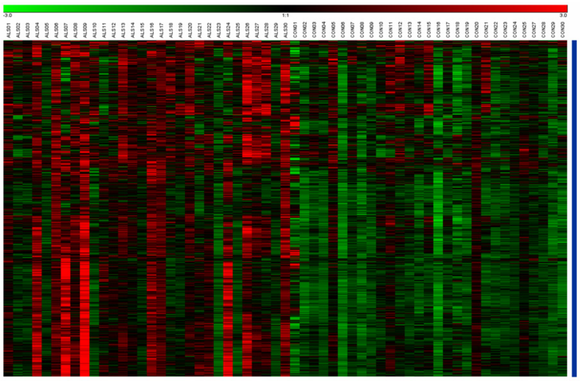
INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa



# Unravelling regulatory modules involved in Amyotrophic Lateral Sclerosis

**Mafalda de Oliveira Ruas Gonçalves**

Thesis to obtain the Master of Science Degree in

**Biomedical Engineering**

**Examination Committee:**

| | |
|---|---|
| Chaiperson: | Prof. Patrícia Margarida Piedade Figueiredo |
| Supervisor: | Prof. Sara Alexandra Cordeiro Madeira |
| Co-supervisor: | Prof. Alexandre Paulo Lourenço Francisco |
| Members of the Committee: | |
| | Doctor Pedro Monteiro |
| | Prof. Nuno Gonçalo Pereira Mira |

**November 2012**

***Let us keep looking, in spite of everything. Let us keep searching. It is indeed the best method of finding, and perhaps thanks to our efforts, the verdict we will give such a patient tomorrow will not be the same we must give this man today.***

Charcot (1889)

# Acknowledgments

Em primeiro lugar, gostaria de agradecer aos professores Sara Madeira e Alexandre Francisco por me terem prontamente dado a oportunidade de realizar esta dissertação de mestrado e, por me terem acompanhado ao longo deste processo.

Importantes também pelo seu apoio e simpatia foram os meus colegas do projecto NEUROCLINOMICS, que potenciaram interessantes discussões filosóficas sobre o tema e muito além deste.

Gostaria de deixar expresso o meu reconhecimento e amizade por todos os amigos que fiz no curso, mas também pelos mais antigos e inestimáveis que me completam a vida.

Agradeço por fim a meus pais. Sem eles não estaria aqui. Obrigada por acreditarem em mim, sempre.

Este trabalho foi suportado pela FCT através do projeto NEUROCLINOMICS (PTDC/EIA-EIA/111239/2009).

# Abstract

Amyothrophic Lateral Sceloris (ALS) is a devastating disease, whose pathogenesis is still not fully understood. In the literature, evidence regarding the genetic framework of disease abounds. In the present work, two different approaches are combined in an attempt to unravel novel disturbed biological pathways in sporadic cases of ALS. First, standard unsupervised data mining techniques are employed to samples and genes: hierarchical clustering and K-Means clustering. Then, the Weighted Gene Co-expression Analysis based on network concepts is used to obtain modules of highly correlated genes. Several network concepts are integrated to guide the algorithm and assist the selection the more cohesive modules. The purpose of using both procedures is to identify genes involved in abnormal biological processes. Toward this end, an overlap study of the resulting clusters and modules was performed. This procedure was followed by functional enrichment with Gene Ontology and KEGG terms. With both approaches significant groups of genes were identified, which should be analysed in depth as future work. The application of WGCNA provided a more straightforward identification of enriched modules. However, clustering techniques also led to results with high correlation with the disease. As a future work, to compare these different approaches and to improve the confidence of the results, a larger dataset should be considered.

# Keywords

Amyotrophic Lateral Sclerosis, K-Means, Hierarchical Clustering, Weighted Gene Co-expression Network Analysis, Regulatory Gene Networks, Transcriptomics

# Resumo

A esclerose lateral amiotrófica é uma doença de efeitos devastadores cuja patogénese não é totalmente conhecida, ainda que existam evidências de contribuição genética. Neste trabalho foram utilizadas duas diferentes abordagens para o tratamento de dados transcriptómicos adquiridos com a tecnologia de microarray para casos esporádicos desta doença. O objectivo principal prendia-se à potencial descoberta de vias biológicas e genes perturbados nesta condição relativamente ao padrão saudável. Primeiro, técnicas não supervisionadas, o *clustering* hierárquico e o algoritmo de *k-means*, foram utilizadas para agrupar tanto as diferentes amostras como os genes. Seguidamente, um método para a construção de redes de regulação genética foi aplicado a este conjunto de dados para determinar módulos de genes correlacionados, o *Weighted Gene Co-expression Network Analysis* (WGCNA). No processo de determinação destes módulos foram integrados conceitos da literatura de redes complexas. Para melhor comparar os diferentes métodos, realizou-se uma sobreposição de todos os resultados seguida de enriquecimento funcional com a *Gene Ontology* e a KEGG. Em todos os métodos foram identificados grupos de genes interessantes com a presença de funções potencialmente relevantes no estudo desta doença que devem ser analisados com mais detalhe em trabalhos futuros. Finalmente, é sugerido comparar estes procedimentos utilizando um conjunto de dados com maior número de amostras.

# Palavras Chave

Esclerose Lateral Amiotrófica, K-Means, Clustering hierárquico, Weighted Gene Co-expression Network Analysis, Redes de Regulação Genética, Transcriptoma

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

## Contents

## 1.1 Context and Motivation

Microarray's technology has the ability of parallel assessment of thousands of gene expression profiles and, is nowadays an established tool in the study of the genes' behaviour under different conditions. In the past decades, the evolution of this technique was accompanied by the advent of database technology and it is now possible to composed datasets of different studies to increase the confidence of results.

These advances were rapidly followed by the emergence of several different platforms and processing techniques. In fact, the variety of processing techniques and the lack of consensus regarding the standardization of the pipeline is one of the main challenges of this technology. The two main approaches to unravel the information of a microarray are either to use techniques from the data mining literature, such as supervised and unsupervised learning methods, or based on the network literature.

Supervised learning techniques are focused on the identification of elements that allow the classification of samples, whilst unsupervised (or clustering) are turned to the identification of patterns of similarity. In fact, using clustering techniques is a common practice in microarray's studies. However, each of such techniques presents different pitfalls and thus there is no consensus regarding the best one. Two of the standard procedures are the hierarchical clustering and the K-Means algorithm, that group genes based on the assumption their similar expression implies they are also being co-expressed.

On the other hand, gene regulatory networks try to identify how genes are related to each other in an attempt to identify gene regulatory sub-networks. A recent method is Weighted Gene Co-expression Network, that has been successfully applied in several different contexts to identify modules containing relevant genes for the problem under study. This method was been previously applied to Amyotrophic Lateral Sclerosis (ALS) with the identification of two modules enriched with differentially expressed genes. These modules presented significant, although very general, gene functions regarding this disease, such as post-translational modification, infection mechanism, neurological disorder, genetic disorder, inflammatory disease and skeletal and muscle disorder.

ALS is a devastating motor neuron disease with a fatal outcome. Although several affected biological pathways have been identified, its pathogenesis is still largely unknown. Genetic factors have been undoubtedly associated with this disorder, either in its familial or sporadic form. However, more genes and functions have been identified with the familial form and currently evidences of their involvement in the sporadic form was verified.

## 1.2 Problem Formulation

As enough evidences of genetic contribution to the pathogenesis of Amyotrophic Lateral Sclerosis have been yet identified, transcriptomic studies focused on the identification of perturbed gene expression profiles may provide novel insights into this perturbed mechanism.

The main goal of the present work, is to explore an Amyotrophic Lateral Sclerosis dataset in an attempt to identify groups of genes that present differences in their expression profiles when comparing with controls. Two main approaches to unravel the information contained in this data are used, clustering techniques and a correlated-based network inference method. Hierarchical clustering and K-Means methods are chosen for the first approach as they are standard procedures in the processing of microarrays. As a correlation-based method it is used the Weighted Gene Co-expression Network Analysis as it has been successfully applied to this disease.

## 1.3 Main Contributions

The main contributions obtained in this work were:

1. Revision of concepts and methods used in Weighted Gene Co-expression Networks applications;
2. Proposal to use standard networks concepts to help guiding the definition of modules in the WGCNA approach;
3. Comparison of the latter network approach with standard clustering techniques with the use of a real dataset;
4. Identification of a set of genes related with significant biological functions associated.

## 1.4 Thesis Outline

This work is divided into four main sections that represent the pipeline followed.

First the state of the art is revised in the section Background (Section 2). It was considered relevant to describe the disease (Section 2.1) and the technology of microarray (Section 2.2). Then, the current state of art of the techniques for processing microarrays (Section 2.3) is given to contextualize its use and it is also presented the related work (Section 2.3.5).

The methods used in this work are then fully detailed in the Methods (Section 3). It is presented the dataset used in this work (Section 3.1), the pre-processing steps applied (Section 3.3) as well as the software used (Section 3.2). Data analysis chosen procedures are described in the next sections (Section 3.4): hierarchical clustering algorithm (Section 3.4.1), k-means method (Section 3.4.1) and the Weighted Gene Co-expression Network Analysis (Section 3.4.2).

The results of applying the former methods to a dataset of Amyotrophic Lateral Sclerosis (Section 3.1) are presented in Section 4. Exploratory results (Section 4.1) are performed on this dataset making use of the previously stated clustering techniques (Section 4.1.2). Then the WGCNA method is applied (Section 4.2). Finally, the results from each method are compared and functional enriched (Section 4.3).

# 2

# Background

## Contents

## 2.1 Amyotrophic Lateral Sclerosis

Amyotrophic Lateral Sclerosis (ALS) is also colloquially known as Lou Gehrig's disease (in American English), Motor Neurone Disease (in British English) [1] and Charcot's disease [2] after the French neurologist that first described it in 1869. It is the most frequent motor neuron disorder with adult onset and is characterized by the progressive degeneration of the upper motor neurons (UMN) of the corticospinal tract and the lower motor neurons (LMN) of the spinal cord anterior horns [3–5]. It leads to progressive weakness and atrophy of muscles, paralysis and ultimately death [6, 7]. Death usually results from respiratory failure and occurs approximately 3 years after symptom onset [2, 8]. Each year, around 5,600 people are diagnosed with ALS with the incidence rate of two per 100,000 a year [6].

**Clinical Manifestations** ALS is mostly characterized by motor system perturbations, such as muscle weakness, cramps and twitching [2]. In approximately half of the patients cognitive perturbations have been identified, at least at a subtle level, as perturbations in attention, verbal fluency, short-term memory, visuospatial abilities, executive functioning, among others [9]. The causes for this cognitive deficits have not yet been identified although some theories have been proposed [9]. In some cases, ALS has been related to frontotemporal dementia (FTD, FTDL) [8, 10].

There are two main clinical manifestations of ALS, which allow to classify the disease into bulbar or limb onset motor neuron disease regarding whether it has started in the bulbar or spinal innervated muscles [10]. In some cases, it may be further characterized into progressive muscular atrophy, primary lateral scleroris, or simply ALS and then classified by the El Escorial classification system as definitive, probable or possible. Other phenotypes have also been described as ALS, such as flail arm syndrome and flail leg syndrome [8].

**Diagnosis** Due to the devastating effects of ALS diagnosis and since several diseases mimic its phenotype, it is essential to achieve a high certainty in the diagnosis of ALS. Nowadays, clinicians use a combination of approaches of clinical assessment and laboratory investigation. The final diagnosis is difficult to accomplish and thus the El Escorial criteria defines levels of certainty associated to the disease's diagnosis but further physiological tests should be performed [10].

**Pathogenesis** The pathogenesis of ALS is still largely unknown [5]. Understanding the underlying processes that lead to motor neuron degeneration and death is essential to better perceive the disease and to develop therapeutic targets [7, 10].

Relatively to the genetic origin of the disease, 5-10% of the ALS cases are of familial origin (fALS), whilst the majority are sporadic (sALS) [7]. Both these types present indistinguishable clinical manifestations [7] and some of the genes that have been identified in the familial form are also present in the sporadic form [2]. Taking these findings into account as well as others, some authors [8, 11] consider there is enough evidence of genetic contribution in all ALS cases although with some patient's genetic heterogeneity [5]. The list of genes associated with the familial form keeps increasing [11], some of which are presented in Table 2.1. The first gene related with ALS was SOD1, which encodes for copper/zinc ion-binding superoxide dismutase [8, 10]. In approximately 30% of the familial cases of ALS present mutations in SOD1, TARDBP, fused in sarcoma (FUS) and the optinerin (OPTN) gene [3]. In most sporadic ALS patients, the nuclear protein TAR DNA binding protein 43 (TDP-43) is accumulated in the cytoplasm [7]. Recently, the most common genetic mutation, in both fALS and sALS, was associated with gene C9ORF72 in a non coding sequence [3].

ALS' pathogenesis is considered multifactorial [10] with the possible involvement of a variety of biochemical pathways [11]. First, it is now clear that motor neuron degeneration requires perturbation of non-neuronal cells, specially astrocytes and microglia [1, 3, 10]. Several biological and molecular processes have been identified as potentially involved in motor neuron death in this disease, such as neuroinflammatory processes [1], protein aggregation or inclusions [1, 8], oxidative stress and abnormal axonal transport or axonopathy [7, 11]. In fact, in the majority of ALS cases, ubiquinated cytoplasmatic inclusions formed by TAR DNA-binding protein (TDP-43) appear, and so this manifestation is considered a pathological characteristic of the disease [3, 8]. An important role in the motor neuron degeneration in this disease has been attributed to glutamate-mediated excitotoxic effects [7]. Other processes have been also studied, such as the potential role of the retinoid signaling pathway [11]. Finally, dysfunction of some cell organelles, such as mitochondria are also present in the disease [7, 10, 11]. The exact role of these perturbed biological processes as well as the identified genes in the development of the disease, remains unclear [1, 3, 8].

Two currently updated and complementary databases contain the information relatively to the relationships between gene variants and ALS phenotypes as well as current analysis of ALS genes. These are ALSoD (genotype–phenotype correlation with an emphasis on familial ALS) [12] and ALSGene (meta-analysis of association data and genome-wide association studies) [13].

**Research and Treatment** Currently, there is no effective therapeutic interventions to significantly slow or even stop the progression of the disease [1, 2, 11]. Regarding pharmaceutical care, the only effective drug is Riluzole, which inhibits presynaptic glutamate release, and increases life expectancy by 2-3 months [7]. Therefore, healthcare of these patients is usually a combination of different expertises directed at the management of the disease's symptoms [10] and improvement of the quality of life. Focus is given to maintenance of adequate nutrition and the use of non-invasive ventilation support for respiratory symptoms [7]. Current research for therapeutic targets and treatments is reviewed in [1, 7].

| Gene | loci | Heritance | Protein | Pathway or effect |
|---|---|---|---|---|
| ANG | 14q11.2 | Dominant | Angiogenin | rRNA transcription |
| ALS2 | 2q33 | Recessive | Alsin | Endosome/membrane trafficking |
| C9ORF72 | 9p21.2 | Dominant | Uncharacterized | Altered C9ORF72 RNA splicing, formation of nuclear RNA foci |
| FIG4 | 6q21 | Recessive | FIG4 homolog | Endosome trafficking |
| FUS | 16p11.2 | Both | Fused in sarcoma | Altered RNA processing, formation of inclusion bodies |
| OPTN | 10p13 | Both | Optineurin | Golgi maintenance, membrane trafficking and exocytosis, formation of inclusion bodies |
| SETX | 9q34.12 | Dominant | Senataxin | DNA and RNA processing |
| SOD1 | 21q22.11 | Almost always | Superoxide dismutase-1 | Protein aggregation, possible gains of redox function, impaired axonal transport |
| SPG11 | 15q21.2 | Recessive | Spatacsin | Impaired axonal transport |
| TARDPB | 1p36.22 | Dominant | TAR DNA binding | RNA processing, formation of protein inclusion bodies |
| UBQLN2 | Xp11.231 dominant | X-linked | Ubiquilin-2 | Proteasomal protein degradation, inclusion body formation |
| VAPB | 20q13.32 | Dominant | Vesicle-associated membrane protein VAMP | Vesicle trafficking |
| VCP | 9p13.3 | Dominant | Valosin-containing protein | Proteasomal degradation, endosomal trafficking, vesicle sorting |

**Table 2.1:** Genetic factors and loci identified in previous studies by [1] and [8] as related with familial form of ALS.

## 2.2 Microarray Technology

The human genome has been sequenced as a result of the Human Genome Project started in 1990. In 2004, the sequenced human genome consisted of 2.85 billion nucleotides interrupted by 341 gaps, which covered approximately 99% of the genome with an error rate of approximately 1 event per 100,000 bases. By then, it was believed that there were only 20,000-25,000 protein-coding genes [14]. More recently, the human genome was described as presenting around 30,000 genes, which in turn represented the production of more than 100,000 proteins [15]. This high quantity of different proteins may be explained by a process called alternative splicing [15].The molecule pre-mRNA, contains exons and introns regions, and passes through a process where exons are removed in order to form mature mRNA which will be translated into a protein. However, during this process, exons may be kept and introns removed and alternative splicing takes place. The modification of genetic information may lead to different proteins, which may or not present abnormal function and result in a genetic disorder [15].

A molecule that may influence the final production of proteins is microRNA, small segments of RNA containing around 22 nucleotides [15]. They bind to their complementary mRNA sequence and facilitate their degradation, which results in repression of transcription. These molecules may be transcribed by their genes or exist in intronic regions of mRNA. There are estimated 1,000 miRNA encoded in the human genome that are predicted to bind to 60% of all mRNA transcripts. Therefore, miRNA and their dysregulation are pointed in a number of different diseases. Although not standardized and still in its beginning, there are already some miRNA microarrays to study their expression as its effects on gene expression [15].

Besides, some authors [16] have proposed genomic variance in the population, that resulted in different phenotypes and should be considered when studying gene expression. This variance may be due to single-nucleotide polymorphisms, small insertion-deletion polymorphisms, variable numbers of repetitive sequences, and genomic structural alterations [16]. In this context, understanding the human genome, its variance and regulation will contribute to a better understanding of the physiology of most diseases but also how evolution had occurred [15].

On the other hand, although several genomes have been sequenced, including the human, not all genes' function and location have yet been identified. Towards this end, high-throughput technologies that produce several types of 'omics' data have been used. As predicted in 2004 [17], the technology of microarrays had great impact in biomedical studies and it is nowadays a common practice.

A brief description of microarrays' technology and its applications in biomedical studies is provided in the next sections.

**Definition** The term microarray was empirically defined by [17] as "a high-throughput assay system which utilizes spatially ordered discrete, high-density arrangement of biologically important entities immobilized on a solid platform", where these entities may be nucleic acids fragments, proteins, carbohydrates, whole cells, or tissues. It had also been previously defined by [18] as "monolithic, flat surfaces that bear multiple probe sites, often hundreds and thousands, and each bear a reagent whose molecular recognition of a complementary molecule can lead to a signal that is detected by an imaging, most often fluorescence". These platforms might be used for assessment of the behaviour of multiple targets in a single platform.

The development of these arrays were due to the contribution of different publications, as reviewed in [17–19]. The first microarray commercialized was an oligonuleotide-based microarray called GeneChip® and made available by Affymetrix in 1996. Since then, several arrays have been developed, that take into consideration different concerns and improve efficiency and screen power of this initial one.

**Types of microarrays** What determines the type of microarray is its biological target and its complementary probe. The most commonly used type of microarray is the one applied to nucleic acids analysis in order to monitor gene expression levels, by parallel assessment of thousands of genes at the same time [18, 19]. These microarrays may in turn be divided into two types based on the material used as probe: (1) complementary DNA (or cDNA), and (2) oligonucleotide microarray [17, 20]. These two types of design have different probe and target preparation (schematized in Figure 2.1). cDNA microarrays' probes are PCR fragments amplified from specific target sequences or clones from a cDNA library. The length of such probes can vary from a few hundred base pairs (pb) to a few thousand long [17], cDNA microarrays have been found useful in non-model organism and in identifying heterologous genes across species [21]. On the other hand, oligonucleotides' microarrays do not use cDNA library as their probes are synthetized on-chip or first synthesized and then immobilized on the platform. These probes are considerably smaller, usually 20-80 mer long [17, 21].

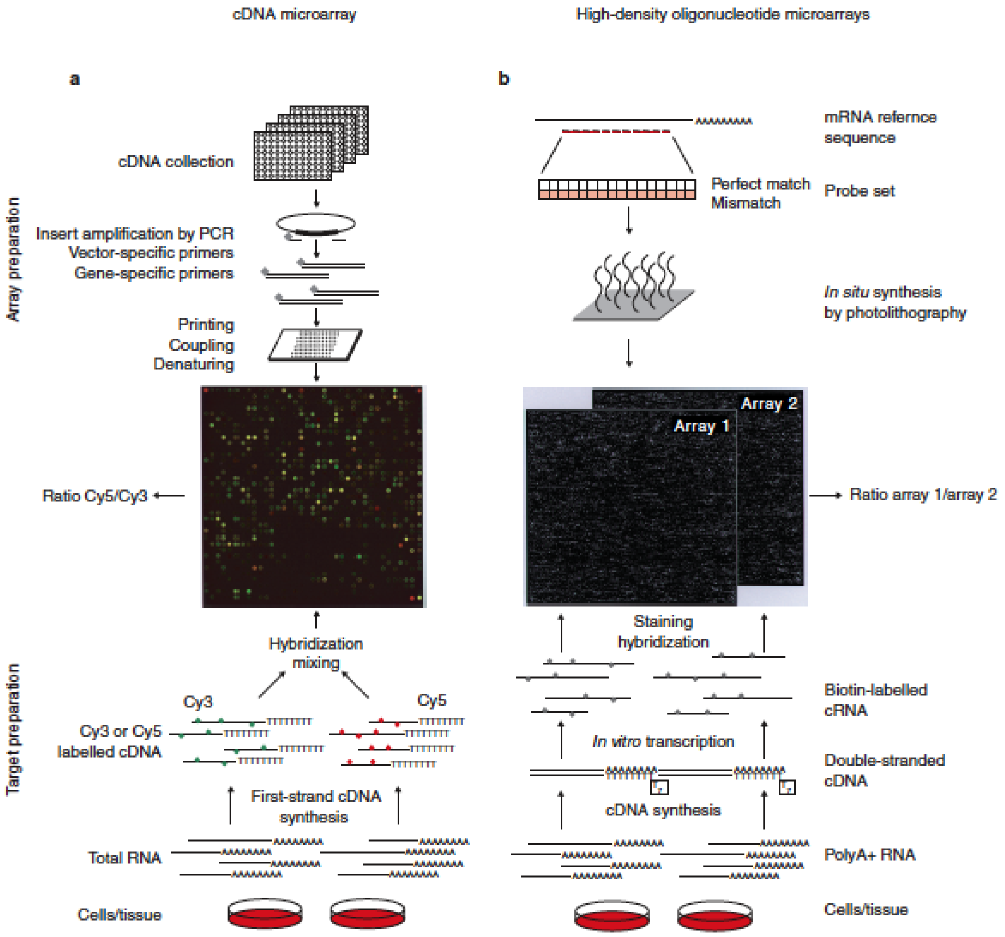


**Figure 2.1:** Schematic overview of nuclei acids microarrays (adapted from [20]): an overview of probe array and target preparation for spotted cDNA microarrays and high-density oligonucleotide microarrays. (a) cDNA microarrays. (b) High-density oligonucleotide microarrays.

However, these arrays are not enough to estimate all cell's activity and so, other types were created. Presently there are microarrays of proteins, antibodies, carbohydrates, entire cells and even tissues [17, 22]. Due to the different complexity and structure of these probes, such arrays face different challenges as the nucleic acids ones. For example, proteins three-dimensional structure is essential to its activity and proteins' chips have to be design to take this limitation into consideration. Good reviews on these arrays, their applications and challenges are available in [17, 22]. From this point further, only DNA microarrays are going to be discussed as they are the focus of the present work.

**Usage** The sample of DNA or RNA is first isolated from the biological sample and amplified. The sample is labelled with a fluorescent dye, such as Cy3 and Cy5, or radioactively. Then, specific hybridization is promoted between samples and their complementary probes attached to the microarray's surface and a double stranded molecular structure is formed. The chip is rinsed to remove non-specifically bounded molecules and the resulting fluorescent or radioactive image is recorded. Usually a laser scanner attached to a confocal microscope and a CCD camera is used in the case of fluorescent dyes. A fluorescent image is produced by the excitation of the labels by the laser, which is captured by the camera. On the other hand, hybridization is detected when using radioactive labels by a phosporimager [17]. Both hybridization images present information regarding where hybridization has successfully occur and the relative expression of a given gene is inferred by the intensity of the spot where its specific probes are located. Higher intensity is equivalent to high gene expression. Therefore, in a single experiment and, using an appropriate array, an entire genome may be scanned [22].

The underlying principle of microarray technology is the specific hybridization between two strands of DNA, in which hydrogen bonds are formed between complementary nucleotide base pairs [23]. Validation of the expression of specific sequences may be verified using conventional techniques like northern blot, RT-PCR, or nuclease protection assay [17].

**Applications** As this technology provides the means to measure a wide variety of messenger RNA transcripts, it allows a new insight into some of the cellular processes. The applications of microarrays are vast in clinical practice and genetic research as it offers the possibility to register gene expression response to external factors or differences in the individual's phenotype [21, 24]. Therefore, they can be applied in the research and diagnosis of diseases, pharmacogenetics, drug discovery, toxicogenomics, and functional genomics [17, 19, 21]. They may also be used in studies of gene expresion, genome mapping, SNP discrimination, transcription factor activity and many other [23].

In research, microarrays may be used to study genomic related diseases, such as cancer, in an attempt to identify, for example, specific molecular markers related to the different stages of the disease. They are considered a good alternative to the study of cognitive diseases where animal models fail. On the other hand, microarrays may be use in whole genome studies to screen for mutation point screening or in an attempt to identify SNPs [17], as well as study of non-diseased related disorders, such as drug addiction [21].

Relatively to the therapeutics, it may be used in the study of the influence of drugs in gene expression and screening for potential therapeutic candidates [17].

Currently, there are several public data repositories, where data from different experiments is shared between researchers, the two largest repositories are the Gene Expression Omnibus (GEO) from NCBI and ArrayExpress from EBI [25].

## 2.3 Microarray Data Analysis

Processing microarrays presents a basic pipeline that more or less is followed by every bioinformatician. These studies start with low-level operations, such as filtering and normalisation and finish with high-level ones, like clustering techniques or other pattern recognition algorithms [26]. Accordingly to [26] it is the low-level tasks that are poorly understood as most of them are empirically determined. This basic pipeline is briefly described in the following paragraphs, where the precise algorithms used at each step vary slightly between one-channel arrays (Affymetrix) and two-channel arrays (Illumina).

**Pipeline** [26] Each microarray used in a study is individually processed, that is, quantified, filtered and normalized. The hybridization image acquired in the experiment is transform into numeric data. This raw data is filtered to remove high-noise and low-signal spots. Then, a normalization step is required to remove spatial variability, channel imbalances and inter-array heterogeneity, which is often performed twice. These microarrays are usually combine into a dataset that requires a normalization inter-array. Processing this dataset may unravel important biological meanings hidden in the data. Differential expression analysis may be applied to identify individual genes that present a significantly changed expression profile between experiments and it is a common practice. This differential analysis step may be used to select the significant differentially expressed genes to be given as input to the high-level procedures in order to reduce the data dimensionality [23], but it is not mandatory. As an alternative or a complement, datasets are used as input into clustering analysis. The resulting subgroups of genes present a similar or related expression profile and dissimilar behaviour from genes contained in other subgroups or clusters. More recently, there has been made efforts to reproduce the natural genetic networks by the identification of how genes expression profiles are correlated [26].

