in seq_along(time)) {
			data[, j, i] <- data_temp[, which(time==min(time))]
			data_temp <- as.matrix(data_temp[, -which(time==min(time))])
			time <- time[-which(time==min(time))]
		}
	}

	#print(data)

	### Sorting the time points into a vector named "time"

	timepts <- mat.or.vec(1, (length(tp)/max(repl)))
	timepoints <- tp
	for(i in seq_along(timepts)) {
		timepts[i] <- min(timepoints)
		timepoints <- timepoints[-which(timepoints==min(timepoints))]
	}

	#print(time)

	n <- dim(data)[1]
	t <- dim(data)[2]
	r <- dim(data)[3]

	### Calculating the squared Euclidean distances

	Dist <- mat.or.vec(n, n)
```

```
	Slopesim <- mat.or.vec(n, n)

	for(rep in 1:r) {
		Dist <- Dist + (as.matrix(dist(data[, , rep])))^2
	}

	### Calculating the slopes

	slopes <- array(0, dim=c(n, (t-1), r))
	for(k in 1:r) {
		for(i in 1:n) {
			for(j in 1:(t-1)) {
			slopes[i, j, k] <- data[i, (j+1), k] - data[i, j, k]/(timepts[j+1] -
timepts[j])
			}
		}
	}

	### Calculating the squared STS distances

	for(rep in 1:r) {
		Slopesim <- Slopesim + (as.matrix(dist(slopes[, , rep])))^2
	}


	### Standardizing the distance matrices

	rd <- range(Dist)
	if(rd[2]==0 & rd[1]==0) {Dist=Dist
		} else if((rd[2] - rd[1])==0) {Dist = Dist / (rd[1])
		} else {
		Dist <- Dist / (rd[2] - rd[1])
	}

	rc <- range(Slopesim)
	if(rc[2]==0 & rc[1]==0) {Slopesim = Slopesim
		} else if((rc[2]-rc[1])==0) {Slopesim = Slopesim / (rc[1])
		} else {
		Slopesim <- Slopesim / (rc[2] - rc[1])
	}


	### Combining the distance matrices

	sim <- (w * Dist) + ((1-w) * Slopesim)
```

```
rownames(sim) <- rownames(d)
colnames(sim) <- rownames(d)

### Clustering the genes with a hierarchical clustering

cluster <- hclust(as.dist(sim), method = "ward")

between1 <- mat.or.vec(1,(maxcl-mincl+1))
within1 <- mat.or.vec(1,(maxcl-mincl+1))
between2 <- mat.or.vec(1,(maxcl-mincl+1))

for(cn in mincl:maxcl){ #for1
        cl <- cutree(cluster, k=cn)
        with1 <- mat.or.vec(1,cn)

        btw1 <- mat.or.vec(1,cn)
        btw2 <- mat.or.vec(1,cn)

        for(i in 1:cn){ #for2
                temp <- sim[which(cl==i), which(cl==i)]
                with1[i] <- sum(temp)/2

                b1 <- mat.or.vec(1,(cn-1))
                b2 <- mat.or.vec(1,(cn-1))
                v <- 1
                        for(j in 1:cn){#for3
                        if(i==j){v=v
                        }else{
                        temp <- sim[which(cl==i),which(cl==j)]
                        b1[v]<-min(temp)
                        b2[v]<-mean(temp)
                        v<-v+1}
                }#for3
                btw1[i]<-min(b1)
                btw2[i]<-min(b2)
        }#for2

within1[(cn-mincl+1)]<-max(with1)

between1[(cn-mincl+1)]<-min(btw1)
between2[(cn-mincl+1)]<-min(btw2)

}#for1

ww<- within1
```

```
        bw<-rbind(between1,between2)

        time <- proc.time() - ptm
        result <- list(HClust = cluster, DistMat = sim, EuclideanDist = Dist, SlopeDist = Slopesim, expvals = data, within = ww, between = bw, mincl = mincl, maxcl = maxcl, pr.time = time, timepts = timepts)
}


### Drawing Cluster Validation graphs


valid.graph <- function(val = valid, min = 7, max = 25) {


ww <- val$within
bw <- val$between
par(mar = c(4, 7, 3, 2))
par(mfrow = c(1, 2))
plot(min:max, (ww[(min - val$min + 1):(max - val$min + 1)]/bw[1, (min - val$min + 1):(max - val$min + 1)]), xlab = "Cluster Number",
ylab = "VD1", type = "b", axes = FALSE, main = "")
box()
axis(1, seq(min, max, 2))
axis(2)
plot(min:max, (ww[(min - val$min + 1):(max - val$min + 1)]/bw[2, (min - val$min + 1):(max - val$min + 1)]), xlab = "Cluster Number",
ylab = "VD2", type = "b", axes = FALSE, main = "")
box()
axis(1, seq(min, max, 2))
axis(2)
}
```