Microarray studies, as they allow parallel assessment of thousands of genes, generate large quantity of data. Appropriate statistical and computational methods have thus been used to extract useful information from these studies. Several approaches have been proposed over the years to uncover biological meaning from microarray data but still many questions remained unanswered [27]. Besides, the results obtained with this technique would be more robust if the analysis would be made in a standardize way [21, 23].

Towards this end, genes may either be considered individually in a procedure known as differential expression analysis or in a more global way, using clustering/biclustering and classification techniques or network inference [23, 28]. The most common procedures are the first three, although in recent years other types of studies have been performed on this kind of data, such as transcription factor binding site analysis, pathway analysis, protein-protein interaction network analysis and gene enrichment analysis (further details, as well as a presentation of methods and tools available, on these latter methods, please refer to [23]).

There appear to be no consensus as to which is the best method to be used but rather that different techniques may explore different aspects of the data. Differential analysis (Section 2.3.1) is based on the concept that the modification of one genes' expression profile may be solely responsible for a given condition. On the other hand, co-expression and network analysis are used to study the potential transcriptional co-regulation and gene interactions. Practical applications of using such approaches are to unravel or validate disease subtypes, predict unknown gene functions or transcriptional regulations, dimension reduction and gene clustering [25].

### 2.3.1 Differential Analysis

Differential expression analysis focuses on the identification of genes (or pathways), which present significantly different expression profiles between two groups of experiments. The first used approach to determine such genes was a calculation of a fold change between the two groups. Folds above a given threshold, for example two-fold, were considered biologically relevant. Later, various univariate statistical methods started being used, such as t-tests, SAM (modified t-tests), two-sample t-tests, F-statistic, Bayesian models [23], p-value cuttoffs, Bonferroni corrections, Fishers exact test [21], Wilcoxon test and Kruskal-Wallis [27].

Analysis of Variance techniques (ANOVA) are chosen for datasets with multiple classes [21, 23]. Finally, the standard multi-testing procedure to select the final list of significantly differentially expressed genes is the false discovery rate (FDR) [29].

Interesting gene lists may result as output of these methods, which may be potential drug targets and biomarkers [23]. There can be a bias in the conclusions of these studies as many of them result in large lists of genes, the researchers frequently restrict their conclusions to the already established genes related with the condition under study [20]. It is also difficult to predict if genes are affected as a cause or a consequence of the disorder.

As another pitfall, the output of some of these methods, such as fold change and statistical cuttoffs, vary when the significance values are changes. This could lead to misinterpretation of the data [21].

### 2.3.2 Supervised Learning Methods

Supervised approaches are chosen to classify the samples into known classes, which requires prior knowledge of the data structure [26, 27]. The basic concept underlying this approach is to train the model with some examples in order for the classifier to know how to classify an unseen test case into one of classes [23]. The most interesting application would be to identify diseases types and sub-types in diagnosis [23], by finding a list of relevant genes that allow this classification [27]. The most commonly used methods for microarray data analysis using this approach are: random forests (RF), k-nearest neighbours (KNN), support vector machine (SVM), Artificial Neural Networks, weighted voting and discriminant analysis (DA) [23, 27]. The latter was described in the previous chapters (Section 2.3.1) and for more details on such methods, please refer to [24, 26, 30].

In most biological problems targeted by microarrays, the class labels, required for the supervised algorithms, are not provided [24]. Therefore, in microarray data analysis, unsupervised methods are more frequently used as they attempt to unravel novel or unexpected hidden patterns in the data [23, 26].

### 2.3.3 Unsupervised Learning Methods

There are numerous types of clustering algorithms that need to be selected based on the features one is more interested in. One may be interested in grouping genes expression profiles, samples or both[31]. When data reduction is required, PCA or SOMs [27] are taken into consideration, for example. Unfortunately, none of these approaches presents a clear superior performance regarding the others [30]. A relevant pitfall is that it has the property of identifying patterns even in random data and so, their output has to be validated statistically and experimentally [26].

**Why cluster genes?** In the past few years it has become clear genes and other cell's components, like proteins and RNA, interact with each other in order to respond to cell's biological functions. In fact, it is estimated that one gene interacts on average with four to eight other and is involved in ten biological functions [27]. Thus genes do not act on their own but rather in complex interactions in also complex regulatory networks [29].

Clustering genes makes the assumption that genes in the same cluster are co-expressed, that is, they present related expression patterns. Indeed genes coding for proteins that belong to some pathway or that are elements in a protein complex are likely to have similar expression patterns [32]. Therefore, the biological motivation to consider co-expressed genes *in silico* is that they are co-regulated *in vivo* [26] and that the magnitude of their co-expression gives the likelihood of their interaction [25]. In fact, it has been verified that gene expression clusters tend to be significantly enriched with specific functional categories [33], which may be used to infer the functionality of unknown genes belonging to these clusters using the principle of 'guilty-by-association' [34, 35]. Besides, there is a clear advantage in the data reduction of microarray studies as it makes it is easier to visualize, study and understand a few clusters than thousands of gene expression profiles [26, 33].

Studying co-expressed genes may contribute to the knowledge of regulatory mechanisms [29]. On the other hand, clustering may be also used to cluster samples or genes and samples [27]. Although there is evidence of these interactions among genes, and other components, which may be unravelled by pattern discovery techniques, *in vivo* or *in vitro* testing would provide a definitive evidence [26, 36].

Clustering algorithms are a data mining concept applicable to different contexts [28] and that may be divided into hierarchical and partitioning (or non-hierarchical) methods [33]. The three most common used clustering algorithms are hierarchical clustering and two partitioning techniques, k-means and self-organizing maps [23]. Over the years, several techniques, some of which derived from these three, have been widely applied to microarray data, such as K-medoid; Two way clustering; Density based clustering; Graph theoretic based clustering; Genetic K-Means, among others [28].

### 2.3.4 Coexpression Networks

Complex networks have been used to model several real network as they allow intuitive interpretation of the relations between its nodes [37]. Furthermore, one may take advantage of the extensively used and studied network concepts to understand these relations [38]. A network may be generally defined as a group of nodes, which may or not be connected pairwise.

Several network models have been proposed in an attempt to explain the topological properties of real networks, as random graphs [37], the small-world model of Watts and Strogatz [37], generalized random graphs [37], scale-free networks of Barabási and Albert [37]. Some of these models are of networks with some specific features, like geographic networks and networks with community structure [37].

To describe the interactions between genes that collaborate to engage cell functions, it may be used gene regulatory networks. Using such networks, as in other types of networks, allows an intuitive visualization of the relation between nodes, which are, in this case, genes [39]. Four methods are described in the literature to construct such types of networks [39]: (1) probabilistic networks-based approaches, namely Bayesian networks [27], (2) correlation-based methods, (3) partial-correlation methods, and (4) Information-theory-based methods (more information of these types is given in [39]). Gene expression microarrays with their power of monitoring expression of thousands of genes at the same time provide an useful tool to build this type of networks and explore gene relationships.

As motivated before, using correlation-based methods provides an intuitive approach to study the relationship between genes, which is also valid for gene co-expression networks. The first step in these type of methods is usually to define a similarity matrix, $S = [S_{i,j}]$, using the expression matrix. Each entry of the similarity matrix, $s_{i,j}$ corresponds to the pairwise transcription correlation coefficient between gene $i$ and $j$. Then a soft or hard threshold is applied to this matrix in order to obtain an adjacency matrix, a symmetric matrix where each entry, $a_{i,j}$, represents the strength of the connection between the nodes $i$ and $j$ and varies in the interval $[0, 1]$ (in the case of soft-thresholding) or takes value 0 or 1 (in the case of hard-thresholding). This matrix contains the biological meaningful relationships between genes and may be used to build a network, where connections between genes may be weighted (soft-thresholding) or unweighted (hard-thresholding). Weighted Gene Co-expression Network Analysis (WGCNA) [38] is a recent method that uses correlation-based methods and that has uncovered several disease-related genes [39].

**Scale-free Topology Networks** Metabolic networks are described as approximately presenting scale-free topology [40]. Where the defining property of a scale-free network is that the probability of a node to be connected to $k$ other, which is the degree distribution $p(k)$ of a network, decays as a power law $p(k) \sim k^{-\gamma}$ [38]. These networks are prone to be very heterogeneous and with their topology dominated by few highly connected nodes, or hubs, that link the rest of the less connected nodes to the system [37]. As a result, these networks are highly sensitive to the removal of these hubs, but highly resistant to random perturbations [38]. Another intrinsic characteristic is that the probability of a new node $i$ to be add to an existing one $j$ is proportional to the degree of $j$ and is given by $P(i \to j) = \frac{k_j}{\sum_u k_u}$. This is known as “the rich get richer paradigm” and it is thought to characterize the evolution of biological systems [37].

**Weighted Gene Co-expression Network Analysis (WGCNA)** In 2005, Zhang and Horvath [38], demonstrated that other types of cellular interactions, like gene co-expression networks and protein-

protein interaction also approximately follow scale-free topology. This was the motivation for the development of an algorithm to determine gene co-expression networks, known as Weighted Gene Co-expression Network Analysis (WGCNA) [38]. This procedure was implemented as a package of functions [41] in R project [42] that should be used following the basic pipeline defined in [38] but that do confer some flexibility of specific choices to the user. One may use this method to build a graph, where nodes correspond to genes and edges account for significant co-expression relationships. Namely, one may construct weighted or unweighted and signed or unsigned correlation networks. To identify modules, this method makes use of hierarchical clustering where a topological measure is used as a distance measure.

WGCNA also presents the advantage of including interfaces with different frequently used tools for biologic network visualization, such as VisANT and Cytoscape, and enrichment analysis (DAVID) [39]. Accordingly to [39], WGCNA method performed well in constructing gene regulatory networks.

**Why using networks?** In general, networks may be used to model cellular interactions [36]. In the case of gene co-expression, the underlying concept is the same of co-expression, as there exists consensus that genes interact in complex regulatory networks. As most common diseases are not simple Mendelian disorders but rather a complex modification of genes interactions, it is thought that studying the differences between the gene co-expression networks may be advantageous [29]. Moreover, networks have been used in many different biological contexts, such as study functional enrichment, analyse the structure of cellular networks, model biological signalling or regulatory networks, re-engineer metabolic networks and to understand the dynamic behaviour of gene regulatory networks [43].

### 2.3.5 Related Work

Saris et al. [5] studied gene relationships using an Amyotrophic Lateral Sclerosis dataset, comprising three sets of sporadic ALS patients and matching controls. For the first dataset (ALS and controls), named discovery set, they applied differential expression analysis making use of Student's t-test and for each probe in the dataset a statistical significance value (p-value) was computed and a false discovery rate of 0.05 was considered. From the initial 8,000 probes, only 2,300 were considered significantly differential expressed. Then, Random Trees and K-Nearest neighbour ($k = 10$) were used to predict ALS status based on gene expression profiles and proved to correctly classify 80% of the samples. The WGCNA was applied to this discovery dataset to obtain a weighted and unsigned network, given by an adjacency function $a_{ij} = |cor(x_i, x_j)|^\beta$, where the soft threshold power $\beta = 6$ was chosen using the scale free topology criterion. From this network construction process, resulted 5 co-expression modules. The procedure of network construction was repeated for two other datasets and two modules were considered highly preserved in the 3 datasets. Both these modules were enriched with the differentially expressed genes even when using the Bonferroni comparison. A multivariate Cox regression analysis that regressed survival time on the module eigengenes, site of onset, sex and age at onset, resulted in no significant p-values. Ingenuity pathway analysis was used to study the functional enrichment of the disease related modules and ALS related genes. Functional enrichment analysis with DAVID was applied only to the 100 most highly connected genes in the two identified modules, where the weighted connectivity was defined as the sum of all strength of connections shared by the gene.

Interested in studying the properties of weighted gene co-expression networks formed from multiple microarray datasets and the essentiality of hub genes, Carlon et al. [44] applied WGCNA to yeast microarray data. Networks were constructed using the same reasoning as in the previous work, with different soft threshold powers for each of the three datasets considered ($\beta = 10, 12, 18$). The relative importance of a gene was given by its weighted connectivity and it was clear there was a relationship between this value and the gene essentiality. To compare the networks, a hyper-geometric p-value was used to evaluate the significance of the overlap of the module genes across the 3 datasets and it was proposed a way to determine the upper limit of the p-value. As correlation between intramodular connectivity and essentiality was observed in all datasets, they concluded that modules obtained by this method present enough information to indicate which genes are essential to a particular function.

Oldham et al. [35] studied the difference in the functional organization of the transcriptome in human brain making use of WGCNA. Four datasets corresponding to different brain regions were composed with microarray data samples gathered from different published studies. Parallel construction of the gene co-expression network was performed on each of them using the power adjacency function with different soft threshold power for each of them ($\beta = 4, 5, 6$). Then, hierarchical clustering was used with average linkage and topological overlap as dissimilarity measure and the dendogram was cut using a dynamic tree-cutting algorithm [41] that allows some genes to remain unassigned. For each dataset, 19, 23 and 22 modules were identified. The measure of module membership was defined as the Pearson correlation between a given gene and the principal component obtained by singular value decomposition of the module (module eigengene [45]) it was assigned to. A module eigengene network [45], consisting of all pairwise Pearson correlation between module eigegene, was built to study the relationship between modules identified in the same network. To compare these modules between networks, the overlap between all possible pairs of modules was calculated along with the probability of observing such overlap by chance. One-sided hypergeometric test was used to assess the significance of module overlap between networks and a Bonferroni correction used to account for multiple comparisons. The color of modules with significant overlap, that is corrected hypergeometric P value $< 0.05$, was converted to be the same (in the case of overlapping with several modules, the least significant value was the one considered). Oldham et al. [35] tested the ability of topological overlap to predict know functional relationships in the human brain by verifying that mean topological overlap was significantly higher for interacting proteins than for randomly selected ones.

Mason et al. [46], applied WGCNA in the context of transcriptional regulation in murine embryonic stem cells. In this work it was demonstrated that for that specific problem, the signed weighted gene co-expression analysis (WGCNA) performed better than its unsigned counterpart as more significant modules were encountered. Two networks were built in parallel, using the power function to soft threshold the adjacency matrix, with $\beta = 6$ and $\beta = 12$ in the unsigned and signed network, respectively. Modules were determined in a similar way as described above. Instead of performing module overlap, the authors processed the two networks independently and verified if both were equally enriched with the relevant genes under study. The module eigengene was used to summarize the expression profiles of the identified modules and module membership was defined as intramodular connectivity as in [44] or module eigengene based connectivity as in [35]. A gene significance (GS) measure was defined as the t-statistic from the paired Student's t-test of expression between known genes associated with the case under study and their profile expression. A strong linear relationship between this measure and module eigengene based connectivity was obtained in two of the modules. Differential expression was also applied to this data, and some of the modules encountered using WGCNA presented a strong relationship between module membership and differential expression using fold change.

More recently, Miller et al. [47] studied the differences between human and mouse brain transcriptome in order to verify that modules are highly preserved between the two species and also found some robust human-specific modules. Gene expression and connectivity, the sum of all gene co-expression pairwise relationships in the network, also tend to be preserved between species. Two independent network were built using data using the most connected genes (5,000 and 3,000 in the case of human and mouse dataset, respectively). Then, genes were re-assigned to modules, based on a threshold of module eigengene membership, that allows module overlapping. It was verified an high preservation between the modules identified in both networks. Encountered modules were enriched and annotated using Gene Ontology (GO) and Ingenuity pathway analysis, which also confirmed the high overlapping between human and mouse modules.

### 2.3.6 Biological interpretation of the results

Analysing genes and gene products requires the integration of their functions in the cell and how their disturbed behaviour can contribute to a given clinical state. It was therefore used an annotation based on Gene Ontology terms [48] and also Kyoto Encyclopedia of Genes and Genomes (KEGG) [49].

The Gene ontology project [48] has the aim of standardizing the representation of gene and gene product attributes across species and databases. In that sense it was created three different ontologies, which provides vocabularies and classifications for molecular and cellular biology. These ontologies are cellular component, the parts of a cell or its extracellular environment; molecular function, the elemental activities of a gene product at the molecular level, such as binding or catalysis; and biological process, operations or sets of molecular events with a defined beginning and end, pertinent to the functioning of integrated living units: cells, tissues, organs, and organisms. The definition of a biological process is a series of events accomplished by one or more ordered assemblies of molecular functions. The GO ontology's terms are organized to form a directed acyclic graph (DAG), in which nodes correspond to individual terms. Each term could be connected to one or more parent node, less specific than it, and one or more child node, which presents a more specific definition. This ontology follows a true-path rule, a convention which states that whenever a gene is annotated to a term it is also implicitly associated with all the less specific parents of that term. The GO vocabulary is designed to be species-neutral, and includes terms applicable to prokaryotes and eukaryotes, single and multicellular organisms. It includes more that 20,000 terms each of which has an accession number, a name, a more detailed definition, and other information relating a term to its parent terms [48].

KEGG [49] is a dataset focused on the understanding of high-level functions of the biological system, such as the cell or the entire organism, and is based on genomic and molecular-level information. It consists of 16 main databases that are categorized into genomic (genes and proteins), chemical (chemical substances) and systems (relation networks) information. Graphical representation of these interacting parts is also available.

The Babelomics [50] is an online platform that allows the integration of both tools, GO and KEGG, and was used in this work to annotate the group of genes identified in this work.

# 3 Methods

Contents

## 3.1 Data

Data used in this work corresponds to a microarray dataset comprising samples from sporadic Amyotrophic Lateral Sclerosis (ALS) patients and their matching controls, also used in [5], and made available by the authors in .csv format. Patients were diagnosed accordingly to the El Escorial Criteria for 'probable' and 'definite' ALS. Some clinical information on the patients is also provided, such as gender and age of disease onset, defined as the time of initial weakness, dysarthria or dysphagia. Controls are genetically unrelated individuals accompanying the patient during their outpatient visits.

Messenger RNA was obtained from peripheral blood samples and the gene expression data acquired using Illumina Sentrix HumanRef-8 Expression BeadChip with more than 22,000 current RefSeq curated gene targets [5]. Previous to the present work, this data was preprocessed using cubic spline normalization in Illumina's software package Beadstudio and imported into R. Only one third of the probes (8000) were found significantly expressed in peripheral blood at measurable levels (Bead studio mean detection level of $p < 0.05$). These probes were selected as the starting point of the co-expression network analysis.

The complete dataset is divided into three subsets of ALS patients and their matching controls. However, in this work only the first two were used and correspond to 30 patients and corresponding controls. These datasets were designed to be similar regarding proportion of female/male patients, average of ages and proportion of spinal/bulbar onset in patients.

**Table 3.1:** Clinical Information regarding the dataset used in this study.

| Clinical Variables | Dataset | Patients | Controls |
|---|---|---|---|
| Number | 1 | 30 | 30 |
| | 2 | 30 | 30 |
| Male Gender (%) | 1 | 15 (50) | 15 (50) |
| | 2 | 14 (47) | 15 (50) |
| Age at blood collection | 1 | 63.8 (41.0-76.0) | 62.8 (42.8-80.8) |
| | 2 | 63.7 (35.3-79.5) | 64.8 (36.2-75.8) |
| Age at disease onset | 1 | 62.8 (40.5-75.6) | |
| | 2 | 62.5 (34.2-78.4) | |
| Bulbar Onset (%) | 1 | 15 (50) | |
| | 2 | 15 (50) | |

## 3.2 Software

Most of the procedures carried out in this work where performed in R version 2.14.2 (2012-02-29), a freely available online tool specially designed for statistical studies [42]. The motivation to use such platform was the fact that the Weighted Gene Coexpression Network Analysis (WGCNA) method was developed as a R package [41]. This package contains several functions concerning different steps of microarray processing analysis. The Genesis 1.0 software [51] was also used in some of the procedures, specially in (Section 4.1). Also freely available, it is an independent Java suite, which integrates several tools for processing microarrays. For functional annotation of the results, an online platform, Babelomics [50], was selected as it integrates both GO terms and KEGG. For exploring and processing microarray data, other data mining tools have also been developed, a recent review on this topic is performed in [23].

## 3.3 Data Preprocessing

As referred by [26], the preprocessing steps are the ones less understood but they have great influence in the determination of the biologically relevant classes [52].

Generally, when working with microarray data, two types of preprocessing are required. First, one has to use image processing techniques to acquire expression values from the microarray images [30]. Usually the microarray scanner manufacturers provide software to deal with this step of the procedure although some other alternatives are available by researchers interested in improving the expression values estimates [24]. Then, one has to decide how to handle missing values and standardize the data in order to obtain expression values comparable across chips [24], either from different platforms or simply to reduce errors associated with the experiments. These pre-processing steps might affect the ability to identify biologically relevant classes [52].

To produce more updated results, the probes identification number was converted by a recent available online tool named Array Information Library Universal Navigator (AILUN) ([53]) instead of using the provided by the authors [5].

Gene expression values in microarrays are obtained by its representing probe sets and a single gene may be represented by more than one. Correctly combining the information of these probes, which may present inconsistent or contradictory results, is essential to obtain the most accurate gene expression value. Three approaches are proposed in the literature to solve this issue: (1) average expression values of all probe sets mapping for that gene; (2) create a way to redefine this mapping in such way that there is a single correspondence between genes and redefined probe sets; (3) select the most representative probe set for each gene [54]. Each of them present clear disadvantages, using the first one may introduce noise of the possible outliers probe sets, the second may become too complicated when dealing with more than one dataset. The third one is considered to be as simpler as the first and may be more accurate [54]. A method, part of the WGCNA package, (*collapseRows()*) is also easily interpreted and used in this context and allows to choose: (1) the probe set with highest (the default) or (2) lowest mean value; (3) the probe set with the highest variance; the probe with the (4) highest or (5) lowest mean absolute for each ; (6) the first principal component of the probe sets in the sample or (7) the simple average. In this work, the standard, which corresponds to the average [52] was used.

### 3.3.1 Gene Selection

As mentioned, gene selection may be used before high-dimensional procedures. The motivation behind this choice is to eliminate possible noisy genes and aliviate the computational burden of the analysis, without the loss of information [30]. Several methods have been proposed to perform this task (for further details see [52]). In the present work, all genes in the dataset were included in the analysis in an attempt to assess if the following processing steps were robust to this noise.

### 3.3.2 Standardization

Some of the procedures that will further be applied to this data, like clustering, require the standardization of the data [30, 52]. Therefore, for consistency, it was applied as a global pre-processing step. Data was first normalized by sample such that each sample presents zero mean and unit as standard deviation. Then a normalization by genes was performed.

### 3.3.3 Identifying samples' outliers

Several methods are available to identify outliers' in the samples of a microarray study in order to reduce the variability in the experiments. A common approach is to apply clustering techniques, usually hierarchical clustering to group similar samples and to identify dissimilar ones. In the WGCNA literature, hierarchical clustering with Pearson correlation is used to obtain a so-called Inter-Array Correlation (see [35]).

**Sample Network** Also in the WGCNA literature is stated a recent method that, based on the network properties of each sample, identifies samples' outliers. In fact, Oldham et al. [55] stressed out the importance of studying sample significance before proceeding to the analysis of genomic data. This study resulted in a standardized platform implemented in R project to explore the samples network relationships. A sample network is built, by defining a network adjacency measure between samples, which is useful for not only sample detection but also class comparison. A sample that presents an overall connectivity significantly lower than all other samples' in the dataset is identified as a potential outlier. Instead of using this method to simply detect outliers, one may be interested in exploring the significant differences between samples or classes of samples. This method is further detailed in Section 3.6.3.

## 3.4 Data Analysis

### 3.4.1 Clustering Techniques

Clustering techniques or unsupervised methods are a common tool to extract information from microarray's studies, either by grouping genes, samples or both. The goal is to reduce the data by grouping similar objects into the same cluster in such way objects in the same group are highly similar among each other and dissimilar to the ones grouped in different ones [52]. Therefore, whenever one uses clustering two main decisions are required: how to define the similarity between objects and how to use this similarity to cluster objects. Frequently, these decisions may be taken independently, which may be the reason for the variety of clustering approaches [33].

As no clustering algorithm clearly outperforms all others in every context, it is suggested to use more than one clustering algorithm. Additionally, algorithms that may present different results when the initial conditions change, should be rerun multiple times to check for stability and find the best possible solutions. One should pay attention to the fact that most clustering algorithms may find patterns in random data even when no structure is present and so should be tested in random data also or validated with external information [33].

In what follows, we first present the most common similarity measures used in clustering algorithms, which may be divided into metric and semi-metric measures [24].

**Metric Distances** Metric distances follow four basic rules and must be:

1. A positive measure, $d_{ij} \geq 0$
2. Symmetric, $d_{ij} = d_{ji}$
3. Zero if it is self measure, $d_{ii}$ =0
4. Respect the triangle rule, when considering three objects, $i,\ j,\ k$ the distance from $i$ to $k$ is always less than or equal to the sum of the distance from $i$ to $j$, and the distance from $j$ to $k$, $d_{ik} \leq d_{ij} + d_{jk}$

The $x$ and $y$ in the following equations are $n$-vectors of measurements on the objects to be clustered. In the case of microarray datasets for clustering of genes, $x_i$ and $y_i$ correspond to the expression values for each gene $X$ and $Y$ in each of the experiments [24].

The standard metric distance used is the Euclidean Distance [24, 28], which is a generalization of the Pythagorean theorem. Initially defined in 3-dimensional space, may be extended to higher dimensional spaces.

$$d = \|x - y\| = \sqrt{\textstyle\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.1}$$

Other examples of metric distances is the squared version of the Euclidean distance and the Manhattan distance [31].

**Semi-metric Distances** Distance measures that are semi-metrics do not follow the fourth rule previously listed. There are several of these distances to calculate the similarity between gene expression values. The most used one is the Pearson correlation coefficient (or centred Pearson correlation coefficient)[24, 28], $r$, which is given by:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{3.2}$$

where $\bar{x}$ e $\bar{y}$ are, respectively, the mean values for the $X$ and $Y$ objects. This correlation ranges between $-1$ and $1$, taking the value zero when two vectors are completely independent from each other, which means that they are uncorrelated or orthogonal vectors. The maximum ($r = 1$) and minimum ($r = -1$) value occurs when the vectors are, respectively, identical (perfect correlation) or exact opposites (perfect anti-correlation). The measure of dissimilarity, using these semi-metrics, is given by $1-r$. As an advantage, this measure is invariant to changes in location or scale of either $x$ or

$y$, and thus must be used when one is interested in the 'shape' of the expression vector rather than its magnitude [24]. The distance metrics based on the Pearson correlation are given by $D_{ij} = (1 - r_{ij})/2$ [28].

Variations of this measure include an uncentered Pearson correlation coefficient, that should be considered when the relative expression level is relevant [28]. However, the latter two distances do not predict when two genes' expression levels are anti-correlated and thus if one is interested in this type of relation too, the squared correlation of Pearson should be used. It ranges from 0 to 1, where 0 still corresponds to uncorrelated vectors, but 1 can correspond to both perfectly correlated or anti-correlated expression vectors [28].

$$r = \left( \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \right)^2 \tag{3.3}$$

These latter dissimilarity measures (1 - correlation) may be related to the Euclidean distance. In fact, considering that $\tilde{x} = \frac{x - \bar{x}}{\sqrt{\sum_i (x_i - \bar{x})^2 / n}}$ and $\tilde{y} = \frac{y - \bar{y}}{\sqrt{\sum_i (y_i - \bar{y})^2 / n}}$ , then $\|\tilde{x} - \tilde{y}\|^2 = 2n(1 - r)$, which means that the squared Euclidean distance for standardized objects is proportion to the correlation of the original objects [31].

To choose a similarity measure, one has to decide what is the definition of a similar gene profile in order to guarantee that these genes are placed in the same cluster. It is clear in the literature, that the two most commonly used are Euclidean distance and the Pearson correlation coefficient [28]. The Pearson correlation has the property of being location/scale invariant, which will disregard changes in the average measurement level or range of measurements between samples [31], whilst the euclidean distance is sensitive to those [33]. The absolute Pearson correlation is considered the standard co-expression measure and it is commonly used in gene expression cluster analysis [38].

**Topological Overlap** This measure was initially proposed to measure the relatedness of the substrates forming a metabolic network [40]. However, since genetic and protein domain network also present an approximate scale-free topology, this framework was extended to these types of networks [38].

$$w_{ij} = \frac{\sum_u a_{iu} a_{uj} + a_{ij}}{min \left\{ \sum_u a_{iu}, \sum_u a_{ju} \right\} + 1 - a_{ij}} \tag{3.4}$$

It was initially proposed for unweighted networks, where $a_{ij}$ is binary, but the formula was later extended by [38] to the weighted also. In the latter case, the values of the adjacency matrix vary between $0 \leq a_{ij} \leq 1$, which implies that $0 \leq w_{ij} \leq 1$. It takes the maximum value when two nodes are connected and share exactly the same neighbours. The opposite situation, that is when it takes the value zero, occurs when they are not connected and share no neighbours [38].

The three most popular algorithms used for clustering microarrays are the hierarchical clustering, k-means and self-organizing maps (SOM) [33]. The public available statistics software R [42] has many implemented algorithms useful to process microarrays. There are many cluster analysis packages, such as SOM, hierarchical clustering, k-means, among others [27].

**Hierarchical Clustering** Hierarchical clustering depends on the computation of a similarity matrix between all objects being clustered, which is then used to guide the algorithm. This method may be an agglomerative or divisive procedure. The agglomerative approach is the most commonly used and thus will be characterized in this work [23].

The distance matrix is first computed to reflect the pairwise distances of all objects in the dataset, where all objects are mapped to a singleton cluster. Usually the euclidean distance is the one taken in this step [28]. Then the two closest clusters are merged and the distance matrix is recalculated. To compute the distance between two clusters, one of the following inter-cluster measures are chosen. This process is finalized when all objects belong to the same cluster.

The **Hierarchical clustering algorithm** is as follows [56].

Given a object matrix and a user-defined similarity measure and inter-cluster distance:

1. Compute the similarity matrix by determining the distances between all pairs of objects;
2. Merge the two closest objects $r$ and $s$;
3. Replace $r$ with the new cluster and delete $s$;
4. Repeat step 1 to 3 until number of clusters equals 1.

The most commonly used inter-cluster distance measure is the average-linkage [24], which generally works well with standardized microarray data [30]. It has the advantage of being insensitive to outliers, whilst the complete linkage is very sensitive to outliers [28]. According to [33] and [28] the most common inter-cluster distances are single, complete, average and centroid linkage. These measures give the distance between two clusters and may be defined as:

- single linkage (also known as nearest neighbour linkage) - minimum distance between all possible pairs of objects, one from each cluster;
- complete linkage - opposite to the single linkage as it takes the maximum of these distances. It is sensitive to outliers;
- average linkage (UPGMA - unweighted pair-group method using arithmetic averages) - average of distance between all possible pair of objects;
- centroid linkage (UPGMC - unweighted pair-group method using centroids) - distance between the centroids of the two clusters.

Hierarchical clustering using average, single or complete linkage to measure inter-cluster distances produces different results [55]. In [56] and [24] another method is also described, that is Ward's Method, which consists on the computation of the total sum of squared deviations from the mean of a cluster in a way it minimizes the increase in the sum of squared errors. In these methods, clusters are only obtained by cutting of the resulting tree (current methods are discussed in 3.4.1).

Visualization of the hierarchical clustering is done on a dendrogram. Therefore, when the number of objects being clustered is too large, it may be impractical to observe the organization of the tree [55].

Branch or tree cutting (or dendrogram pruning) is essential to detect clusters, normally corresponding to branches of the tree. The most commonly used is a fixed height branch cut in which the tree is cut at an user-defined height. The complete branches below that height are considered separated clusters. Although very efficient in many applications, it does not recognize nested clusters that are usually presented in complex dendrograms, such as gene co-expression networks [57].

Langfelder et al. [41] proposed two approaches of dynamic tree cutting to take these nested clusters into consideration. Two approaches were defined for this dynamic tree cut, a top-down algorithm, “Dynamic Tree” cut, and a bottom-up, the “Dynamic Hybrid cut”. The former respects the order by which the objects were placed on the tree and which has been successfully used for determination of significant modules in several species gene co-expression networks. This algorithm consists on setting a fix high height (typically of the interval of heights in the dendrogram), which will result in large clusters and a number of cluster-based dendrograms $\Omega = H_1, H_2, ..., H_m$. A dynamic and iterative algorithm is then applied to each of the dendrograms present in $\Omega$ in order to identify sub-clusters, which are added to this list of dendrograms and recursively decomposed until the number of clusters is stable. The user may define a minimum number of objects per cluster and a combination of the clusters which do not fulfil this requirement is performed [57].

In this work, whenever an hierarchical dendrogram was produced and there was interest in cutting the tree and study the resulting clusters, the Dynamic Hybrid Cut method was used. This approach is a bottom-ups cluster assembling divided into two steps. First, primarily clusters are identified by selecting branches that satisfy a given criteria. Then, a dissimilarity measure is used to try to map the unassigned objects to these clusters. If these objects are considered too distant, they remained unassigned and are coloured grey. In this latter step a partitioning based approach is applied that can be considered a modified k-medoid partitioning (also known as PAM). Therefore, the hybrid version, complements the information in the dendrogram by taking into consideration the dissimilarity among the objects, which makes it more sensible to the detection of outliers and improve cluster quality [57].

The **Dynamic Hybrid Cut Algorithm** [57] is as follows.

Consider a dendrogram $\Omega$ and the following definitions:

- the cluster core is the lowest-merged objects in the cluster;
- scatter of cluster core, $\bar{d}$ - the average of all pairwise dissimilarities between objects belonging to this core;
- cluster gap $g$ - difference between the scatter of the cluster core and the joining height of the cluster to the rest of the dendogram;
- cluster radius - maximum of the average dissimilarity of each object with the rest of the cluster (when the dissimilarity measure is the average); maximum of the dissimilarities of the cluster’s medoid to all other objects in the cluster (when using medoids);
- The clusters that are considered too small may be entirely merged to another cluster or all of its objects kept unassigned (this option may be turned off).

Set some user-defined parameters:

- $N_0$ - the minimum cluster size;
- $h_{max}$ - the maximum joining height;
- $g_{min}$ - the minimum gap;
- $d_{max}$ - maximum scatter;
- dissimilarity measure, $ds$- can be either (1) average dissimilarities of unlabeled objects to clusters (recommended) or (2) object-medoid distantances (medoid-medoid in the case of small unassigned objects);
- $ds_{max}$ - maximum dissimilarity;
- deepSplit - control of relative sensitivity of cluster splitting.

Apply the algorithm:

1. Selecting branches of the dendogram that:
   (a) contain a minimum number of objects given by $n_c = min\{int(N_0/2 + \sqrt{N - N_0/2}\}$, where $N$ represents the total number of objects in the branch;
   (b) all joining heights are less than $h_{max}$;
   (c) the cluster gap is more than a minimum gap $g_{min}$;
   (d) $\bar{d}$ is at most $d_{max}$.
2. For each unassigned objects and branches excluded in step 1, do:
   (a) compute the dissimilarity measure;
   (b) if dissimilarity measure is less than $ds_{max}$ and smaller than the cluster radius, then assign object to cluster.

This algorithm does not provide a way to choose the optimal partition method but rather offered a flexibility in the choice of parameters by the user [41].

The parameter *deepSplit*, implemented for both algorithms, allows a rough control over sensitivity to cluster splitting. In the hybrid algorithm it may take integer values from 0 to 4, where zero will produce a smaller number of large and well-defined clusters. Increasing this value will progressively result in a larger number of clusters, which may in turn present larger core scatter and be separated by smaller gaps. The authors advice reducing the number of minimum cluster size when using higher values of $deepSplit$ [57].

The PAM-like stage of assignment is not advised when one is interested in specificity over sensitivity, or at least set the maximum object-cluster dissimilarity to zero. This would produce tight clusters with few misclassification. Contrarily, if one is more interested in increasing sensitivity than specificity, this option should be selected and use a high object-cluster maximum dissimilarity (the default, this measure to be equal to the maximum joining height $h_{min}$ is usually enough) [57].

The motivation behind the use of this algorithm was the flexibility to choose the parameters, its ability to identified nested clusters and also the fact it does not force all objects to be attributed to a cluster which makes it less susceptible to outliers [41]. The default parameters in both functions have work well in past applications by the authors [41], but in the present work were varied in other to understand how they influence the partition of the presenting dataset. The $deeepSplit$ values were varied from 0 to 4 and seven values (4, 8, 16, 32, 64, 128, 256) of the minimum cluster size were selected.

**K-Means** Given $X = X_1, X_2, ..., X_n$ as the set of genes, the goal of K-Means is to divide this $n$ objects for $k$ (positive integer) number of clusters. Each of these objects are represented by $X_i = x_{i1}, x_{i2}, , ..., x_{im}$ that represent its value through the experimental conditions $m$. To test all possible partitions for a given dataset in order to select the best one is impracticable [58]. Therefore, the minimization of a given cost function is used to guide the algorithm. The commonly used cost function [56, 58] is the trace of the within cluster dispersion matrix, which can be defined as:

$$E = \sum_{l=1}^{k} \sum_{i=1}^{n} y_{il} d(X_i, Q_j) \tag{3.5}$$

Where $n$ is the number of objects in a data $X$, $Q_l$ is the mean of cluster $l$, $y_{il}$ is an element of a partition matrix $Y_{nxl}$ and the $d$ is a similarity measure often defined as the squared Euclidean distance [59]. The $Y$ presents two basic properties: (1) $0 \leq y_{il} < 1$ and (2) $\sum_{l=1}^{k} y_{il} = 1.$ If $y_{il} \epsilon \{0, 1\}$ then $Y$ is called an hard partition. Otherwise, it is a fuzzy partition (for further details, please refer to [58]).

The **K-Means Algorithm** is as follows [59]:

1. Random selection of the initial $k$ means for $k$ clusters;
2. Computation of the dissimilarity between an object and the mean of a cluster;
3. Mapping of objects to the clusters whose mean is nearest to the object;
4. Re-calculation of the mean of a cluster from the objects allocated to it;
5. Repeat step 2 to 4, until convergence.

### 3.4.2 WGCNA

Weighted Gene Co-expression Network Analysis (WGCNA) [38] was the method chosen to built a weighted network on ALS dataset. This network presents a scale-free topology, where nodes represent genes, which are connected if there is significant evidence of their co-expression.

Data is assumed to have been pre-processed, i.e., quantified and normalized in an appropriate way. Commonly the dataset is restricted to a subset of genes, for example the most varying, to remove irrelevant genes and decrease the computational burden of the analysis.

This methodology offers flexibility and the user must make several choices along the way. First, co-expression similarities are computed for every gene in the dataset and presented in a similarity matrix($n \times n$), with values ranging from $0$ to $1$. Then, an adjacency function is applied to this matrix to convert these similarities into connection strengths. This step is crucial to define if the network would be unweighted or weighted depending on the adjacency function used, which will perform, respectively, hard or soft thresholding of the matrix. As a defining property this function is monotonically increasing and maps from the interval $[0, 1]$ into $[0, 1]$. The choice of these parameters will be further motivated later.

The algorithm **Weighted Gene Co-expression Network Analysis** is defined as it follows [38].

Given a dataset of microarray samples.

1. Define a gene co-expression similarity: (a) absolute value of the Pearson correlation ($|cor(x_i, x_j)|$); (b) jacknifed correlation coefficient, (c) $s_{ij} = \frac{1+cor(i.j)}{2}$ or other;
2. Apply this similarity measure for each pair of genes $i$ and $j$: $s_{ij}$;
3. Represent the similarity results in a $n \times n$ similarity matrix $S = [s_{ij}]$ with values in the interval $[0, 1]$;
4. Define a family of adjacency functions: (a) hard thresholding using signum function or soft thresholding using (b) sigmoid or (c) power function:
   (a) signum function: $a_{ij} = signum(s_{ij}, \tau) = \begin{cases} 1 & if \quad s_{ij} \geq \tau \\ 0 & if \quad s_{ij} < \tau \end{cases}$
   (b) sigmoid function: $a_{ij} = sigmoid(s_{ij}, \alpha, \tau_0) = \frac{1}{1+e^{-\alpha(s_{ij}-\tau_o)}}$
   (c) power function: $a_{ij} = power(s_{ij}, \beta) \equiv |s_{ij}|^\beta$
5. Determine the Adjacency Functions Parameters;
6. Define a Measure of Node Dissimilarity: (a) make use of TOM dissilarity measure: $d_{ij}^w = 1 - w_{ij}$;
7. Identify Network Modules: Hierarchical clustering using node dissimilarity as distance metric and apply a tree cutting method;
8. Relate Network Concepts to Each Other;
9. Relate the Network Concepts to External Gene or Sample Information.

Usually, in WGCNA applications, the adjacency matrix is defined using the power function, and produces a signed or unsigned weighted adjacency matrix, whether function 3.6 or 3.7 is used, respectively. Both these type of networks have been used in the context of WGCNA, as a signed example is the work performed by [46].

$$a_ij = (0.5 + 0.5cor(x_i, x_j))^\beta \tag{3.6}$$

$$a_ij = |cor(x_i, x_j)|^\beta \tag{3.7}$$

Signed networks preserve better the continuous nature of correlations as they preserve the sign information and equally robust with respect to the soft threshold power parameter [55]. Besides, a signed correlation network may be consider equivalent to a network based on the Euclidean distance between scaled vectors [55].

The definition of modules is the one from Ravasz et al. [40], where a module are groups of nodes with topological overlap, but other definitions of modules are stated in the literature [60]. According to [60], the topological overlap may be considered as a filter of the spurious or missing connections between network nodes.

**Extensions of the topological overlap measure** The WGCNA method may be perceived as a top-down approach to identified clusters, where relevance is given to the neighbourhood of genes by the use of topological overlap measure (TOM). Other two extensions of the TOM have been proposed [60], the Multi-Node Topological Overlap Measure (MTOM), a bottom-up approach; and Generalized Topological Overlap Measure, where higher level neighbours are considered when comparing the relation between two genes in contrast to the one-level neighbour analysis performed by TOM [27].

**How to choose the soft thresholding parameter** When making use of soft thresholding, one has to decide which value to use as $\beta$. Zhang et al. [38] proposed a scale-free topology criterion. In a logarithmic scale, the weighted network adjacency is linearly related to the co-expression similarity as $log(a_{ij}) = \beta \times log(s_{ij})$. Usually, the choice of this parameter is based on the scale free topology fitting index $R^2$ that should be as close to unit as possible, without reducing the mean connectivity too much. This linear regression model fitting index or its truncated version to fit a linear regression line into the plot of $log_{10}(p(k))$ vs $log_{10}(k)$ [38, 44].

## 3.5 Practical Decisions

There is no clear consensus regarding the best clustering technique as it may also depend on the dataset itself [33]. The most common partitioning algorithm, K-means, was chosen for its efficiency and to provide a different approach to the WGCNA method, since the identification of clusters in the latter method is based on an hierarchical clustering procedure. Hierarchical clustering was also used in this dataset to cluster both genes and samples.

Relatively to the network construction, using WGCNA, it was chosen a step-by-step approach, in order to better understand the underlying concept. As the soft thresholding allows for more biologically relevant modules and the results are robust to the choice of parameter $\beta$ [38], it has the chosen approach.

## 3.6 Evaluation of WGCNA results

When using clustering procedures, one may use the internal structure of clusters by analysing the properties of the cluster. Another approach is to use external data that was not considered in the clustering procedure, such as gene ontology enrichment [33]. The internal quality of clusters are to be as compact or cohesive as possible as well as being clearly separated from others. In this work, as the motivation is to identify clusters of genes belonging to the same regulatory network, network concepts will be applied to evaluate the quality of clusters generated the WGCNA method.

### 3.6.1 Eigengene

The singular value decomposition ($X = UDVT$) is used to determine the principal component of each cluster and determine the so-called module eigengene. The module eigengene is the first principal component of gene expression of each module and is the dimension which better explains most of the gene expression variance in each module [61]. The motivation behind the use of this measure, is that WGCNA assigns gene to a single cluster when in fact it may be involved or participating in multiple functional pathways [35]. Therefore, this expression vector, module eigengene, attempts to represent the cluster and is correlated to the ones from other modules to analyse cluster determination efficiency. Determination of an eigengene network, where each module is represented by an eigengene, results in coexpression modules more rich than a catalogue of module memberships, that is traduced by the profiles of individual genes [27].

### 3.6.2 Network Concepts

In the complex network literature [37], there are different types of measurements to evaluate distinct characteristics of the network. However, only some of these measurements were found relevant for the study of gene co-expression networks and that are present in the associated literature: (1) clustering, cycle and rich-club coefficient to study the module structures in the network; (2) degree distribution and correlations; and (3) centrality measurements, as nodes with higher number of connection are considered more important.

The key concept of networks is the node **connectivity**, or degree, as it measures the relative importance of the node in the network [37] and it is most widely used measure to distinguish network nodes [55]. It has been found relevant in biological applications, for example, to identify significant genes in cancer and primate brain development [43]. The higher this value, the higher the importance of a given gene in the network. Genes that are highly connected are named 'hub' genes and are thought to play an important role in the structure of this biological networks [43].

$$k_i = \sum_j a_{ij} = \sum_i a_{ij} \tag{3.8}$$

The **maximum connectivity** is thus given by, $k_{max} = \max_i(k_j)$ and the **scaled connectivity**, $K_i$, by $K_i = \frac{K_i}{k_{max}}$ [55]. An also interesting measure, is the **standardized connectivity**, $Z.K_i$, of $i$-th node:

$$Z.K_i = \frac{K_i - mean(K)}{\sqrt{var(K)}} \tag{3.9}$$

However, **connectivity** or centrality, in the case of a WGCNA network, may also be defined as $k(i) = \sum_{j\epsilon N, j\neq i} |Cor(x_{i,}, x_j)|^\beta$, where $N$ refers to the total number of genes in the network [5]. It can also be applied as intramodular connectivity $k^q(i)$, as a more biologically significance measure, and may be defined in different ways:

1. Intramodular Connectivity [44, 46]: $k(i)^q = a_{i1} + a_{i2} + ...a_{in}(q)$;
2. Simple Module Membership [5]: $k(i)^q = \sum_{j\epsilon q, j\neq i} |Cor(x_{i,}, x_j)|^\beta$, where $q$ refers to a specific module and gives a measure of the relation between a gene $i$ and all other genes in its module;
3. Fuzzy Module Membership Measure or Eigengene-based Connectivity [5, 35, 46] measures the correlation between the $i$-th gene and the $q$-th module eigengene and is defined as $MM^q(i) = Cor(x_i, ME^q)$ .

These three measures may be interpreted as the higher its value, the more centrally located is the gene in the module [46]. Besides the first and last measure are highly correlated [46]. The Eigengene-based Connectivity presents several advantages, such as (1) is naturally scaled to present values between $-1$ and $1$; (2) a correlation test may be used to compute the p-value for a given gene's module membership; (3) may be calculated for any gene in the dataset that may not belong to that module and (4) in the case of signed networks, it could be used to identify genes that are anti-correlated with a given module eigengene [46].

Initially, Zhang and Horvath [38] proposed four measures to weighted networks: connectivity, TOM-based connectivity, intramodular connectivity, clustering coefficient. Several fundamental network concepts were then added to the WGCNA literature, by Dong [43] like line density, centralization and heterogeneity, and other authors (refer to [27]). In this work, it was found more interesting to explore the network concepts that had already been extended to WGCNA.

Most of these network concepts are applied to the adjacency matrix, which represents the network. As suggested in [43], the topological overlap matrix may be used instead of the adjacency matrix when computing the network concepts as it presents the same algebraic properties as the original matrix (symmetric and values between $0$ and $1$) and may be useful when dealing with sparse adjacency matrix. However, in gene co-expression networks, as it is essential to obtain high specificity for measuring interconnectedness, as the case of this work, the original adjacency matrix was used.

**Line density**, or mean (off-diagonal) adjacency measure, is closely related to the mean connectivity and is defined as [43]:

$$Density = \frac{\sum_i \sum_{j\neq i} a_{ij}}{n(n-1)} = \frac{mean(k)}{n-1} \tag{3.10}$$

The number of genes is always represented by $n$.

**Centralization** has been used to describe structural differences of metabolic networks [43]:

$$Centralization = \frac{n}{n-2}(\frac{max(k)}{n-1} - Density) \tag{3.11}$$

The **heterogeneity** measure studies the variation of the connectivity in the network. As scale-free topology network are characterized by the presence of hub genes, they tend to be very heterogeneous, with some nodes significantly more connected than others [43]. This measure presents several definitions but here it is chosen the one implemented by [43]:

$$Heterogeneity = \frac{\sqrt{variance(k)}}{mean(k)} \tag{3.12}$$

**Clustering coefficients** are useful to study how cohesive the clustering resulting modules, that is the strength of the connections between the neighbours of the nodes [55]. Two are frequently used in the complex network literature [37], but only one is used in the context of microarrays. The purpose is to measure how well connected is a node to its neighbours. It varies from zero to one ($0 \leq C_i \leq 1$), taking the unit when it is at the center of a fully interlinked cluster and zero when its neighbours are not connected at all. The average clustering coefficient may be used to measure if a network presents a modular organization [38].

$$ClusterCoef_i = \frac{\sum_{l \neq i} \sum_{m \neq i} a_{il} a_{lm} a_{mi}}{\{(\sum_{l \neq i} a_{il})^2 - \sum_{l \neq i} (a_{il})^2\}} \tag{3.13}$$

There is also the distinction between the whole-network connectivity and the intramodular connectivity for a given cluster, where both previous connectivities may be used. Intra-modular connectivity measures have found to be more biological meaningful than the whole-network ones [38].

Several other measurements have been proposed. Zhao et al. [27] makes a good review of the measurements developed in the context of WGCNA. Besides, Dong and Horvath [43] demonstrated that many genes or protein clusters are approximately factorizable and show some simple relationships between disparate network concepts.

As the purpose was to study the modular structure of the clusters identified and how well they represented the sub-networks of a gene co-expression network, cluster coefficient was the measurement chosen to rank the clusters. However, for each cluster, connectivity, density, centralization and heterogeneity were also computed.

### 3.6.3 Sample Network

Sample Network method introduced in Section 3.3.3 was defined to build signed sample networks, that is, for each pair of samples $S_i$ and $S_j$, the connection strength, or adjacency, is defined as:

$$a_{ij} = (\frac{cor(S_i, S_j) + 1}{2})^{\beta}, \tag{3.14}$$

where $\beta$ is defined as equal to 2. The two most frequently used network concepts, connectivity and clustering coefficient, are used in their standardized forms. The **standardized connectivity**, $Z.K_i$, of $i$-th node is given by equation 3.6.2. Using the proposed network, the sample network interpretation of the connectivity is that if all sample correlations are $> 0.6$, the connectivity will be given by $k_i \approx \sum_{i \neq j} cor(S_i, S_j)$. That means that highly positively correlated samples tend to be also highly positively correlated with other samples.

The **standardized clustering coefficient** is given by the following equation.

$$Z.C_i = \frac{C_i - mean(C)}{\sqrt{var(C)}} \tag{3.15}$$

With the interpretation that the higher this value, the higher is the average pairwise correlation among its closest neighbours. It gets the value zero, when all of the sample's closest neighbours have a pairwise correlation of $-1$.

To characterize networks, it is use the mean (off-diagonal) adjacency measure (Equation 3.6.2), which is named *intersample adjacency* (ISA). Using the proposed measure of sample adjacency (signed weighted network with $\beta = 2$, the sample network interpretation of the density is $mean([a_ij]) \approx \frac{\sum_i \sum_{i \neq i} cor(S_i, S_j)}{n(n-1)}$, if all sample correlations are above $0.6$.

Empirical results by these authors, demonstrated that, by removing outliers, data becomes normalized and $Z.K$ and $Z.C$ exhibit a progressively linear, inverse relationship. Through the method developed by these authors, the standardized $C(k)$ curve is a scatter plot between $Z.K$ and $Z.C$, previously defined. A new network concept is introduced, $cor(K, C)$, defined by the Spearman correlation between $Z.K$ and $Z.C$ (where other correlation measures might be used). This correlation may be look at as an indicator of heterogeneity and is dependent on the network size. When using this method, one has to study the differences in the $C(k)$ curve (scatter plot of the of the different groups of samples) to reach conclusions regarding the different behaviours of the groups. This standardized curve was demonstrated to be able to assess the overall consistency of sample behaviour within a dataset, identifying distinct groups of samples and important subsets of features. In the present study, we only have blood samples from the patients and thus this method was used to test the differences between the four identified datasets.

### 3.6.4 Module Preservation

Modules consist on cohesive groups of elements in a network that are assumed to be subnetworks inside the main network. Identifying significant modules in a dataset would be more significant if they are preserved across similar datasets and robust to slight changes in its determination procedure. Besides, this preservation could be used to determine if a module is differently preserved between two phenotypes, for example, a pathway of genes may be perturbed in a diseased condition and not in healthy subjects; or to identify non-interesting and possibly outlier modules [62].

**Cross-Tabulation** The standard and intuitive procedure is to perform a cross-tabulation of module membership between the two networks. The most commonly used one is to report the number of clusters that are shared between two modules and use Fisher's exact test to obtain p-values as a significant level measurement to more easily identify overlapping [62].

Alternatively, a hypergeometric p-value may be computed to evaluate the significance of the module overlap between different networks [35, 44]. Carlson et. al [44] proposed a way to determine the upper limit of the p-value, depending on the dataset (refer to [44]).

Finally, Miller et al. [47] proposed a module preservation evaluation that allows module overlapping. Taking two networks, for example, eigengene module membership is computed for every gene in both networks relatively to both set of module eigengenes found. A threshold is then imposed based on these module membership values. In this way, a gene in the first network may be assigned to one or more module in both networks.

**Network Based Statistics** Making use of the specific definition of module as a sub-network, one may use network concepts to characterize these elements. The determining difference between this methodology and the cross-tabulation is that the latter requires module definition in both datasets whilst the former only requires the definition of modules in the reference network. Network based statistics may be split into 1) density based, 2) separability based, and 3) connectivity based preservation statistics [62]. Density based preservation statistics infers if the module nodes identified in the reference network are highly connected in the test network as well. Separability allows to study if the modules in the test network are significantly separated as in reference network. Connectivity based preservation statistics determine if the connectivity pattern between nodes is similar between reference and test network. Langfelder et al. [62] proposed to summarize these statistics into three easily interpreted Z statistics: $Z_{density}$, $Z_{connectivity}$ and $Z_{summary}$. Where the last one is the average of the other two and may be used when the density based and connectivity based preservation statistics are considered equally important for studying module preservation. The density, $Z_{density}$, summarizes the four density preservation statistics related with the mean correlation, mean adjacency, mean squared module eigengene membership and the mean module eigengene membership. The connectivity, $Z_{connectivity}$, considers the information regarding the three connectivity based statistics, the intramodular connectivity, module membership and the correlation value. These statistics are obtained as correlations between the test and reference network corresponding measures (for further details on these measures see [62]).

Empirical p-values were studied and proposed by these authors to threshold module preservation significance. There was consider to exist strong evidence of preservation if $Z_{summary} > 10$, weak to moderated evidence if $2 < Z_{summary} < 10$ and no evidence when $Z_{summary} < 2$. This method was implemented as a R function, *modulePreservation()* by the same authors [62].

# 4

# Results and Discussion

## Contents

In the present section, results from applying the previously described methods are presented. The data used was the one described in Section (3.1), which required some further pre-processing steps. In the first Section (Section 4.1), it was chosen an appropriate method to pre-process the data as well as understanding its structure and applying some standard clustering techniques, such as K-Means and Hierarchical Clustering (section 4.1.2). Interested in studying gene regulatory networks and how they may be altered in diseased and healthy subjects, it was used Weighted Gene Co-expression Analysis (WGCNA) described in Section 3.4.2 and with the results presented in Section 4.2. Several post-processing techniques were applied to these results in order to verify module preservation and module significance in the context of the problem (Section 4.3).

## 4.1 Exploratory Results using Clustering Techniques

### 4.1.1 Pre-Processing

The data had been normalized previously to the present work, as described in Section 3.1. Before applying more complex processing on this dataset, such as WGCNA, a study of the quality and structure of the data was considered necessary. Therefore, standard microarray clustering techniques, such as hierarchical clustering and k-means, were applied.

Four datasets were considered, two with disease samples and two with their matching controls. From this point further, these datasets will be referred as ALS1 and ALS2, respectively for the first and second dataset with diseased samples, and C1 and C2 for their matching controls. When grouping of these subsets occurs, ALS1 and its matching control is referred as dataset 1 and ALS2 and its matching control as dataset 2.

In this chapter, there are usually presented expression profiles and centroids views of all samples considered in that analysis generated by the Genesis software. The image representing the expression profile presents a magenta line symbolizing the mean expression for each sample and a grey cloud representing the variation of expressions for that sample. The centroid view translates the position of the centroid (with standard deviation indicated by vertical lines) of the expression for each sample. On the other hand, when visualizing the expression image for a given cluster, an image with a gradient from green to red, representing the expression value of each gene (row) in each sample (column). The color red representing up-regulated genes, whilst green down-regulated.

**Missing Data** The complete dataset was imported to R and tested for missing values using the R function *goodSamplesGenes()* (part of the WGCNA package). No missing values were identified in the data.

**Data Properties** Hierarchical clustering was chosen to study the relationship between samples as it allows simple and intuitive visualization and interpretation of the structure of the data. As distance metric, it was used the standard distance for this type of clustering, that is the Euclidean distance, but also, in some cases, the Pearson correlation, as it was the preferred distance in the WGCNA approach. In the following steps, regardless of the normalization performed, probe conversion to gene symbols using [53] as well as averaging of probes corresponding to the same gene was applied in the end of the normalization procedure. Therefore, only 6777 unique gene symbols of the original 8,000 probes are used from this point further.

When clustering samples using hierarchical clustering, samples that are closer, relatively to the distance metric chosen, are grouped together first and are connected at a lower height than more distant samples. Therefore, when cutting the dendrogram at a given height, one obtains branches of samples that are closer than the given height of cut. The distance obtained between samples, using the Euclidean distance, is very large (Figure 4.1) and so if one wants to use this or other clustering methods that takes the Euclidean distance as the similarity measure, further normalization of the data is required in order to decrease this inter-sample dissimilarities. On the other hand, if one intends to use the Pearson correlation as the similarity measure, it is observable that the cutting of the dendrogram would be produced several clusters of samples with the same phenotype, that is the intended characteristic of the data.

In the present work, both measures will be used as similarity measures and so further normalization was considered necessary.

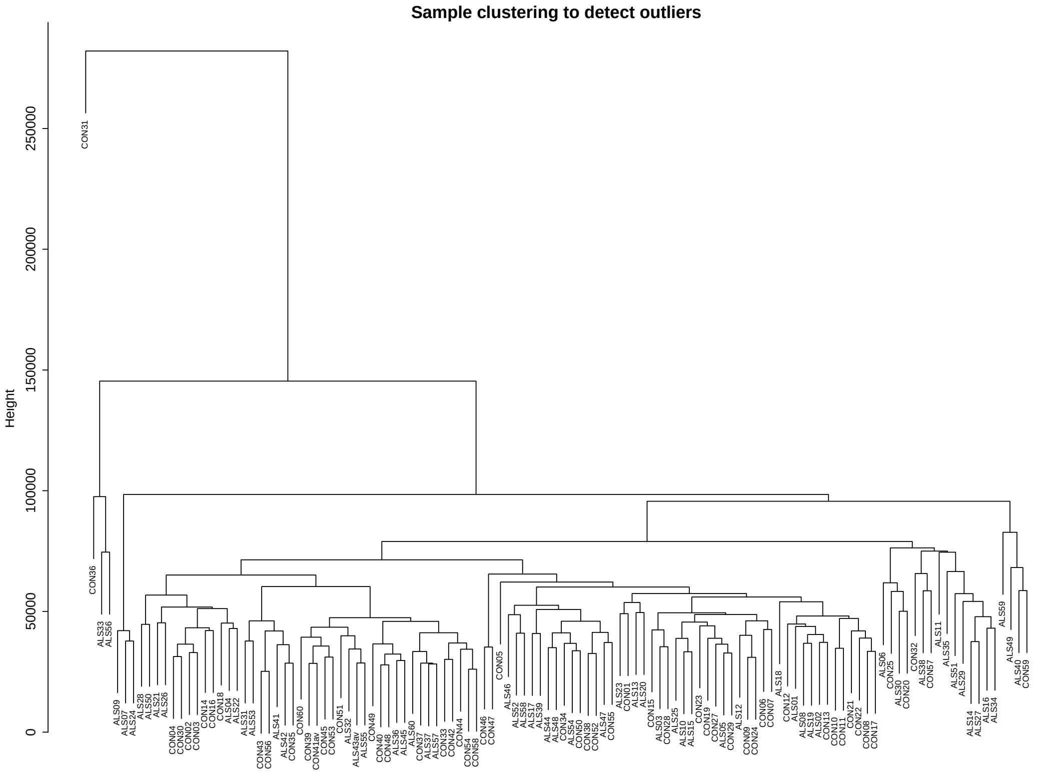


**Figure 4.1:** Average linkage hierarchical clustering based on Euclidean distance applied to samples of the four datasets in their original form. The ALS and control samples are named, respectively, ALS and CON followed by its identification number.

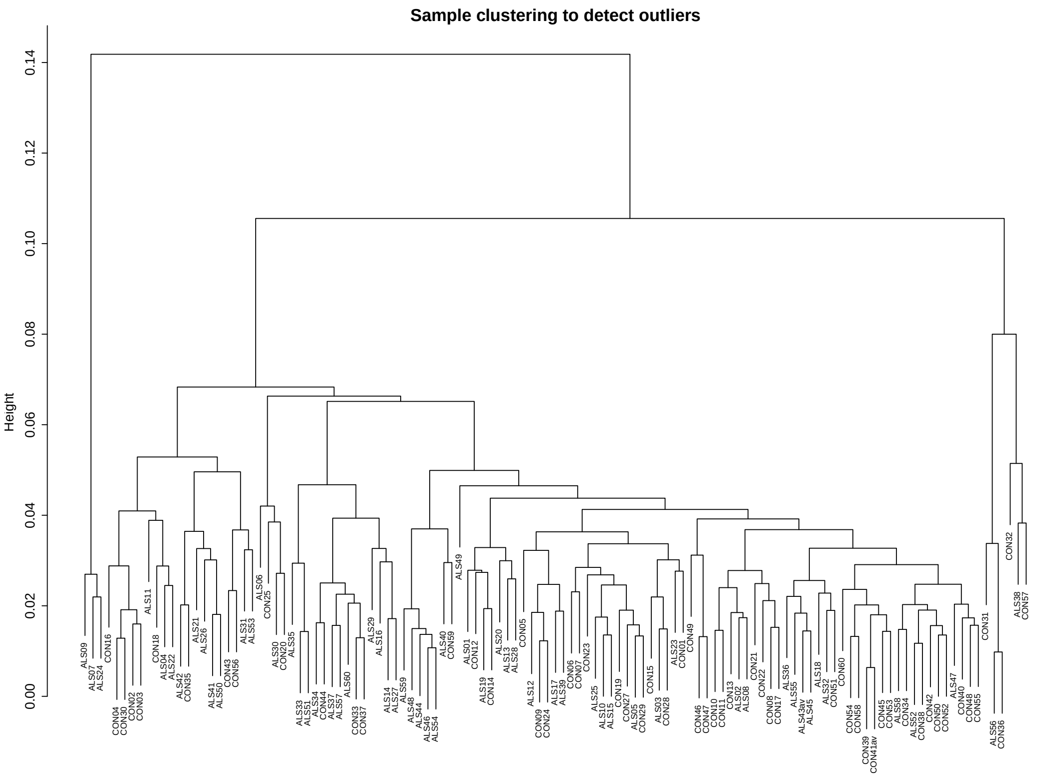


**Figure 4.2:** Average linkage hierarchical clustering with average linkage using Pearson correlation applied to samples of the four datasets in their original form. The ALS and control samples are named, respectively, ALS and CON followed by its identification number.

**Normalization by Sample** Initially, a normalization by sample was performed on the data. When applying hierarchical clustering with Euclidean distance, the distance between samples is smaller, as the samples are grouped at a smaller height in the dendrogram (Figure 4.3). On the other hand, observing the expression values and the centroid position of each sample (Figure 4.4), there are barely no negative expression values and all samples' centroids are approximately horizontal align at zero value. Besides, applying a k-means algorithm ($k = 10$) with Euclidean distance to the entire dataset (Figure A.11), one can verify that all clusters present similar expression distributions, either by their aligned centroids or the expression cloud that varies in an approximately similar way for all samples. Using this method with the Euclidean distance makes it difficult to determine clusters with different expression profiles for each of the datasets, as intended to identify differently expressed genes.

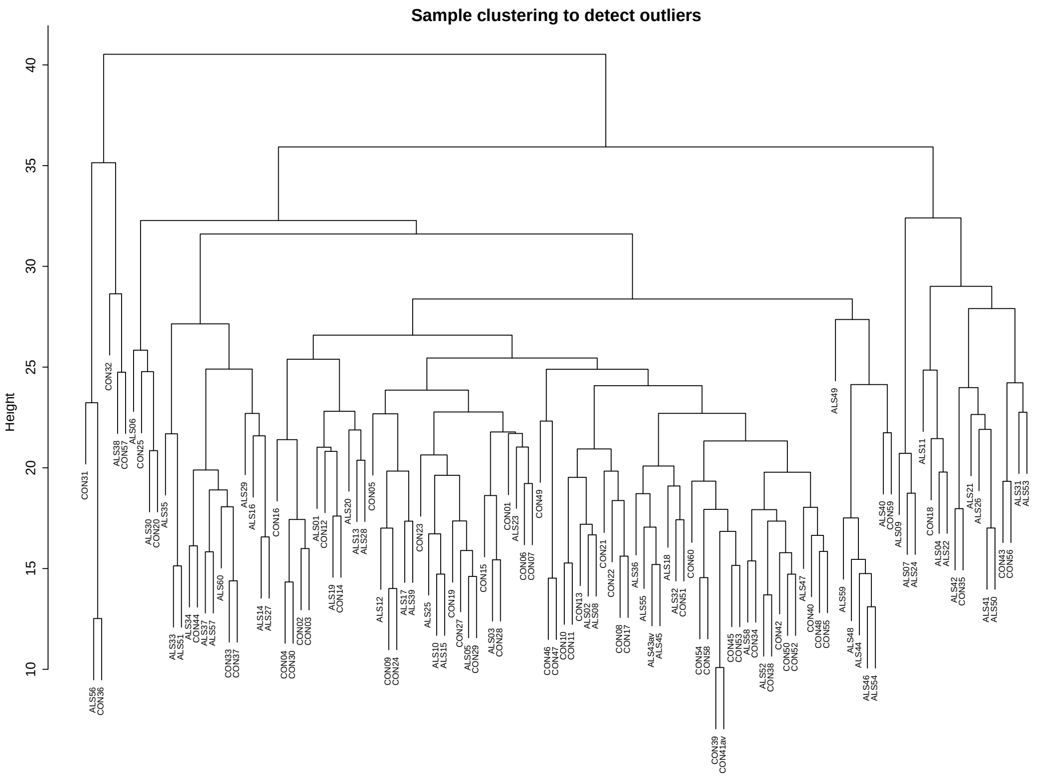


**Figure 4.3:** Average linkage hierarchical clustering based on Euclidean distance between expression values of the four datasets after sample normalization. The ALS and control samples are named, respectively, ALS and CON followed by its identification number.

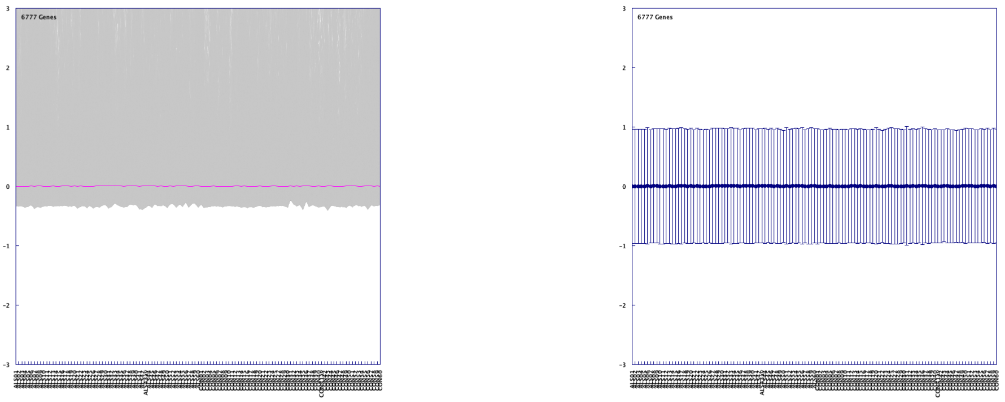


**Figure 4.4:** Expression profiles (left) and centroid view (right) of the complete dataset considering sample normalization. Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width).

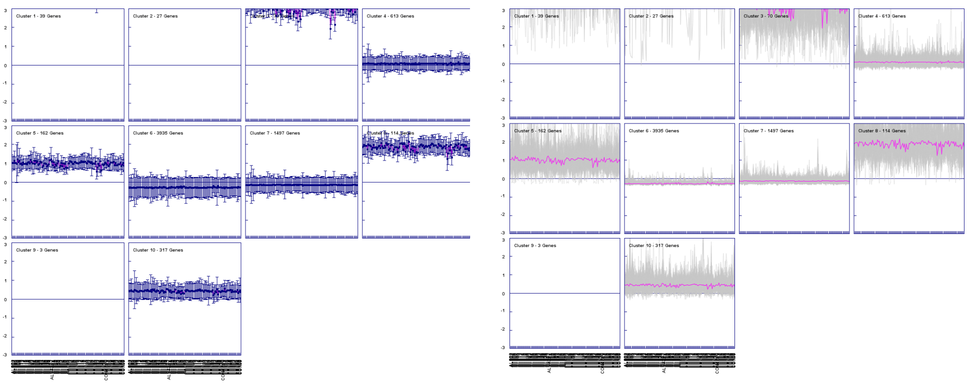


**Figure 4.5:** Resulting clustering from applying the K-Means algorithm with $k = 10$ and using Euclidean distance to cluster genes in the normalized by sample dataset. Centroid view of the clusters (left) and their mean expression (right). Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width).

**Normalization by Sample and Gene** The normalization used in this work was to first normalize the data by samples and then by gene (Figure 4.6). The inter-array variation increased (Figures 4.7) but also the variation between different datasets (Figure 4.8). This normalisation allowed the K-Means algorithm to be used with the Euclidean distance as similarity measure and to obtain discrimination in the expression profiles of different datasets (Figure 4.9).

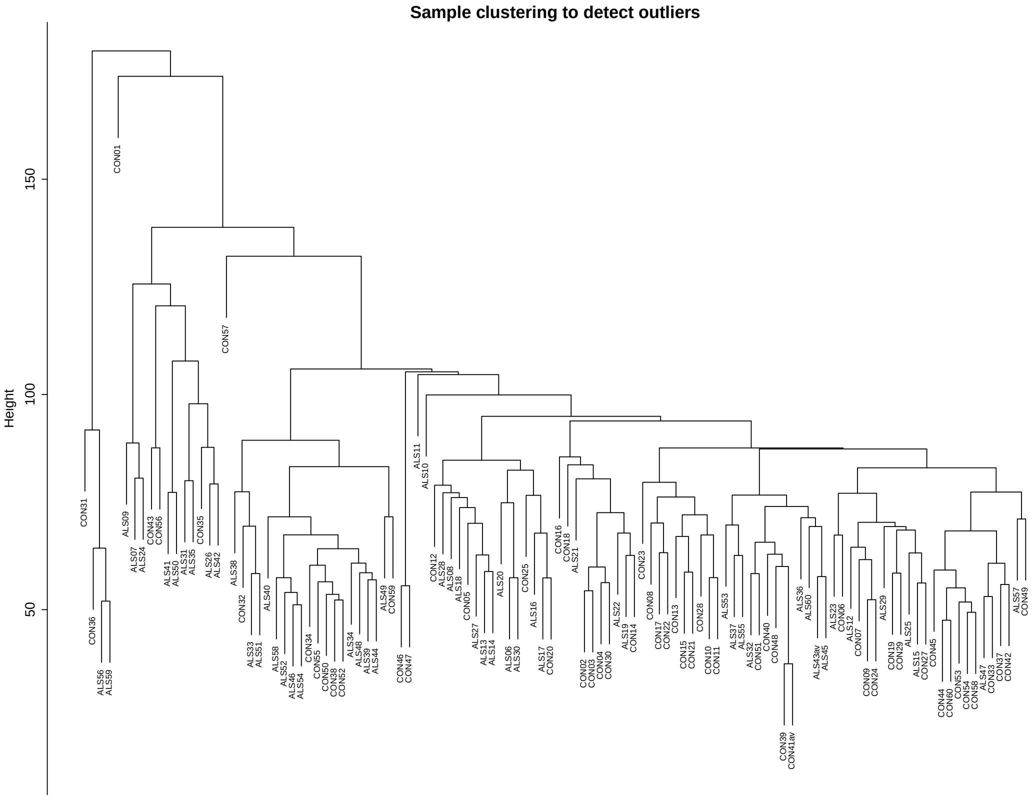


**Figure 4.6:** Average linkage hierarchical clustering based on Euclidean distance between expression values of the four datasets after sample normalization followed by gene normalization. The ALS and control samples are named, respectively, ALS and CON followed by its identification number.

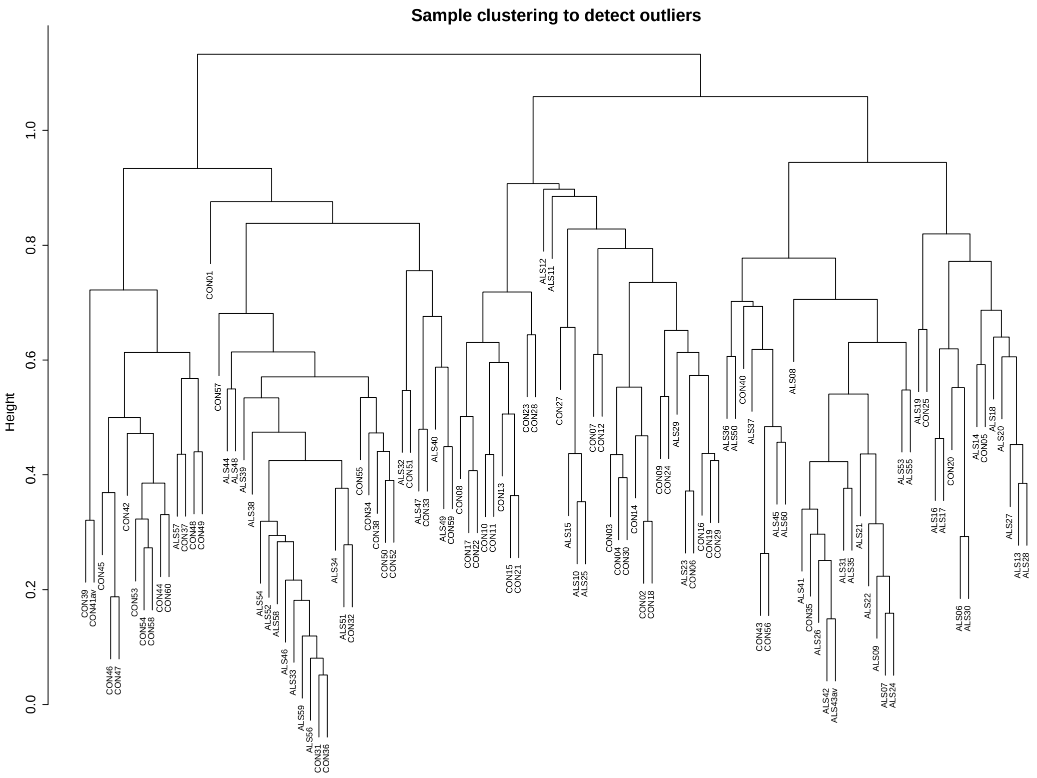


**Figure 4.7:** Average linkage hierarchical clustering based on Pearson distance between expression values of the four datasets after sample normalization followed by gene normalization. The ALS and control samples are named, respectively, ALS and CON followed by its identification number.

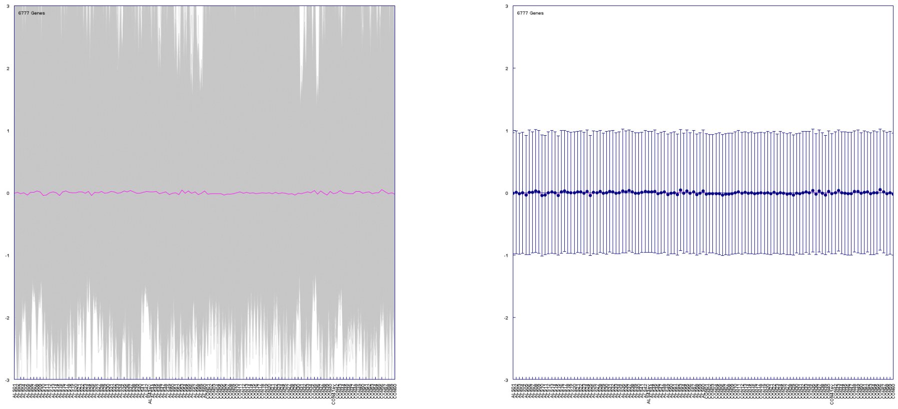


**Figure 4.8:** Expression profiles (left) and centroid view (right) of the complete dataset considering sample and gene normalization. Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width).

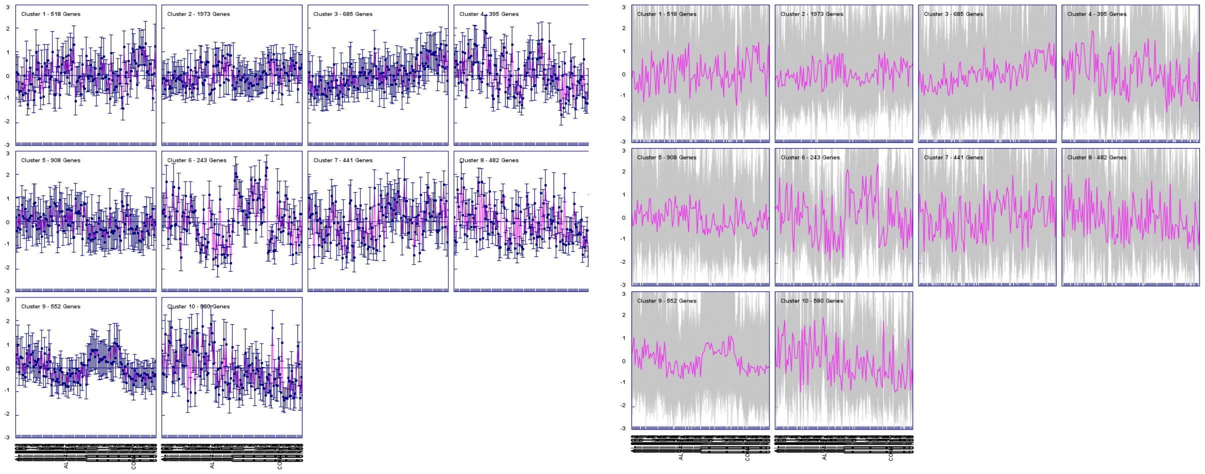


**Figure 4.9:** Resulting clustering from applying the K-Means algorithm with $k = 10$ and using Euclidean distance to cluster genes in the normalized by sample and gene dataset. Centroid view of the clusters (left) and their mean expression (right). Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width).

**Sample Outliers using Sample Network** The standard procedure to detect sample outliers is to perform hierarchical clustering on the samples [55] as performed in the previous sections. Recently, Oldham et al. [55] proposed a method to evaluate the relationships between samples by building a correlation network (described in Section 3.6.3), which was applied to this dataset.

This method was used to study the possible outliers in the dataset, which might be relevant in the gene network building process. Besides, it is expected to be observable a clear separation between samples from different phenotypes. By applying this method to the entire dataset (Figure 4.10) some separation between the four datasets is observable in the dendrogram. The blue dataset corresponding to C1 is the most isolated from the rest. Using this figure to identify sample outliers indicates some samples that might be considered outliers in the ALS2 and C2 datasets but no clear outlier in the first dataset.

Then, each dataset was considered in separate and the same procedure was applied (Figures 4.11 and 4.12). First, considering outliers, 3 ALS samples (ALS07, ALS09, ALS24) are indicated as possible outliers in the first dataset whilst in the second one sample of ALS (ALS56) and two of the controls (CON31 and CON36) are identified.

Sample network construction also offers the possibility to separate samples corresponding to different phenotypes based on their connectivity differences, that is, by analysing the heterogeneity indicator given by the relationship between the standardized sample connectivities $(Z.K)$ and the standardized sample clustering coefficients $(Z.C)$. Considering this heterogeneity indicator, the first dataset (Figure 4.11) presents a more relevant distinction between phenotypes as the difference between curve's slopes of the two phenotypes is more relevant than the same difference in the second dataset (Figure 4.12).

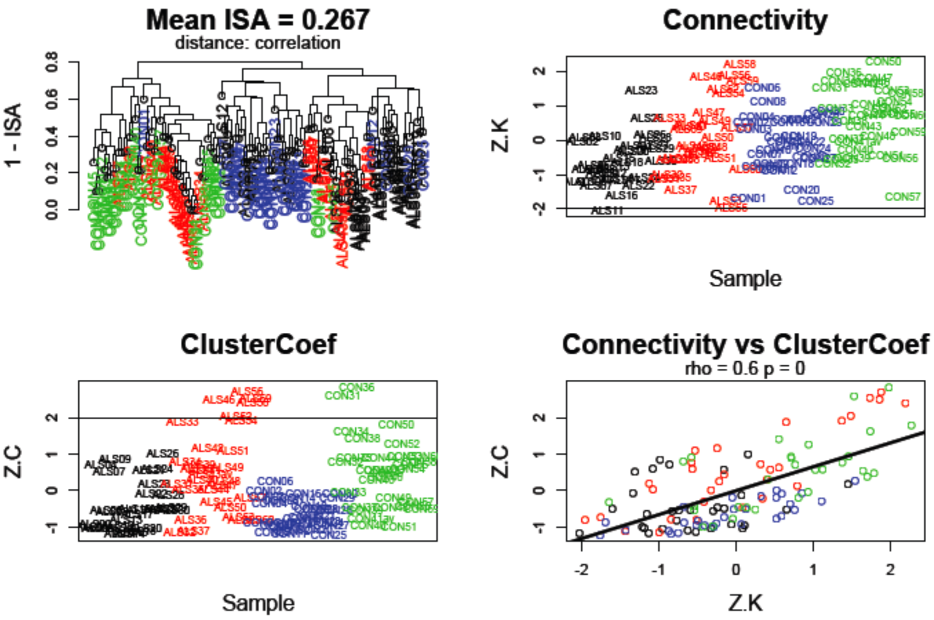


**Figure 4.10:** Sample Network study performed on the four datasets: ALS dataset 1 (black) and 2 (red); control datasets 3 (blue) and 4 (green). (A - top left) Dendrogram resulting from average linkage hierarchical clustering using 1 - ISA (intersample adjacency) for a subset of samples. (B - top right) Standardized sample connectivities ($Z.K$) for all samples. (C-bottom left) Standardized sample clustering coefficients ($Z.C$) for all samples (D - bottom right) Comparison of standardized sample connectivities ($Z.C$) against standardized sample clustering coefficients ($Z.K$).

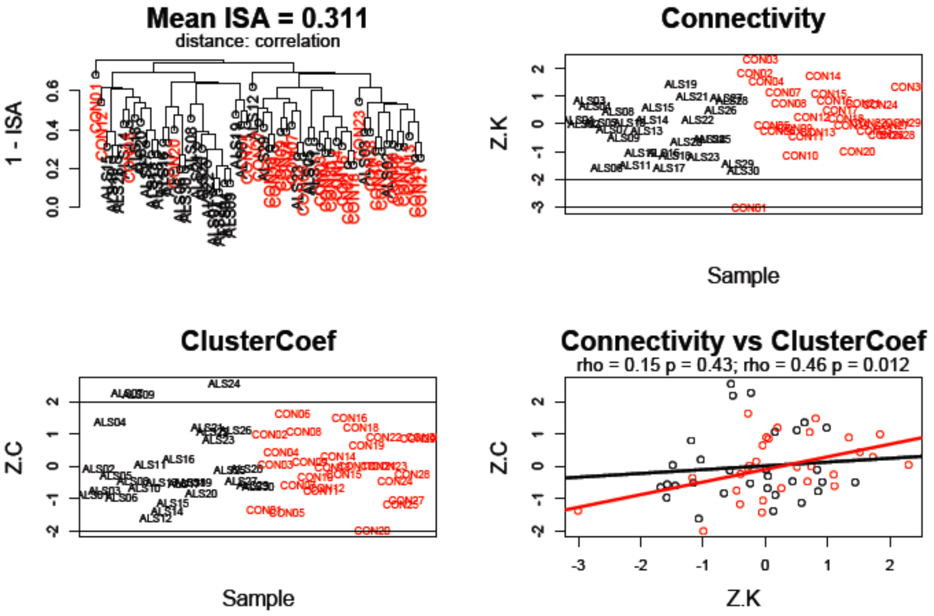


**Figure 4.11:** Sample Network study performed on the four datasets: ALS dataset 1 (black) and control (red). (A - top left) Dendrogram resulting from average linkage hierarchical clustering using 1 - ISA (intersample adjacency) for a subset of samples. (B - top right) Standardized sample connectivities ($Z.K$) for all samples. (C-bottom left) Standardized sample clustering coefficients ($Z.C$) for all samples (D - bottom right) Comparison of standardized sample connectivities ($Z.C$) against standardized sample clustering coefficients ($Z.K$).

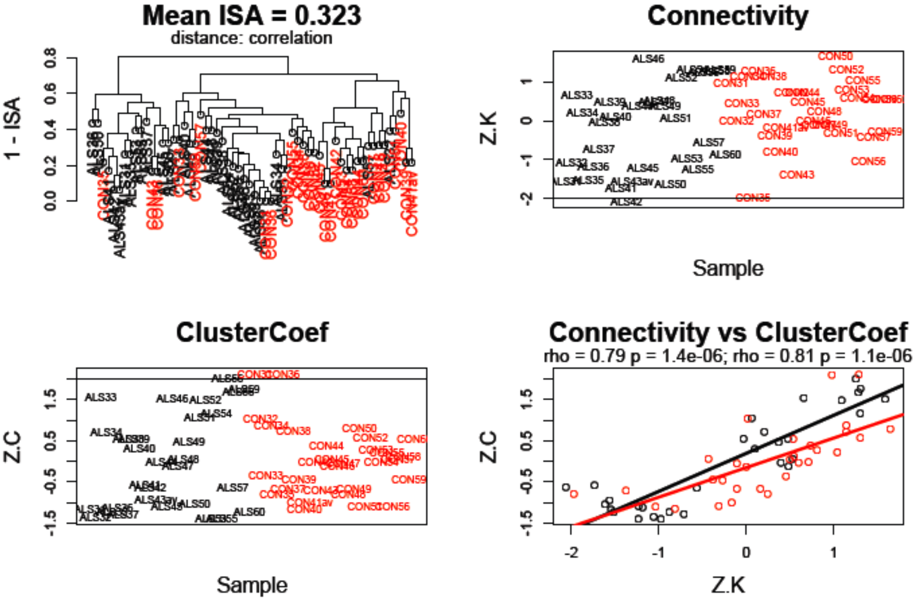


**Figure 4.12:** Sample Network study performed on the four datasets: ALS dataset 2 (black) and control (red). (A - top left) Dendrogram resulting from average linkage hierarchical clustering using 1 - ISA (intersample adjacency) for a subset of samples. (B - top right) Standardized sample connectivities ($Z.K$) for all samples. (C-bottom left) Standardized sample clustering coefficients ($Z.C$) for all samples (D - bottom right) Comparison of standardized sample connectivities ($Z.C$) against standardized sample clustering coefficients ($Z.K$).

**Batch Effects** As some of the clusters identified using the K-Means algorithm (Figure 4.9) appear to be dependent on the dataset considered, some batch effects may be present in the data. As no information regarding the sample batches was provided by the authors and all samples from the four datasets were presented in the same file, batch effects correction was not taken into consideration.

Later, batch effects were tested using a R function named *ComBat.R*. K-Means was re-applied to this corrected dataset (Figure 4.16) and it is no longer clear the dependence of samples regarding its dataset. Similar plots to observe the resulting structure of the data were then obtained and may be observed in Figures 4.13, 4.14 and 4.15. Further studies should be performed to confirm this intuition. To be noted that this correction was not applied to the present dataset but the separation in the processing of the two datasets was considered relevant from this point further.

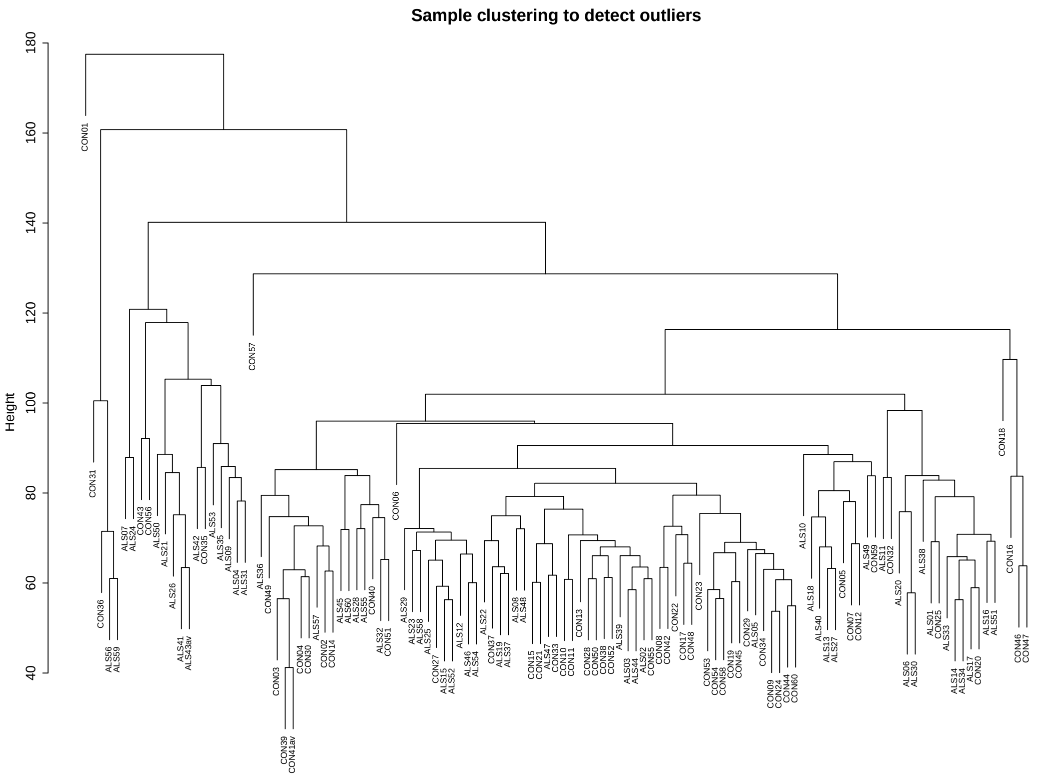


**Figure 4.13:** Average linkage Hierarchical clustering based on Euclidean distance between expression values of the four datasets after sample normalization followed by gene normalization and batch effects correction. The ALS and control samples are named, respectively, ALS and CON followed by its identification number.

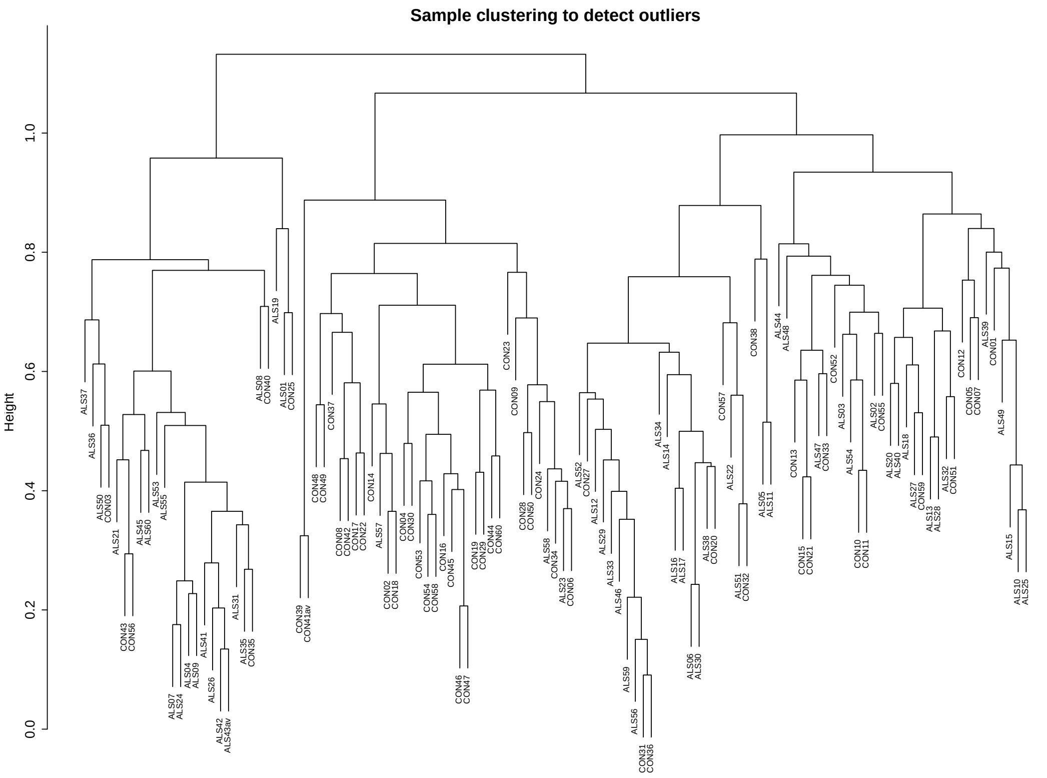


**Figure 4.14:** Average linkage Hierarchical clustering based on Pearson distance between expression values of the four datasets after sample normalization followed by gene normalization and batch effects correction. The ALS and control samples are named, respectively, ALS and CON followed by its identification number.

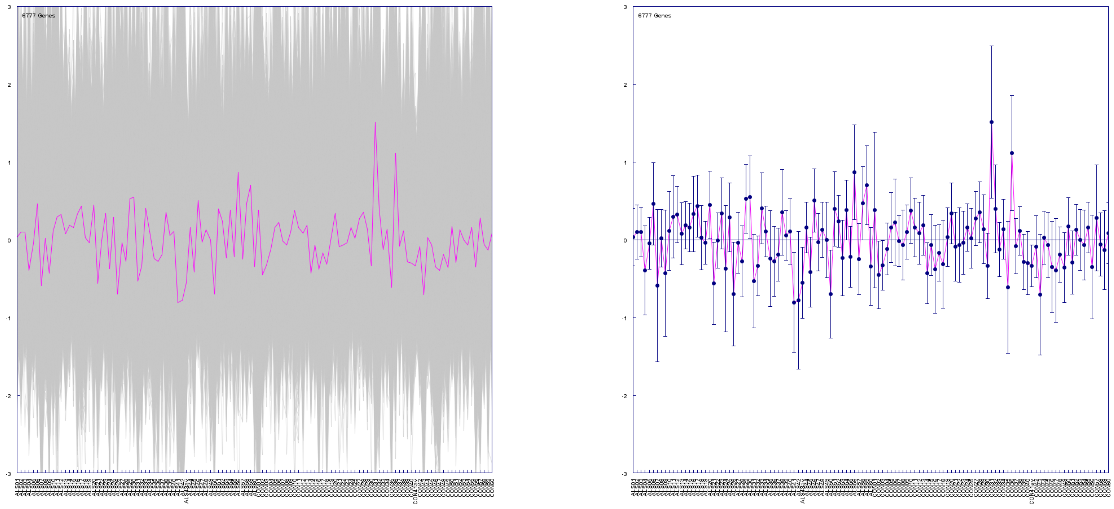


**Figure 4.15:** Expression profiles (left) and centroid view (right) of the complete dataset normalized by sample and gene and with batch effects correction. Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width)

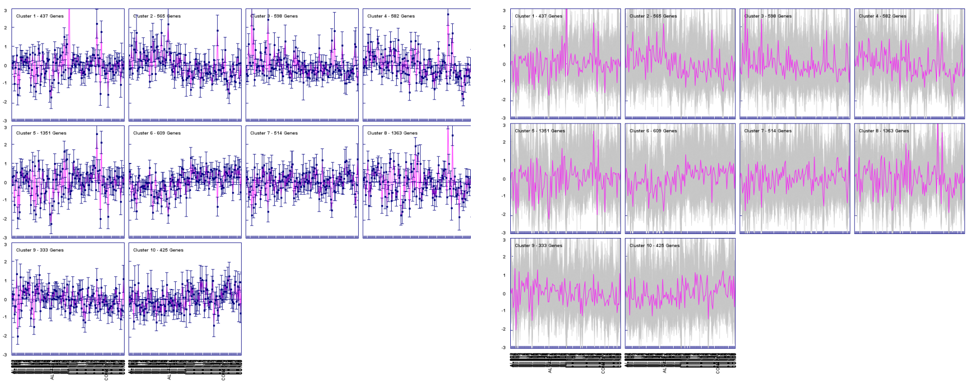


**Figure 4.16:** Resulting clustering from applying the K-Means algorithm with $k = 10$ and using Euclidean distance to cluster genes in the entire dataset after sample and gene normalization and with batch effects correction. Centroid view of the clusters (left) and their mean expression (right). Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width).

### 4.1.2 Clustering Genes

Two standard clustering techniques were applied to cluster genes: hierarchical clustering and k-means. Both were used with the standard similarity measure, the Euclidean distance. The purpose of this clustering analysis was to identify clusters of genes where the expression of ALS samples is significantly different from that in controls (for instance, up regulation in ALS and down regulation in controls) and thus a combination of datasets was used. In these steps, all samples were first considered together, then similar phenotypes were grouped together and finally the two datasets were considered independently, that is, ALS1 with C1 and ALS2 with C2.

**Hierarchical Clustering** As already suspected the batch effects present in the data made it difficult to determine the intended clusters when clustering all samples or combining only one type of phenotype. In fact, when using all samples in the dataset (Figure A.1) there is no clear cluster that separates diseased from control but some clusters do appear to differentiate between datasets, such as Figure 4.17 (and Figure 4.18), which may again indicate the presence of batch effects.

An attempt to identify clusters that were characteristic to a single phenotype was made, by clustering ALS datasets and applying the algorithm. Probably due to the already described differences, no relevant clusters were identified.

However, in dataset 1, two clusters were considered to present differentially expressed genes between ALS and controls (Figures 4.19 and 4.20), as they clearly present a different color (representing the expression values) between these two classes. In the second dataset (ALS2 with C2), no relevant clusters were identified, with some clusters suggesting the presence of sample outliers (Figure A.2).

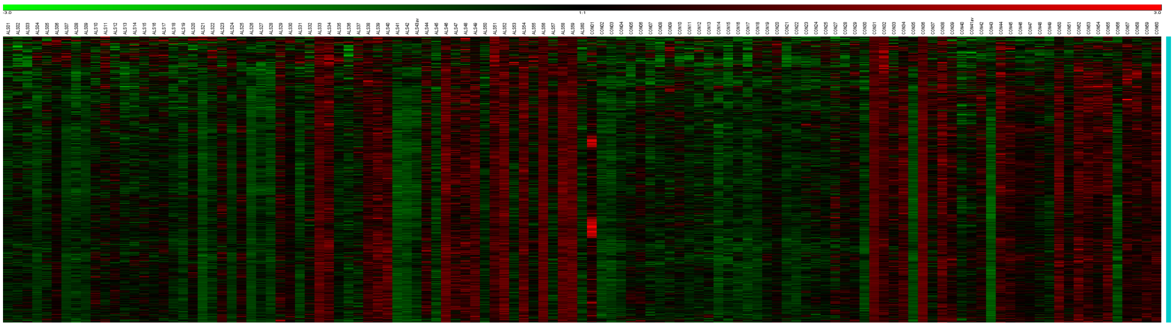


**Figure 4.17:** Cluster identified with average linkage hierarchical clustering based on Euclidean distance applied to cluster genes in all samples. Some evidences of dataset dependence is observable. Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width).

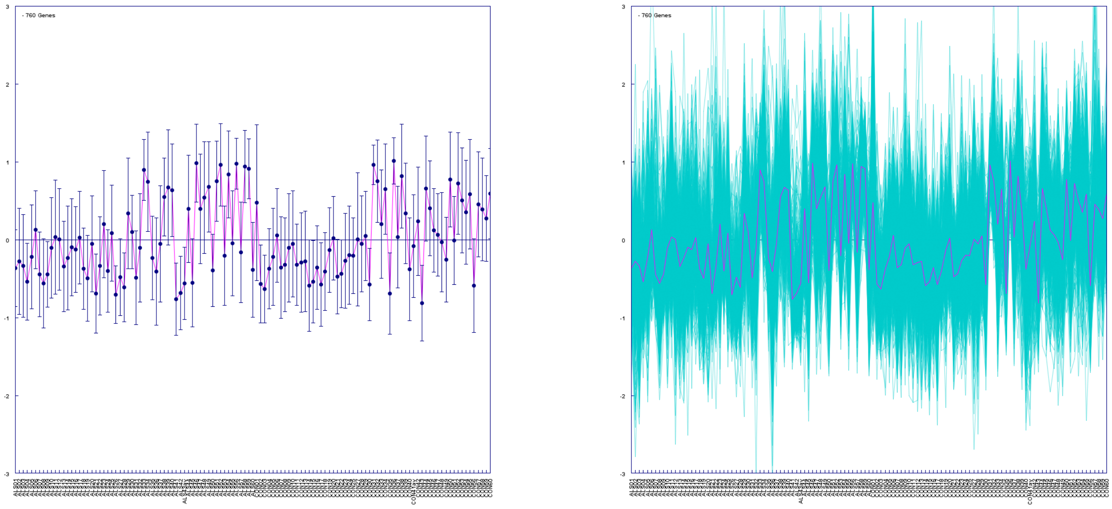


**Figure 4.18:** Centroid view of one cluster (left) and its mean expression pattern (right) that resulted from the average linkage hierarchical clustering using Euclidean distance applied clusters genes in all samples. Some evidences of dataset dependence is observable. Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width).

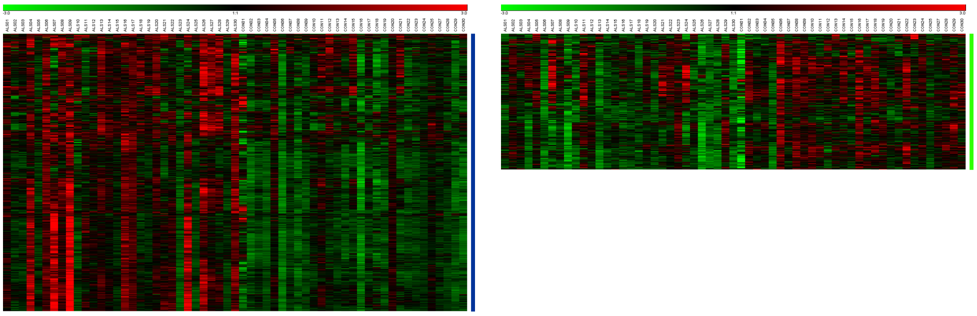


**Figure 4.19:** The two clusters considered containing differentially expressed genes identified with average linkage hierarchical clustering based on Euclidean distance in the combination of ALS1 and C1. The first (left) and second (right) cluster are color label as blue and turquoise, respectively. Dataset order from left to right: ALS1, C1 (each one corresponds to 1/2 of the image's width).

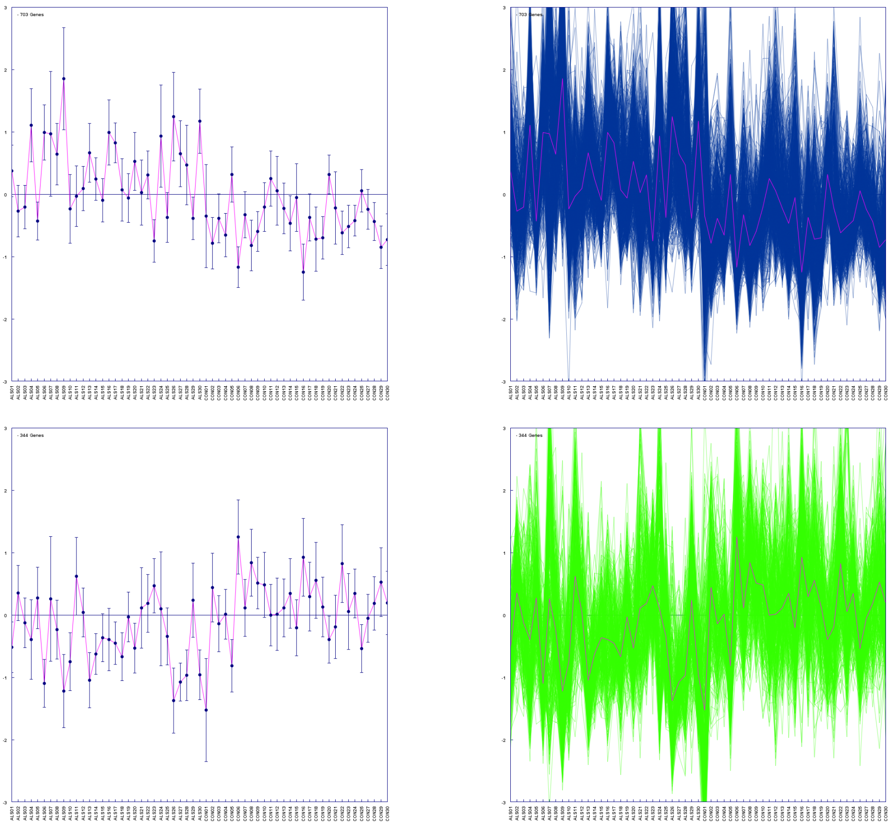


**Figure 4.20:** Centroid view (left) and mean expression pattern (right) of the two clusters considering containing differentially expressed genes identified with average linkage hierarchical clustering based on Euclidean distance in the combination of ALS1 and C1. The blue cluster is represented on the top and the turquoise on the bottom images. Dataset order from left to right: ALS1, C1 (each one corresponds to 1/2 of the image's width).

The use of the absolute Pearson correlation as a similarity measure in hierarchical clustering method was studied for the different groups of datasets. Selection of interesting clusters in this type of clustering is extremely difficult due to the property of, when using absolute Pearson correlation, to obtain clusters of positive or anti-correlated genes. Therefore, no clusters were selected using this approach.

**K-Means** K-Means was applied to the five combinations of datasets previously described (ALL, ALS1 with ALS2, C1 with C2, ALS1 with C1, ALS2 with C2), making use of Euclidean distance and using 4 values of $k$: 10, 20, 50 and 100. The lowest value of $k$ was used to understand if enough variation was presented in the dataset (Figure 4.9). On the other hand, using the highest values of $k$, that is $k = 100$, resulted in too much splitting of the data and was disregarded. By clustering the entire dataset, one may understand how the data influence each other. When using all samples (Figure A.3) and, as expected by previous results, the dataset dependence is observable in some of the clusters when using this method.

As when hierarchical clustering was used, the interesting clusters in the current problem are the ones that allow to identify clusters where the expression between ALS samples is significantly different from that in controls. Therefore, datasets combining ALS and control samples by dataset were used to obtained such clusters. In the first dataset, 3 interesting clusters were identified when using $k = 20$

(Figure A.8) and 6 when using $k = 50$ (Figure A.10). The expression profile of these clusters are depicted in Figures 4.21 and 4.22 in the first case, and Figures 4.23, 4.24 and Figure 4.25 in the second.

Clustering samples from the same phenotype revealed some clusters containing genes with similar expression across all samples, but also many differentiating between datasets (Figures A.5 and A.6). Therefore, clusters resulting from grouping similar phenotypes were disregarded in the present work.

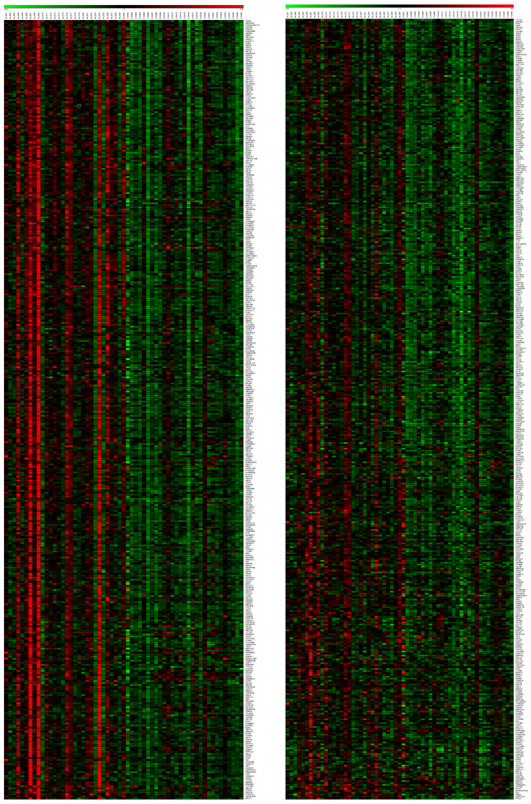


**Figure 4.21:** Expression profile of 1st and 3rd cluster using $k = 20$ with Euclidean distance when clustering genes from samples from dataset 1. These clusters were color named turquoise and blue, respectively. Dataset order from left to right: ALS1, C1 (each one corresponds to 1/2 of the image's width).

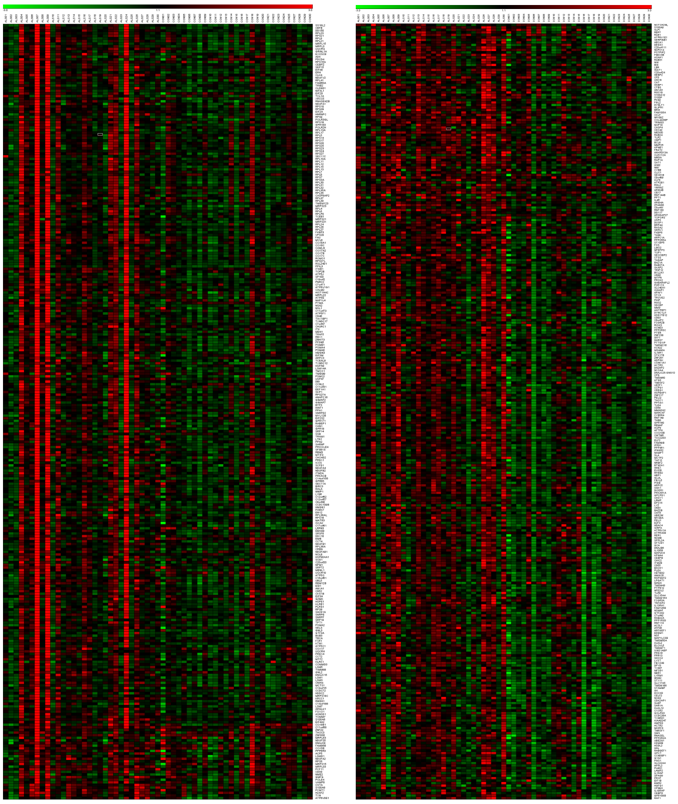


**Figure 4.22:** Expression profile of 18th and 20th cluster using $k = 20$ with Euclidean distance when clustering genes from samples from dataset 1. These clusters were color named brown and yellow, respectively. Dataset order from left to right: ALS1, C1 (each one corresponds to 1/2 of the image's width).

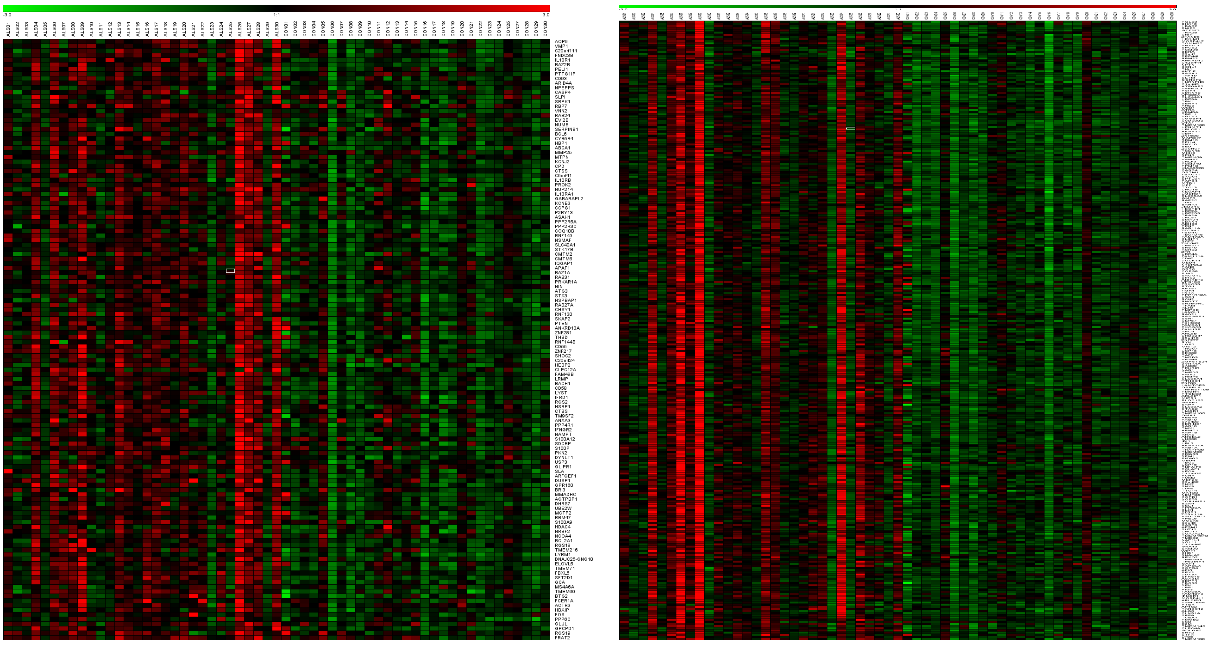


**Figure 4.23:** Expression profile of 28th and 37th cluster using $k = 50$ with Euclidean distance when clustering genes from samples from dataset 1. These clusters were color named turquoise and blue, respectively. Dataset order from left to right: ALS1, C1 (each one corresponds to 1/2 of the image's width).

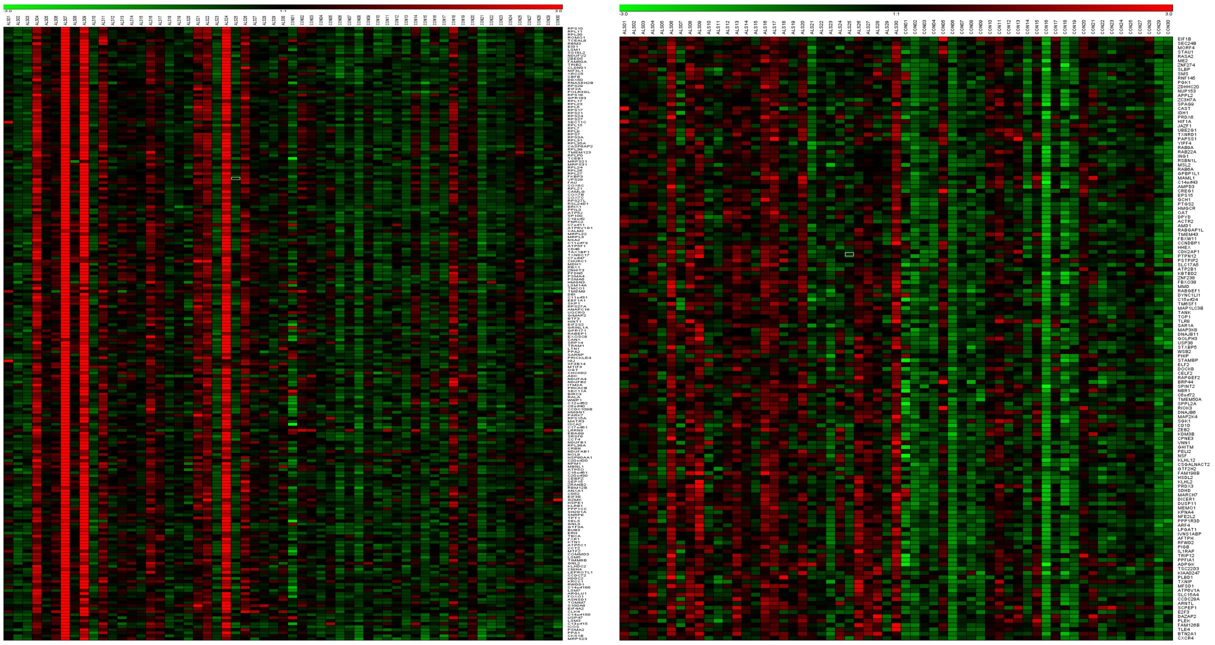


**Figure 4.24:** Expression profile of 40th and 41th cluster using $k = 50$ with Euclidean distance when clustering genes from samples from dataset 1. These clusters were color named brown and yellow, respectively. Dataset order from left to right: ALS1, C1 (each one corresponds to 1/2 of the image's width).

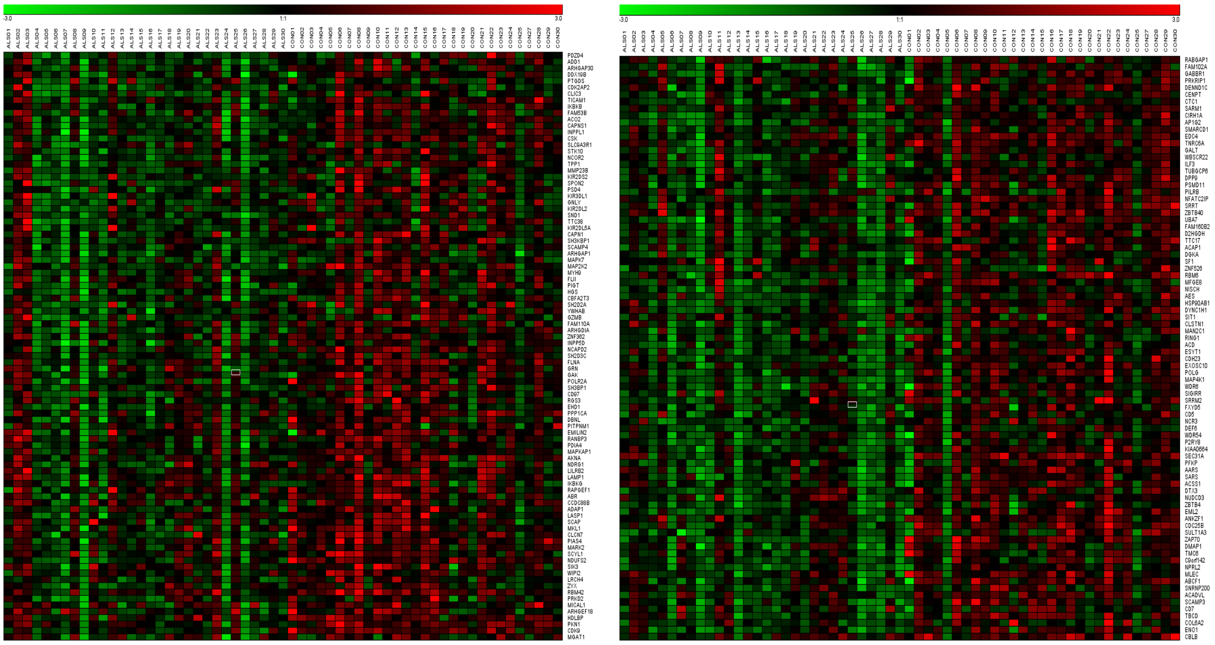


**Figure 4.25:** Expression profile of 42th and 46th cluster using $k = 50$ with Euclidean distance when clustering genes from samples from dataset 1. These clusters were color named green and red, respectively. Dataset order from left to right: ALS1, C1 (each one corresponds to 1/2 of the image's width).

## 4.2 Weighted Gene Co-expression Network Analysis

The Weighted Gene Co-expression Network Analysis (WGCNA) was applied in parallel to each dataset, following the same pipeline.

**Type of network** As described in the pipeline of the algorithm (Section 3.4.2), the first decision is regarding the adjacency function applied to the expression matrix. The power function was the selected one as several papers have indicated it to transform the network into an approximately scale free one, as desirable.

**Soft Power** The soft thresholding power, $\beta$, was 6 as it provides a good compromise between fitting the model to an approximately scale-free and not decreasing too much the mean connectivity (Figure 4.26). In the Figure 4.26 it is represented the model fitting of dataset 1 to a scale free topology, but very similar results were observed for the other 3 datasets and so the same parameter was chosen for all of them (Appendix section A.2.1). Besides, it was also the value used by the authors [5]. For any choice of $\beta > 1$, large values of the absolute correlation would be emphasized, whilst smaller will become even smaller.

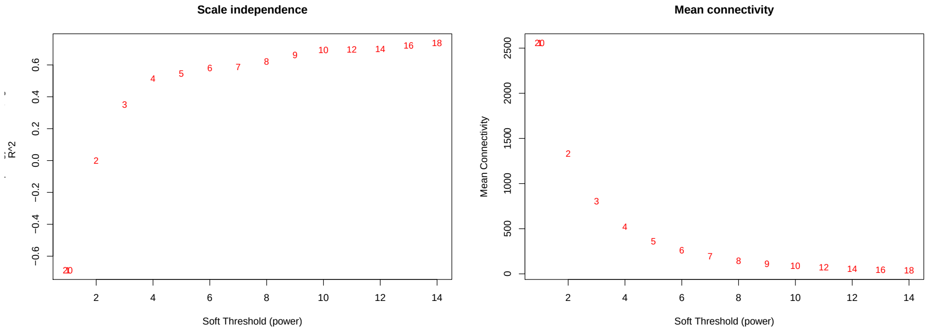


**Figure 4.26:** Fitting of the network to the scale-free topology accordingly to the beta parameter used in the soft-thresholding procedure for dataset ALS1.

Using $\beta = 1$ will correspond to using the absolute Pearson correlation as similarity measure and it is a common practice in microarray's clustering [60]. Therefore the resulting network was computed (Figure 4.27). On the other hand, it was also interesting to use smaller values of beta in order to directly study the impact of the topological overlap similarity measure over the adjacency matrix (Figure 4.27). It was also tested using a small value of beta $(= 2)$ that is depicted in Figure 4.28.

In the literature, it was demonstrated by Zhang et al. [38] that the modules identified by the WGCNA method are highly robust to the choice of the parameter beta. Thus these parameters were only varied to visually observe the differences. Using the topological overlap measure directly on the adjacency matrix (Figure 4.27) will tend to produce a single branch with several small nested clusters. Even using the dynamic hybrid branch cutting algorithm, it is difficult to determine large clusters. When applying the soft-thresholding procedure, using a low soft-thresholding power such as $\beta = 2$ will result in more and more defined nested clusters, that will be more easily identified the cutting tree algorithm.

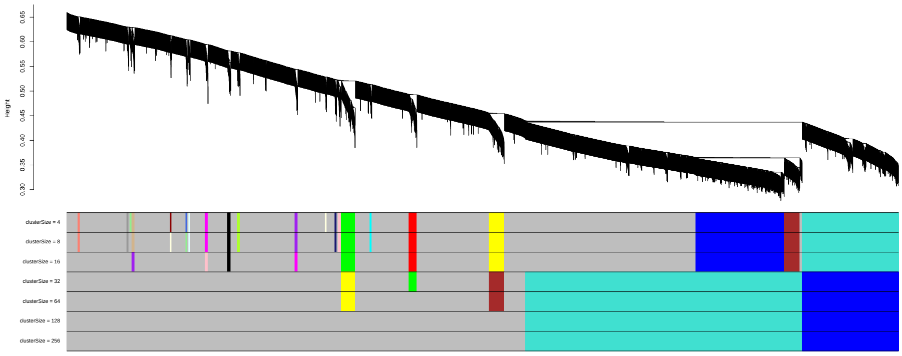


**Figure 4.27:** Average linkage hierarchical clustering dendrogram applied to dataset ALS1, where no soft thresholding was used to threshold the adjacency matrix. Module assignment given by applying dynamic hybrid branch cutting ($deepSplit = 1$ and varying the $minimumclustersize = 4, 8, 16, 32, 64, 128, 256$) is depicted by row color immediately below the dendrogram, with grey representing unassigned genes.

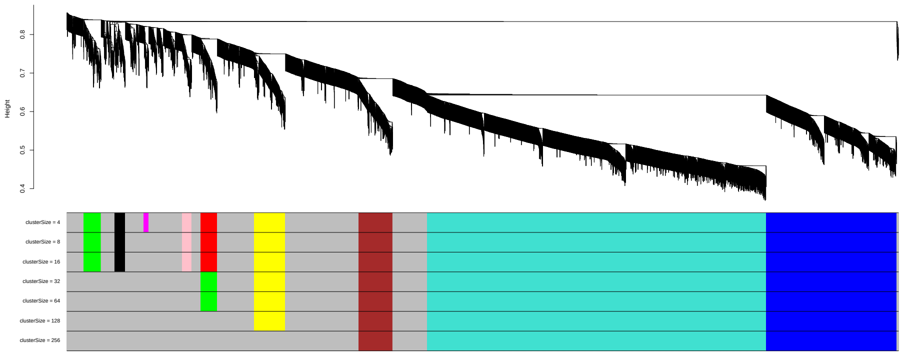


**Figure 4.28:** Average linkage hierarchical clustering dendrogram applied to dataset ALS1, where soft thresholding of 2 was used to threshold the adjacency matrix. Module assignment given by applying dynamic hybrid branch cutting ($deepSplit = 1$ and varying the $minimumclustersize = 4, 8, 16, 32, 64, 128, 256$) is depicted by row color immediately below the dendrogram, with grey representing unassigned genes.

**Color Code** Through out this pipeline, modules and clusters are identified through its color name or simply the color label and thus a matching correspondence of the names and colors is made in Figure 4.29. Whenever it is required instant conversion of colors, the same label is represented.

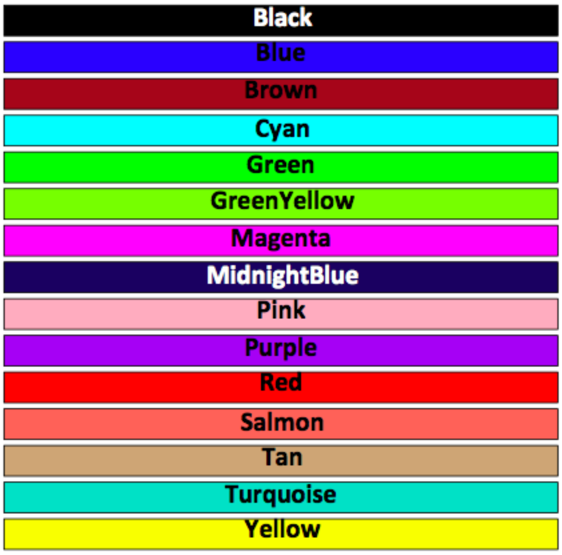


**Figure 4.29:** Color code used in this work.

**How to choose the modules?** To obtain modules in an hierarchical clustering procedure, one has to cut the dendrogram. To do so, branch cutting hybrid method (described in 3.4.1) was used and two parameters of this method were varied: *deepSplit* and *minimum cluster size*. By visual inspection it is clear that the variation of these parameters result in very different modules (see Figure A.2.2). Besides, when using WGCNA, we are interested in identifying modules that would correspond to sub-networks of the gene regulatory network and so, the modules resulting from these cuts should present good network properties. Therefore, it is reasonable to use network properties to guide the choice of the group of modules that, in general, present better properties. Some concepts of the network literature (size, centralization, heterogeneity, mean cluster coefficient and connectivity - Section 3.6.2) were applied to the entire dataset and for each module in each group of modules obtained (the average values of these measures are presented in Table 4.1).

In the original paper that used the present dataset, the choice was to obtain few and large clusters, that were consequently involved in a wide variety of gene functions [5]. Therefore, the reasoning of the present work was to choose the cut that produces, in general, more cohesive modules, which results in an higher number of modules and smaller ones. The measure chosen to determine cohesiveness of modules was the clustering coefficient (Figure A.19 and A.20) although heterogeneity, density and centralization were also taken into consideration where several options were possible.

It was more troublesome to choose the cut for the dataset 1, both ALS1 and C1, as the structure of the dendrogram allowed for more nested clusters. First, it is interesting to identify nested clusters in large branches without excessive partitioning of the branch because in dataset 2 less and larger modules were identified. Therefore, the cuts that apparently followed more this reasoning were $deepSplit = 0, 1$ for small values of *mininum cluster size* (4, 8, 16) for ALS1. Considering that the mean clustering coefficient for the entire network was of 0.448, it is interesting to identify modules with higher mean clustering coefficient (Figures A.19 and A.20). For higher values of *deepSplit*, when the *mininum cluster size* is small, over partitioning occurs. On the other hand when *mininum cluster* *size* is higher, the overall clustering coefficients are much worse. A good compromise between these two limitations is obtained for the chosen cut, that is the $deepSplit = 1$ and $minimumclustersize = 4$ (the same for $8$) represented in Figure 4.30, as it was the one that identified the potential interesting nested cluster in the turquoise module.

In the case of C1, the mean clustering coefficient of the modules encountered in this network are lower than in the previous one. However, the network clustering coefficient is also significantly lower $(= 0.316)$. Therefore, visual inspection of the dendrogram combined with the analysis of the histograms (Figures A.23 and A.24) lead to the selection of $deepSplit = 0$ and *minimum cluster size* equals to 4 (Figure 4.32) for dataset C1.

The second ALS dataset (ALS2) presented a mean clustering coefficient $(0.544)$ slightly higher than the ALS1, as well as the other network concepts. Visual inspection of the dendrogram (Figure A.16) and using the same reasoning, the apparently better cutting is $deepSplit = 2$ and *minimum cluster size* equals to $8$. Analysing the clustering coefficient histograms (Figures A.21 and A.22) and even though only few clusters are above the general clustering coefficient of the network, the selected one is also one (Figure 4.31) of the best possible cuts.

For the last dataset (C2), it was clear that the $deepSplit = 3$ produces the best set of modules with regard to the clustering coefficient (with some exceptions) (see Figures A.25 and A.26). From these the *minimum cluster size* equals to 32 was the selected one (Figure 4.33).

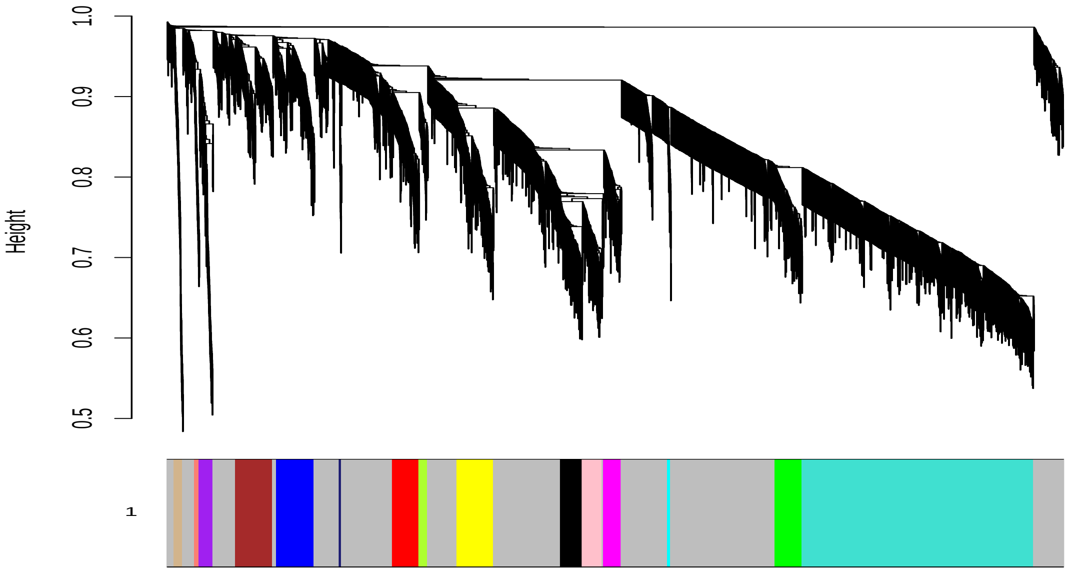


**Figure 4.30:** Average linkage hierarchical clustering dendrogram applied to dataset ALS1. Module assignment given by applying dynamic hybrid branch cutting with $deepSplit = 1$ and $minimumclustersize = 4$ is depicted by row color immediately below the dendrogram, with grey representing unassigned genes.

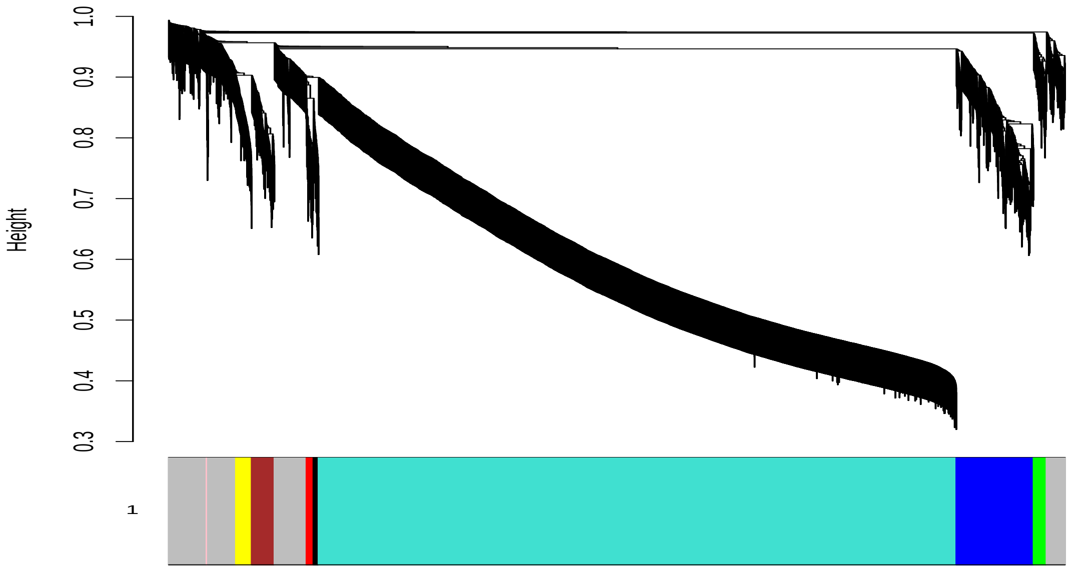


**Figure 4.31:** Average linkage hierarchical clustering dendrogram applied to dataset ALS2. Module assignment given by applying dynamic hybrid branch cutting with $deepSplit = 2$ and $minimumclustersize = 8$ is depicted by row color immediately below the dendrogram, with grey representing unassigned genes.

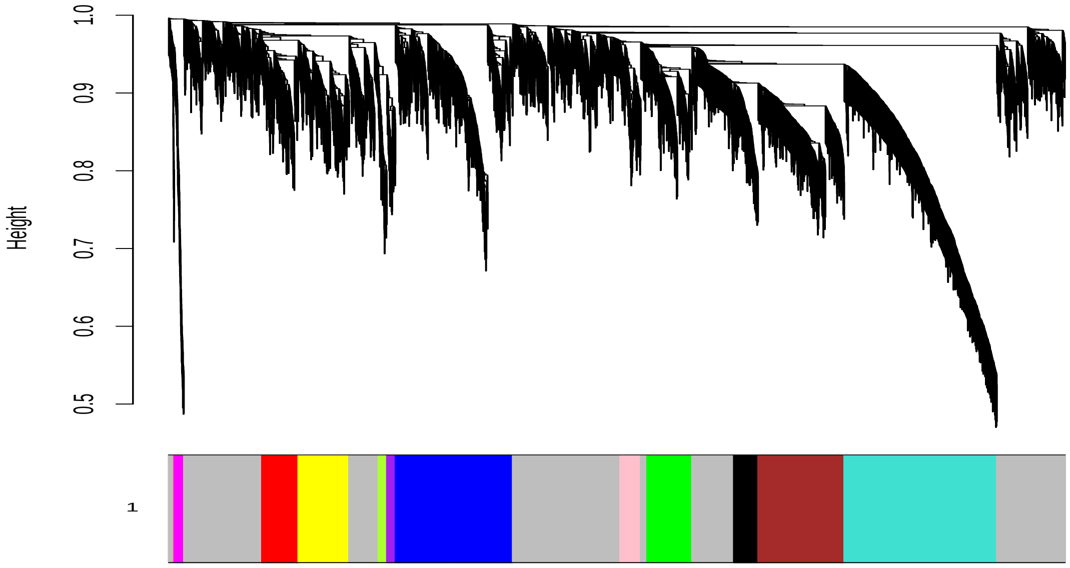


**Figure 4.32:** Average linkage hierarchical clustering dendrogram applied to dataset C1. Module assignment given by applying dynamic hybrid branch cutting with $deepSplit = 0$ and $minimumclustersize = 4$ is depicted by row color immediately below the dendrogram, with grey representing unassigned genes.

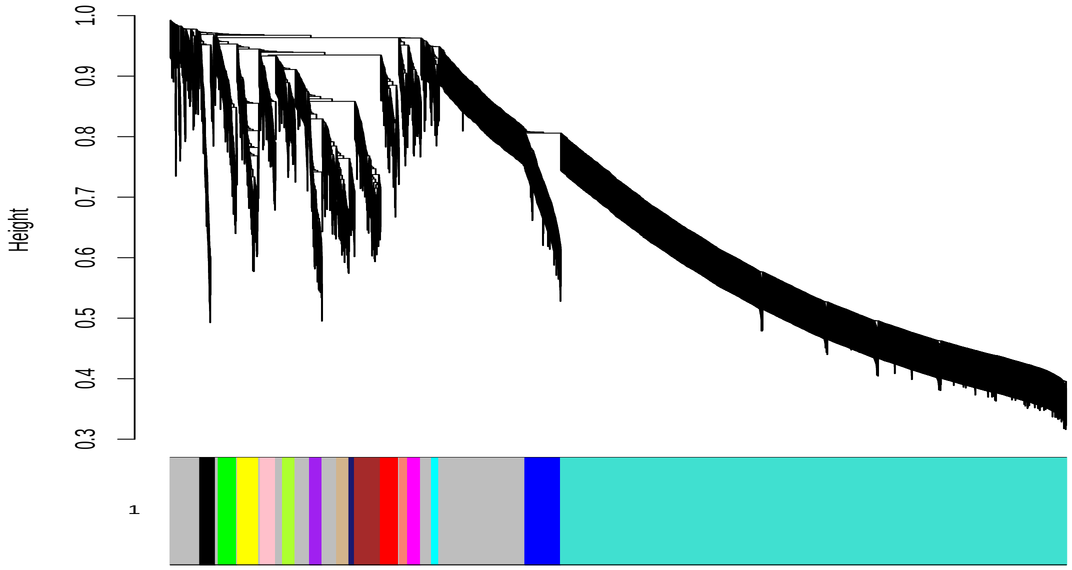


**Figure 4.33:** Average linkage hierarchical clustering dendrogram applied to dataset ALS1. Module assignment given by applying dynamic hybrid branch cutting with $deepSplit = 3$ and $minimumclustersize = 32$ is depicted by row color immediately below the dendrogram, with grey representing unassigned genes.

**Table 4.1:** Network properties applied to the four original datasets and the comparison with the average (and standard deviation) for each module present in the selected set of modules (grey module is not taken into consideration). The selected measures include size, density, centralization, Heterogeneity, Mean Cluster Coefficient and Mean Connectivity within cluster.

| Network | Mean Size | Mean Density | Mean Centralization | Mean Heterogeneity | Mean Cluster Coefficient | Mean Connectivity |
|---|---|---|---|---|---|---|
| original ALS 1 | 6777 | 0.378 | 0.182 | 0.293 | 0.448 | 2558 |
| selected ALS1 cut | 250.4 ± 425.8 | 0.246 ± 0.12 | 0.157 ± 0.03 | 0.378 ± 0.13 | 0.31 ± 0.13 | 52.0 ± 92.87 |
| original ALS 2 | 6777 | 0.446 | 0.183 | 0.330 | 0.544 | 3023 |
| selected ALS2 cut | 1660.4 ± 737.5 | 0.21 ± 0.11 | 0.16 ± 0.03 | 0.479 ± 0.14 | 0.303 ± 0.10 | 124.38 ± 290.7 |
| original Control 1 | 6777 | 0.276 | 0.135 | 0.219 | 0.316 | 1871 |
| selected Control 1 cut | 363.52 ± 353.3 | 0.147 ± 0.11 | 0.139 ± 0.04 | 0.520 ± 0.16 | 0.229 ± 0.13 | 85.46 ± 228.79 |
| original Control 2 | 6777 | 0.431 | 0.174 | 0.312 | 0.521 | 2921 |
| selected Control 2 cut | 957.99 ± 366.4 | 0.227 ± 0.13 | 0.158 ± 0.03 | 0.436 ± 0.16 | 0.312 ± 0.12 | 31.17 ± 33.44 |

## 4.3 Post-Processing Results

We have identified clusters of differentially expressed genes resulting from the clustering techniques (Sections 4.1.2 and 4.1.2) and modules of co-regulated genes resulting from the WGCNA procedure (Section 4.2). Now we are interested in understanding if these groups of genes are significantly related with each other. Then, functional annotation of these sets of genes is performed in order to identify relevant biological pathways or genes' functions associated with ALS disease.

### 4.3.1 Module Preservation in WGCNA networks

First, a study of the module preservation between networks of the same phenotype was performed to verify if the same modules were significantly present in networks of the same phenotype. Therefore, two approaches are used to determine this preservation: cross-tabulation and network based statistics.

#### 4.3.1.A Overlapping Matrix/Cross-tabulation

An overlapping matrix was built between the two module assignments of each pair of networks (identified in Section 4.2) accordingly to the described in Section 3.6.4. As when making use of the K-Means method, it was considered interesting to study the relationships between the module assignment of ALS1 and ALS2 (Figure 4.34), C1 and C2 (Figure 4.35) but also of ALS1 and C1 (Figure 4.36).

The WGCNA method produces a group of non-overlapping modules and so, module overlapping with more than one module in the other network is not unexpected. In fact, this is commonly verified when using this type of techniques [62]. When overlapping of two modules was verified, the module correspondence was made with the one presenting a least significant p-value, as used in [35]. By the analysis of these figures, one may identify a correspondence of each module in one network to the most significant similar module in the other. It is interesting to verify that in the case of ALS1 compared with ALS2, the grey module (of the unassigned genes) is very significantly overlapped, whilst in the other two pairwise comparisons, C1 and C2 (Figure 4.35) and ALS1 and C1 (Figure 4.36), this module significantly overlaps with more than one module. This indicates that the genes assigned to the grey module in one network are considered to have low topological overlap with other nodes and so are not assigned to any specific module. On the other hand, in the other network, they are considered to have high topological overlap with a significant number of other genes.

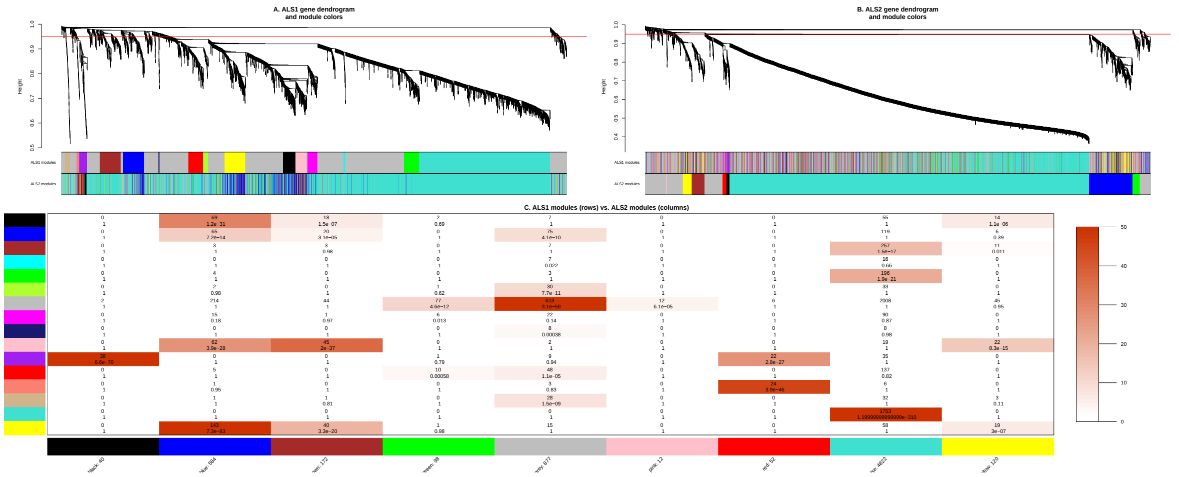


**Figure 4.34:** Overlap Table for the comparison of the modules identified in ALS1 and ALS2. (A) Dendrogram resulting from applying the WGCNA method to ALS1, where the bar code underneath indicates the module membership obtained by the cutting of this dendrogram (top) and the obtained cut in the ALS2 dendrogram. (B) Dendrogram resulting from applying the WGCNA method to ALS2, where the bar code underneath indicates the module membership obtained by the cutting of this dendrogram (top) and the obtained cut in the ALS1 dendrogram (bottom). (C) Cross-tabulation of the ALS1 modules (rows) and ALS2 modules (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side o the table.

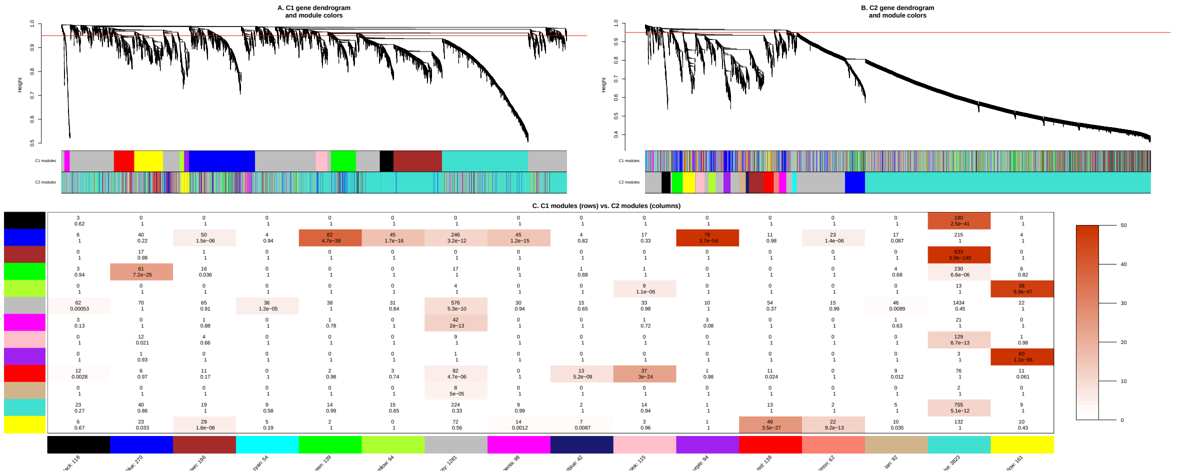


**Figure 4.35:** Overlap Table for the comparison of the modules identified in C1 and C2. (A) Dendrogram resulting from applying the WGCNA method to C1, where the bar code underneath indicates the module membership obtained by the cutting of this dendrogram (top) and the obtained cut in the C2 dendrogram. (B) Dendrogram resulting from applying the WGCNA method to C2, where the bar code underneath indicates the module membership obtained by the cutting of this dendrogram (top) and the obtained cut in the C1 dendrogram (bottom). (C) Cross-tabulation of the C1 modules (rows) and C2 modules (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side o the table.

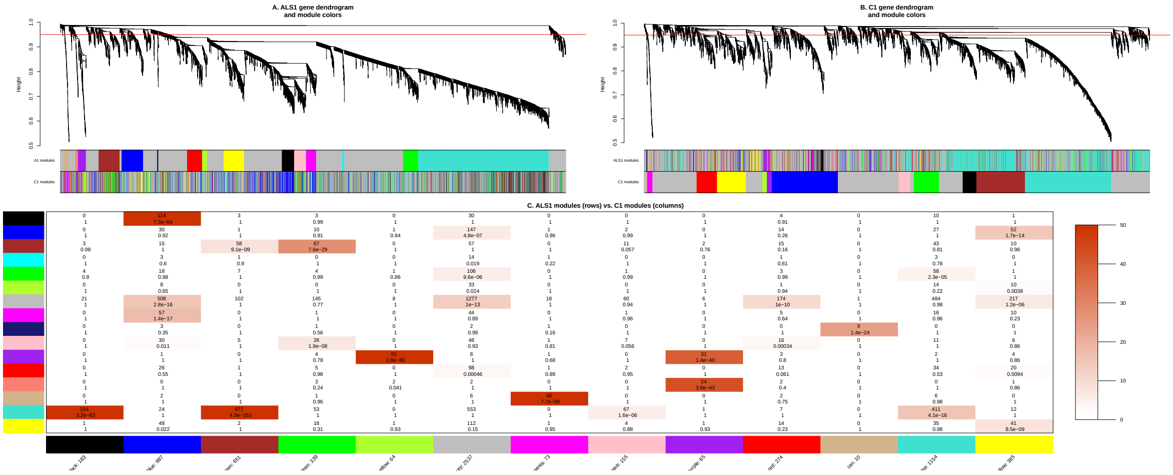


**Figure 4.36:** Overlap Table for the comparison of the modules identified in ALS1 and C1. Overlap Table for the comparison of the modules identified in ALS1 and C1. (A) Dendrogram resulting from applying the WGCNA method to ALS1, where the bar code underneath indicates the module membership obtained by the cutting of this dendrogram (top) and the obtained cut in the C1 dendrogram. (B) Dendrogram resulting from applying the WGCNA method to C1, where the bar code underneath indicates the module membership obtained by the cutting of this dendrogram (top) and the obtained cut in the ALS1 dendrogram (bottom). (C) Cross-tabulation of the ALS1 modules (rows) and C1 modules (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side o the table.

### 4.3.1.B Network Based Statistics

The network based statistics described in Section 3.6.4 were then applied to the selected cuts for the two ALS WGCNA networks (Figures 4.30 and 4.31) and the two control ones (Figures 4.32 and 4.33), with 50 permutations. The results are summarized by the Figures A.27, A.28, A.29 and A.30.

First, considering the comparison of the two ALS networks (Figures A.27 and A.28). A higher number of modules was selected for the first dataset, which was determined also by the structure of the resulting dendrogram (Figure 4.30). As stated, when $Z_{summary}$ is larger than 10 there is considerable confidence in stating that a module is preserved between the two networks. When using dataset 1 as a reference dataset (Figure A.27), most of the modules are above this limit, whilst the others present some evidence of preservation (as they are above the lower confidence limit). The modules considered as presenting only with some confidence of being preserved were the blue, turquoise, tan, salmon and brown (with respect to the ALS1 module labels). When using the dataset 2 as reference network (Figure A.28), there are stronger evidences of module preservation, which maybe due to the fact the number of modules identified is considerably less (almost half) as one may observe in Figure 4.31.

Comparing the controls WGCNA modules between the two control networks (Figures A.29 and A.30) result in better results. Using the same reasoning as above, there is enough evidence to claim that all modules are preserved regardless the network that is used as reference. Closer to the limit of the preservation are the tan and blue module (when considering C1 as reference - Figure A.29) and turquoise and salmon (with C2 as reference module set - Figure A.30).

Although some differences were detected with the standard clustering techniques between the two datasets, modules identified with WGCNA technique are considerably preserved between datasets which motivates the use of the dataset 1 as a reference result in further post-processing methods.

### 4.3.2 Module Eingengene Network

Genes inside a cluster present, in general, high connectivity when using Eigengene-based Connectivity described in equation 3, which accordingly to the literature provides enough evidences of the module quality [41, 45]. Analysing the dendrograms or the heatmaps in Figure 4.37, one may verify that some module eigengenes are highly correlated. Therefore, as the module's chosen representatives, the module eigengenes, present highly correlated expression values with other modules, one may hypothesize the over-partitioning of the data in the clustering procedure of WGCNA and the possibility that these would be merged if the parameters of the cutting algorithm would be different. Merging of similar modules was proposed by [45] by applying a fix height cut of module eigengenes' dendrogram. An attempt to merge modules using this approach was performed on the ALS1 network (Figure 4.30). However, the correlation between the resulting merged clusters was considerably higher than be the original ones in such way that not enough discrimination between the expression profiles was visible in the heatmap. Therefore, to refine the clusters, the previously described variation of the branch cutting parameters was chosen instead.

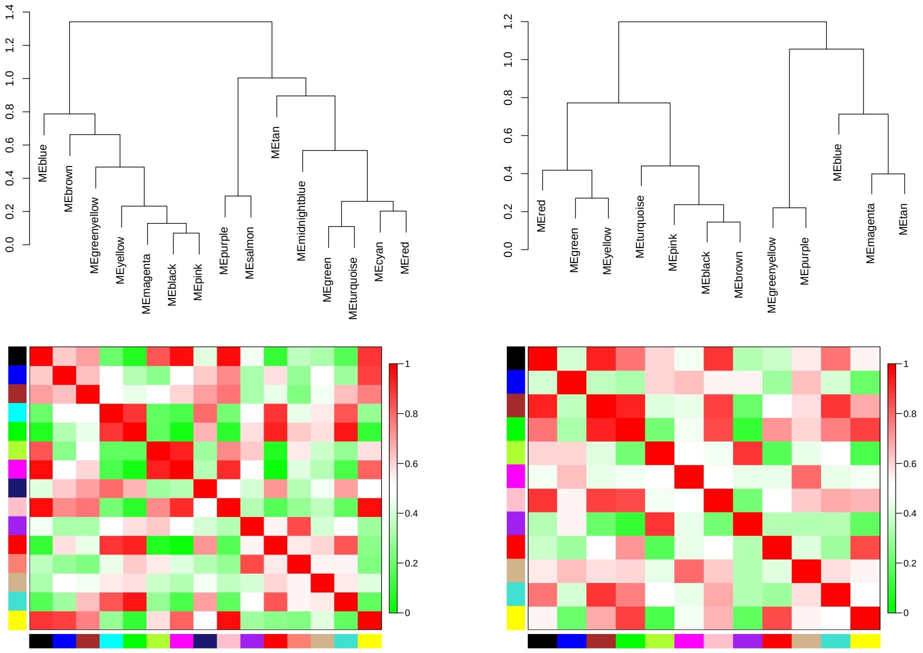


**Figure 4.37:** Study of the module eigengene relationships in WGCNA modules in ALS1 (left) and C1 (right): hierarchical clustering with average linkage and using as distance, 1 - (Pearson correlation), (top) and a heatmap that shows the Pearson correlation between modules in the same ALS network.

### 4.3.3 Comparing Partitions

#### 4.3.3.A Clustering Techniques

Before enrichment with functional annotation terms, a comparison of the overlap of the clusters obtained by the two clustering methods (identified in Section 4.1.2 and 4.1.2) was performed making use of the cross-tabulation method (Figures 4.38, 4.39 and 4.40). To do so, a vector of colors was defined for each set of clusters resulting from k-means and hierarchical algorithms, where only the interesting clusters were considered whilst all other genes were attributed to a grey module. As all the reasoning so far, all the comparisons were made regarding the dataset 1 (ALS1 and C1). In addition, it was also in this dataset that the clusters containing differentially expressed genes were identified.

By comparing Hierarchical clustering (HCL) resulting clusters with K-means for $k = 20$ it is clear that 3 of the 4 clusters resulting from the latter method are included in the blue cluster of HCL; the brown cluster of k-means is not identified by the HCL and the HCL turquoise module does not have a correspondence in the K-Means algorithm.

Further partition of the data occurs when the $k$ is equal to $50$ in the K-Means algorithm, leads to the assignment of the green k-means module to the unassigned HCL cluster. However, the HCL turquoise cluster is now related with one of the k-means (red cluster).

When comparing the two k-means runs, only three clusters are preserved (with the same labels on both datasets), which are the brown, turquoise and yellow cluster. Therefore, two clusters of the $k = 50$ run and one from the $k = 20$ are unpreserved.

When analysing the overlapping between HCL and K-Means the only clusters that were not observable in the the HCL were the brown ones (that is almost the same cluster in both k-means runs) and the green one using $k = 50$ that is not identified in the other run of the k-means method.

As a summary of the former comparisons, both runs of the k-means select a yellow and turquoise cluster that is also identified in the hierarchical clustering algorithm. Besides the turquoise cluster of HCL is only preserved when using the K-Means with $k = 50$.

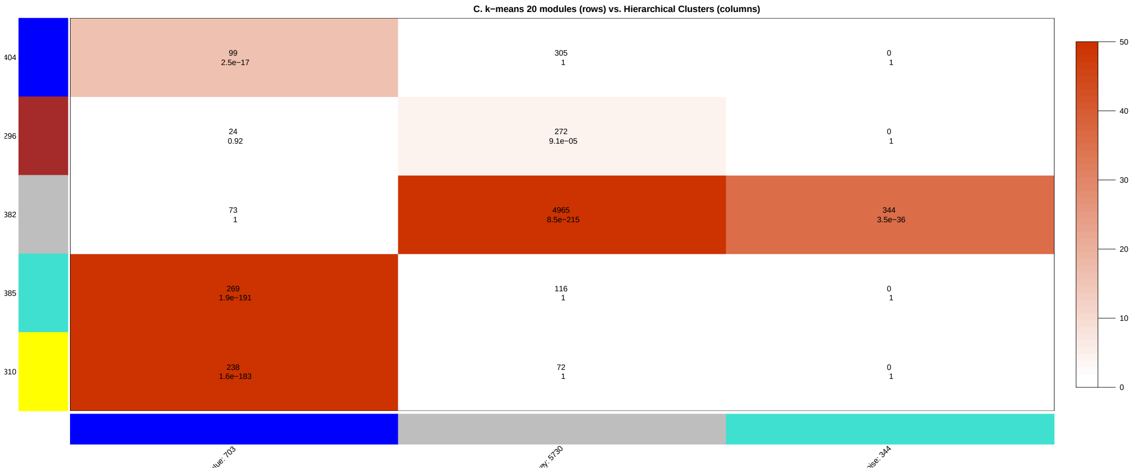


**Figure 4.38:** Cross-tabulation of the clusters obtained with k-means algorithm ($k = 20$) (rows) and clusters obtained by Hierarchical clustering (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side o the table.

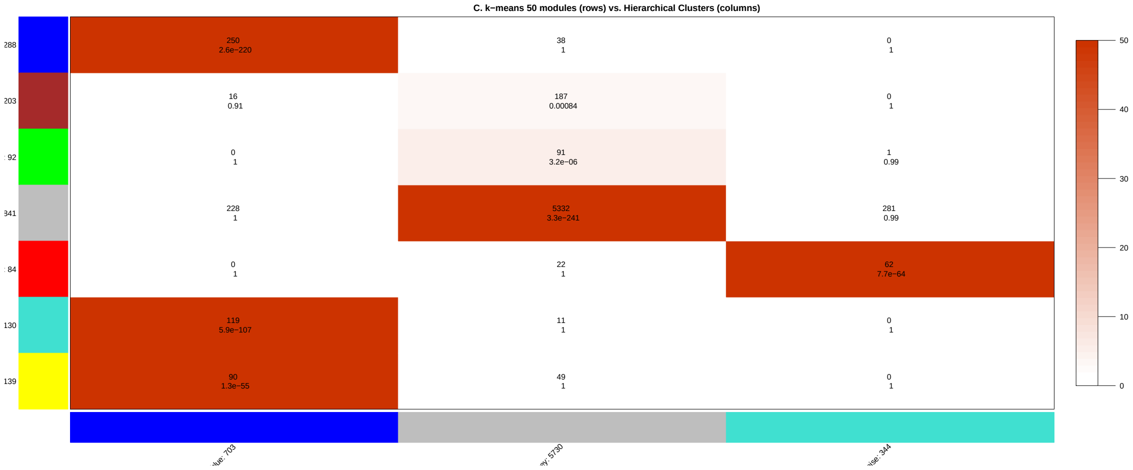


**Figure 4.39:** Cross-tabulation of the clusters obtained with k-means algorithm ($k = 20$) (rows) and clusters obtained by Hierarchical clustering (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side o the table.

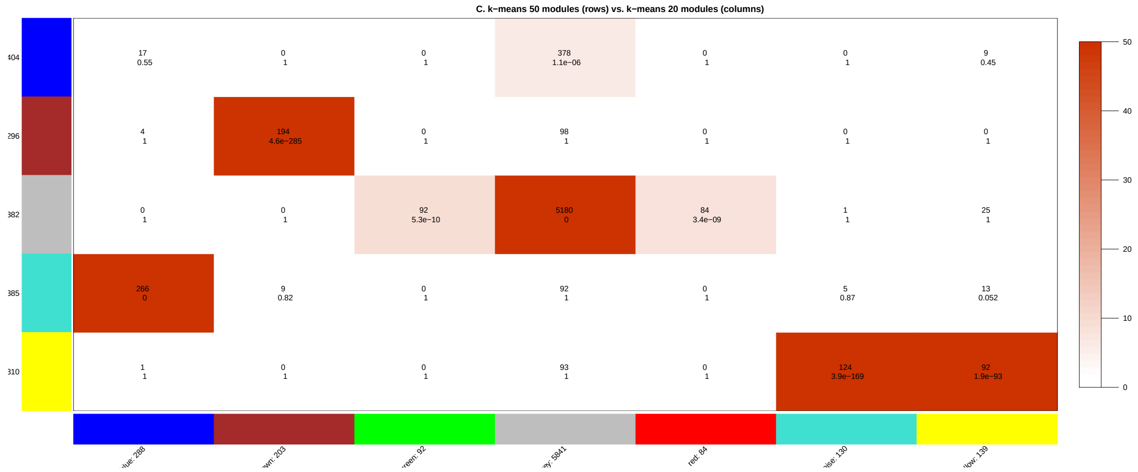


**Figure 4.40:** Cross-tabulation of the clusters obtained with k-means algorithm ($k = 20$) (rows) and k-means algorithm ($k = 50$) (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side o the table.

### 4.3.3.B Clustering Techniques against WGCNA

The cross-tabulation of modules was also applied to compare the clusters obtained in the standard clustering techniques and the WGCNA techniques. As some overlapping is verified in these clusters, it is expected some redundancy in the results of overlapping.

**HCL** Comparing the WGCNA modules with the hierarchical clusters produced interesting and easy to visualize results (Figures 4.41 and 4.42). In the ALS1 network (Figure 4.41), several modules match the hierarchical clusters, although being split most of them between the two clusters. These modules are: blue, brown, greenyellow, grey, pink, red and yellow (WGCNA label names).

Less modules of WGCNA control network match the hierarchical clusters (Figure 4.42). In fact, from a set of 12 modules, only 3 were significantly overlapped with the HCL clusters, which are: green, grey and yellow (WGCNA label names). Besides, no module from this network significantly intersects the turquoise HCL cluster, whilst in the ALS case it significantly intersects 4 modules.

To be noted that the number of genes from the WGCNA grey module that overlaps the hierarchical cluster, in both networks, is a very small subset of the complete module.

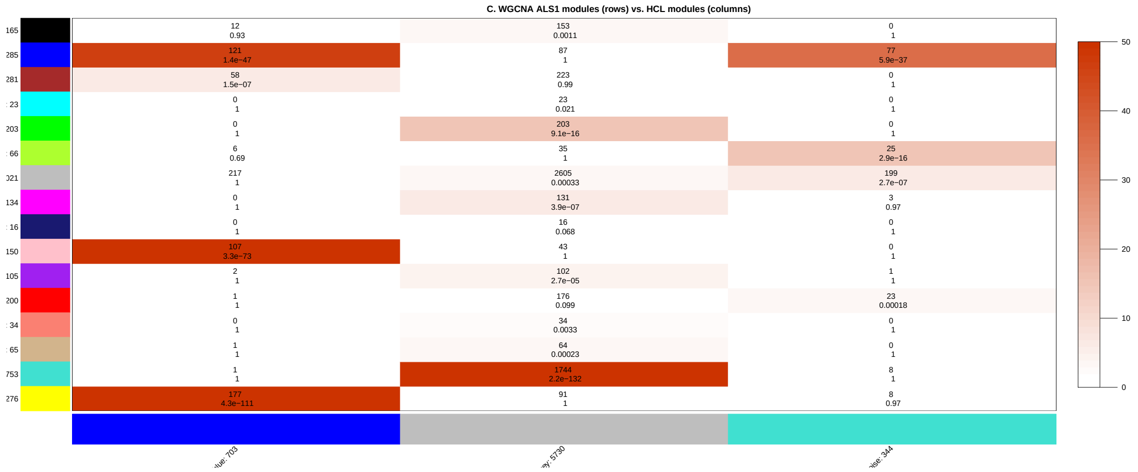


**Figure 4.41:** Cross-tabulation of the ALS1 WGCNA modules (rows) and clusters obtained by Hierarchical clustering (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side o the table.

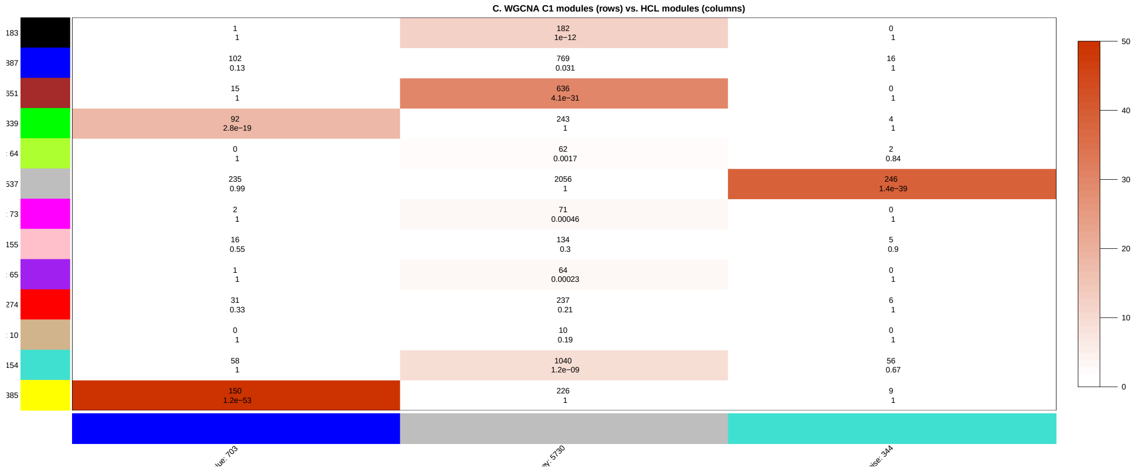
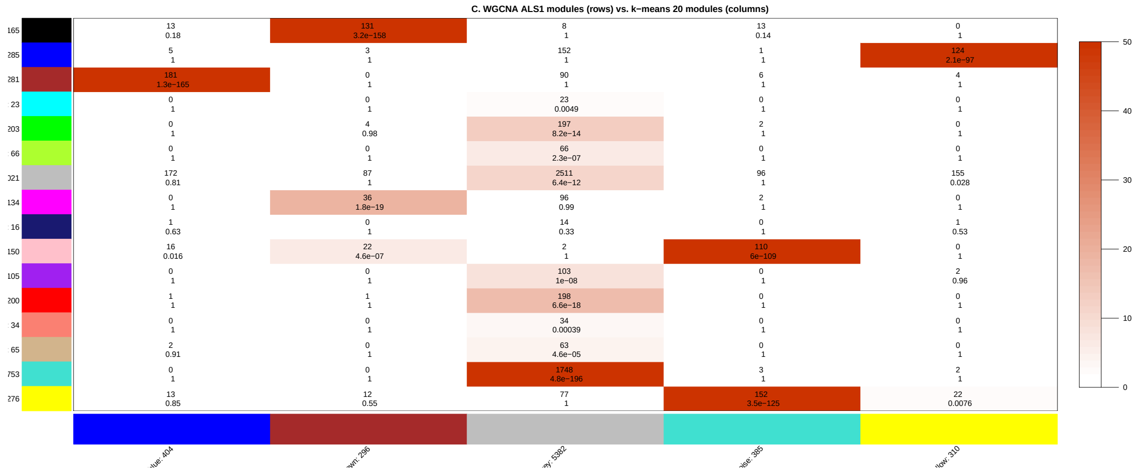


**Figure 4.42:** Cross-tabulation of the C1 WGCNA modules (rows) and clusters obtained by Hierarchical clustering (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side o the table.

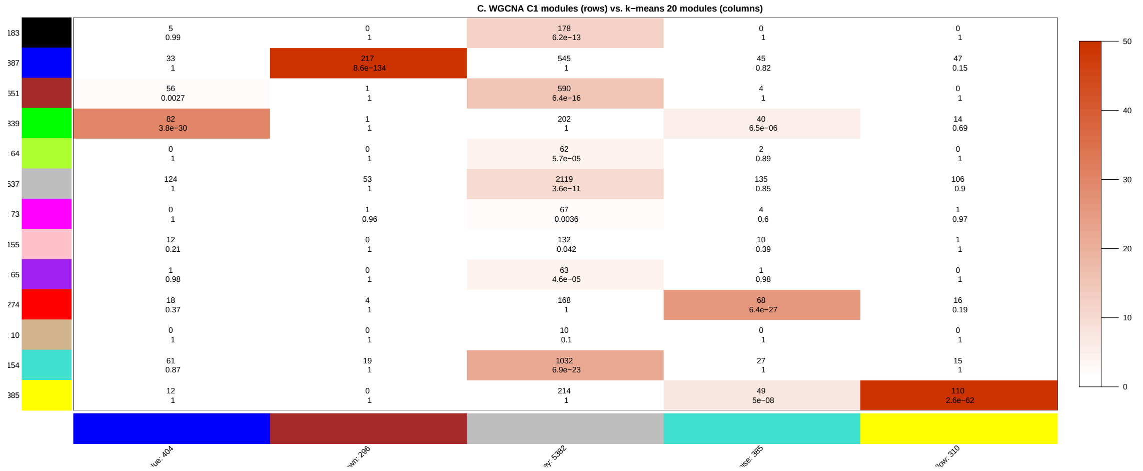
**K-Means** Most clusters that resulted from the k-means method were identified as overlapping the hierarchical clustering ones. Therefore, most overlaps between these clusters and WGCNA modules is expected. The only modules that were not overlap between HCL and K-Means are now overlapping significantly with WGCNA modules both in ALS1 and C1 (Figures 4.43, 4.44, 4.45 and 4.46).

Interestingly, ALS1 WGCNA network modules present better overlapping with clusters containing genes that present significant expression differences between ALS1 and C1 samples and thus further exploration of this dataset will be performed.



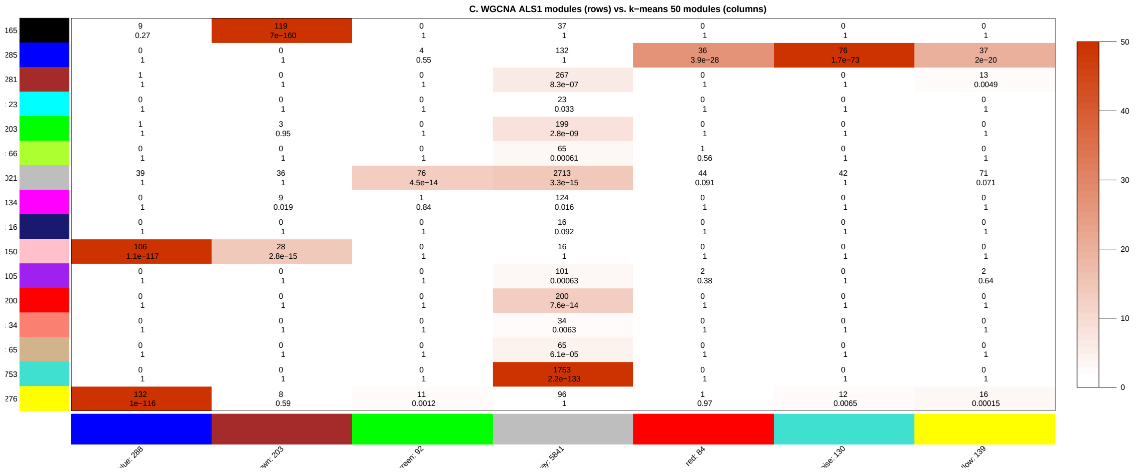
| | lue: 404 | wn: 296 | ey: 5382 | ise: 385 | low: 310 |
|---|---|---|---|---|---|
| 165 | 13<br>0.18 | 131<br>3.2e−158 | 8<br>1 | 13<br>0.14 | 0<br>1 |
| 285 | 5<br>1 | 3<br>1 | 152<br>1 | 1<br>1 | 124<br>2.1e−97 |
| 281 | 181<br>1.3e−165 | 0<br>1 | 90<br>1 | 6<br>1 | 4<br>1 |
| 23 | 0<br>1 | 0<br>1 | 23<br>0.0049 | 0<br>1 | 0<br>1 |
| 203 | 0<br>1 | 4<br>0.98 | 197<br>8.2e−14 | 2<br>1 | 0<br>1 |
| 66 | 0<br>1 | 0<br>1 | 66<br>2.3e−07 | 0<br>1 | 0<br>1 |
| 321 | 172<br>0.81 | 87<br>1 | 2511<br>6.4e−12 | 96<br>1 | 155<br>0.028 |
| 134 | 0<br>1 | 36<br>1.8e−19 | 96<br>0.99 | 2<br>1 | 0<br>1 |
| 16 | 1<br>0.63 | 0<br>1 | 14<br>0.33 | 0<br>1 | 1<br>0.53 |
| 150 | 16<br>0.016 | 22<br>4.6e−07 | 2<br>1 | 110<br>6e−109 | 0<br>1 |
| 105 | 0<br>1 | 0<br>1 | 103<br>1e−08 | 0<br>1 | 2<br>0.96 |
| 200 | 1<br>1 | 1<br>1 | 198<br>6.6e−18 | 0<br>1 | 0<br>1 |
| 34 | 0<br>1 | 0<br>1 | 34<br>0.00039 | 0<br>1 | 0<br>1 |
| 65 | 2<br>0.91 | 0<br>1 | 63<br>4.6e−05 | 0<br>1 | 0<br>1 |
| 753 | 0<br>1 | 0<br>1 | 1748<br>4.8e−196 | 3<br>1 | 2<br>1 |
| 276 | 13<br>0.85 | 12<br>0.55 | 77<br>1 | 152<br>3.5e−125 | 22<br>0.0076 |

**Figure 4.43:** Cross-tabulation of the ALS1 WGCNA modules (rows) and clusters obtained by k-means algorithm with $k = 20$ (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side of the table.



| | lue: 404 | wn: 296 | ey: 5382 | ise: 385 | low: 310 |
|---|---|---|---|---|---|
| 183 | 5<br>0.99 | 0<br>1 | 178<br>6.2e−13 | 0<br>1 | 0<br>1 |
| 387 | 33<br>1 | 217<br>8.6e−134 | 545<br>1 | 45<br>0.82 | 47<br>0.15 |
| 551 | 56<br>0.0027 | 1<br>1 | 590<br>6.4e−16 | 4<br>1 | 0<br>1 |
| 339 | 82<br>3.8e−30 | 1<br>1 | 202<br>1 | 40<br>6.5e−06 | 14<br>0.69 |
| 64 | 0<br>1 | 0<br>1 | 62<br>5.7e−05 | 2<br>0.89 | 0<br>1 |
| 537 | 124<br>1 | 53<br>1 | 2119<br>3.6e−11 | 135<br>0.85 | 106<br>0.9 |
| 73 | 0<br>1 | 1<br>0.96 | 67<br>0.0036 | 4<br>0.6 | 1<br>0.97 |
| 155 | 12<br>0.21 | 0<br>1 | 132<br>0.042 | 10<br>0.39 | 1<br>1 |
| 65 | 1<br>0.98 | 0<br>1 | 63<br>4.6e−05 | 1<br>0.98 | 0<br>1 |
| 274 | 18<br>0.37 | 4<br>1 | 168<br>1 | 68<br>6.4e−27 | 16<br>0.19 |
| 10 | 0<br>1 | 0<br>1 | 10<br>0.1 | 0<br>1 | 0<br>1 |
| 154 | 61<br>0.87 | 19<br>1 | 1032<br>6.9e−23 | 27<br>1 | 15<br>1 |
| 385 | 12<br>1 | 0<br>1 | 214<br>1 | 49<br>5e−08 | 110<br>2.6e−62 |

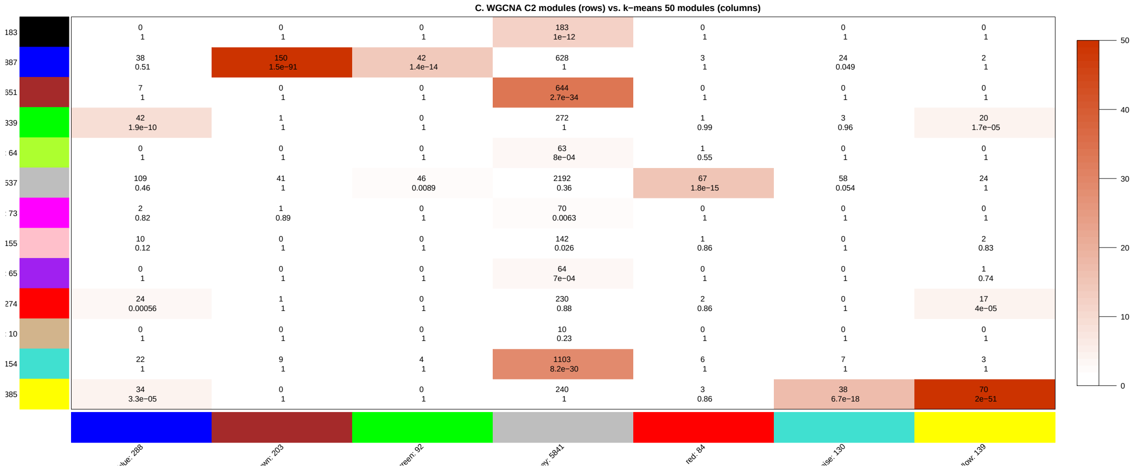
**Figure 4.44:** Cross-tabulation of the C1 WGCNA modules (rows) and clusters obtained by k-means algorithm with $k = 20$ (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side of the table.



C. WGCNA ALS1 modules (rows) vs. k−means 50 modules (columns)

|  | lue: 288 | wn: 203 | reen: 92 | ey: 5841 | red: 84 | ise: 130 | low: 139 |
|---|---|---|---|---|---|---|---|
| 165 | 9<br>0.27 | 119<br>7e−160 | 0<br>1 | 37<br>1 | 0<br>1 | 0<br>1 | 0<br>1 |
| 285 | 0<br>1 | 0<br>1 | 4<br>0.55 | 132<br>1 | 36<br>3.9e−28 | 76<br>1.7e−73 | 37<br>2e−20 |
| 281 | 1<br>1 | 0<br>1 | 0<br>1 | 267<br>8.3e−07 | 0<br>1 | 0<br>1 | 13<br>0.0049 |
| 23 | 0<br>1 | 0<br>1 | 0<br>1 | 23<br>0.033 | 0<br>1 | 0<br>1 | 0<br>1 |
| 203 | 1<br>1 | 3<br>0.95 | 0<br>1 | 199<br>2.8e−09 | 0<br>1 | 0<br>1 | 0<br>1 |
| 66 | 0<br>1 | 0<br>1 | 0<br>1 | 65<br>0.00061 | 1<br>0.56 | 0<br>1 | 0<br>1 |
| )21 | 39<br>1 | 36<br>1 | 76<br>4.5e−14 | 2713<br>3.3e−15 | 44<br>0.091 | 42<br>1 | 71<br>0.071 |
| 134 | 0<br>1 | 9<br>0.019 | 1<br>0.84 | 124<br>0.016 | 0<br>1 | 0<br>1 | 0<br>1 |
| 16 | 0<br>1 | 0<br>1 | 0<br>1 | 16<br>0.092 | 0<br>1 | 0<br>1 | 0<br>1 |
| 150 | 106<br>1.1e−117 | 28<br>2.8e−15 | 0<br>1 | 16<br>1 | 0<br>1 | 0<br>1 | 0<br>1 |
| 105 | 0<br>1 | 0<br>1 | 0<br>1 | 101<br>0.00063 | 2<br>0.38 | 0<br>1 | 2<br>0.64 |
| 200 | 0<br>1 | 0<br>1 | 0<br>1 | 200<br>7.6e−14 | 0<br>1 | 0<br>1 | 0<br>1 |
| 34 | 0<br>1 | 0<br>1 | 0<br>1 | 34<br>0.0063 | 0<br>1 | 0<br>1 | 0<br>1 |
| 65 | 0<br>1 | 0<br>1 | 0<br>1 | 65<br>6.1e−05 | 0<br>1 | 0<br>1 | 0<br>1 |
| 753 | 0<br>1 | 0<br>1 | 0<br>1 | 1753<br>2.2e−133 | 0<br>1 | 0<br>1 | 0<br>1 |
| 276 | 132<br>1e−116 | 8<br>0.59 | 11<br>0.0012 | 96<br>1 | 1<br>0.97 | 12<br>0.0065 | 16<br>0.00015 |

**Figure 4.45:** Cross-tabulation of the ALS1 WGCNA modules (rows) and clusters obtained by k-means algorithm with $k = 50$ (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side of the table.



C. WGCNA C2 modules (rows) vs. k−means 50 modules (columns)

|  | lue: 288 | wn: 203 | reen: 92 | ey: 5841 | red: 84 | ise: 130 | low: 139 |
|---|---|---|---|---|---|---|---|
| 183 | 0<br>1 | 0<br>1 | 0<br>1 | 183<br>1e−12 | 0<br>1 | 0<br>1 | 0<br>1 |
| 387 | 38<br>0.51 | 150<br>1.5e−91 | 42<br>1.4e−14 | 628<br>1 | 3<br>1 | 24<br>0.049 | 2<br>1 |
| 551 | 7<br>1 | 0<br>1 | 0<br>1 | 644<br>2.7e−34 | 0<br>1 | 0<br>1 | 0<br>1 |
| 339 | 42<br>1.9e−10 | 1<br>1 | 0<br>1 | 272<br>1 | 1<br>0.99 | 3<br>0.96 | 20<br>1.7e−05 |
| 64 | 0<br>1 | 0<br>1 | 0<br>1 | 63<br>8e−04 | 1<br>0.55 | 0<br>1 | 0<br>1 |
| 537 | 109<br>0.46 | 41<br>1 | 46<br>0.0089 | 2192<br>0.36 | 67<br>1.8e−15 | 58<br>0.054 | 24<br>1 |
| 73 | 2<br>0.82 | 1<br>0.89 | 0<br>1 | 70<br>0.0063 | 0<br>1 | 0<br>1 | 0<br>1 |
| 155 | 10<br>0.12 | 0<br>1 | 0<br>1 | 142<br>0.026 | 1<br>0.86 | 0<br>1 | 2<br>0.83 |
| 65 | 0<br>1 | 0<br>1 | 0<br>1 | 64<br>7e−04 | 0<br>1 | 0<br>1 | 1<br>0.74 |
| 274 | 24<br>0.00056 | 1<br>1 | 0<br>1 | 230<br>0.88 | 2<br>0.86 | 0<br>1 | 17<br>4e−05 |
| 10 | 0<br>1 | 0<br>1 | 0<br>1 | 10<br>0.23 | 0<br>1 | 0<br>1 | 0<br>1 |
| 154 | 22<br>1 | 9<br>1 | 4<br>1 | 1103<br>8.2e−30 | 6<br>1 | 7<br>1 | 3<br>1 |
| 385 | 34<br>3.3e−05 | 0<br>1 | 0<br>1 | 240<br>1 | 3<br>0.86 | 38<br>6.7e−18 | 70<br>2e−51 |

**Figure 4.46:** Cross-tabulation of the C1 WGCNA modules (rows) and clusters obtained by k-means algorithm with $k = 50$ (columns), where the colors respects the color code defined. In the table, numbers give the number of genes that overlap between the corresponding row and column module. The color-code in this table is given by the Fisher exact test p value, $-log(p)$, accordingly to the bar coded given in the right side of the table.

### 4.3.4 Study of ALS1 WGCNA network

**Analysing complete set of WGCNA modules** First, expression plots of the entire set of genes in WGCNA modules was used in an attempt to observe modules where genes present a different expression profile between ALS and control samples. Using ALS1 modules, 4 modules present some evidences of differential expression (Figure 4.47): black, brown, pink and yellow. It is interesting to verify that all of these modules showed overlapping with clusters resulting from standard clustering techniques.

Using the same criteria of differential expression, 2 modules present differences in expression of ALS and control samples although with less evidence as in C1 case (Figure 4.48): green and yellow. When comparing to the hierarchical clustering clusters, these ones were the only showing overlapping.

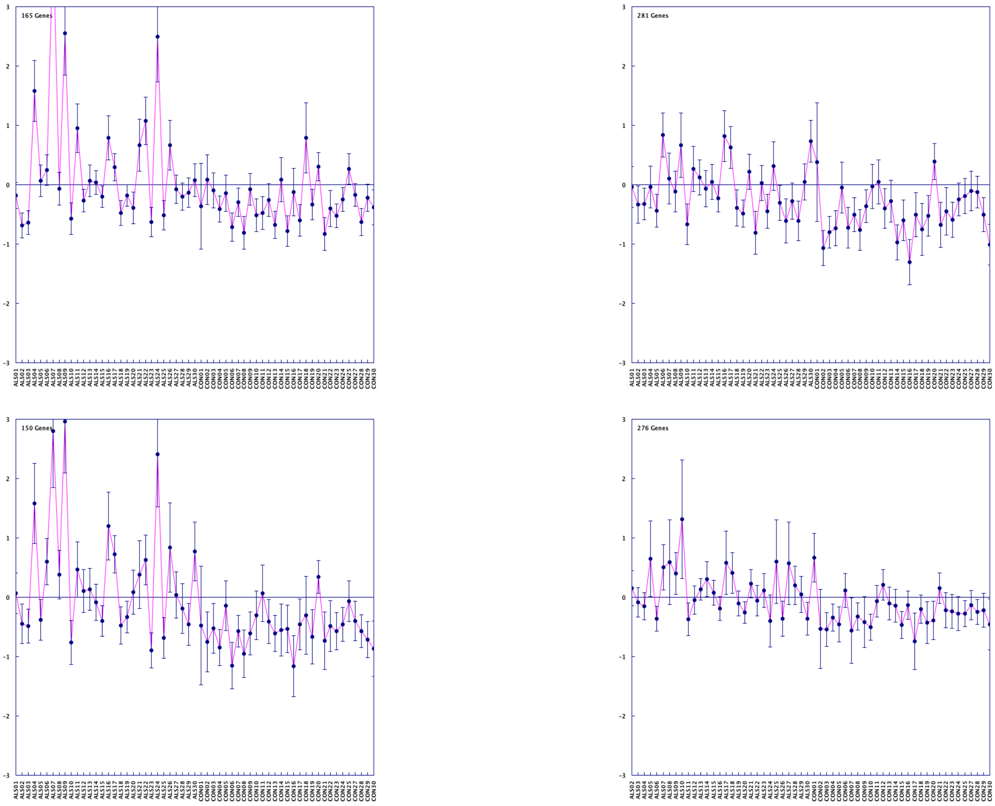


**Figure 4.47:** Centroid view of the modules obtained for ALS1 with WGCNA: black(top left), brown (top right), pink (bottom left) and yellow (bottom right).

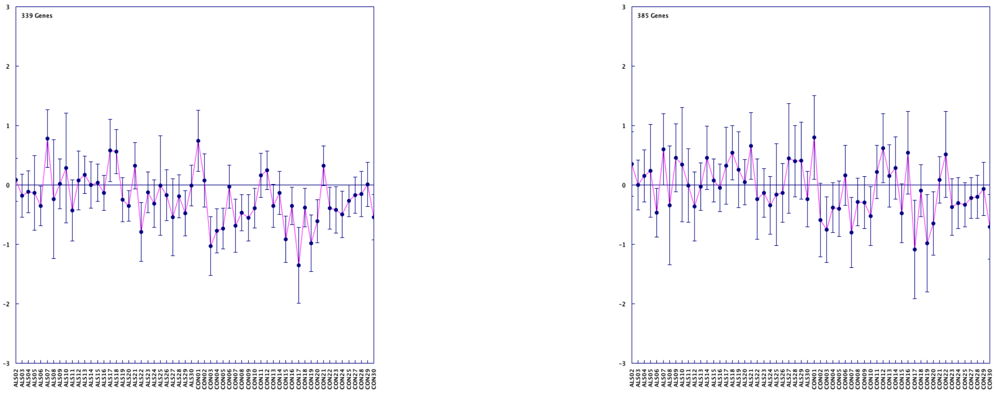


**Figure 4.48:** Centroid view of the modules obtained for C1 with WGCNA: green (left) and yellow (right).

**Analysing the most connected genes in the WGCNA modules** When making use of WGCNA, one is interested in the most important genes in the module. Saris et al. [5] used the 100 top most connected genes in a given module, whilst others [47] enriched the entire modules. Therefore, the connectivity measure based on the correlation with the module eigengene (Equation 3) was applied to the genes in the identified modules of ALS 1 network. Then, a rank of the module genes with respect to this connectivity measure was obtained. The results showed that a large subset of the genes in a module were highly correlated with the respective module eigengene. Therefore, a relatively high threshold ($= 0.85$) was used to select the most connected genes to the module eigengene.

The motivation behind the choice of a threshold, in the module assigned genes, from the ranked gene connectivity was to guarantee that the genes with similar relationships to the module eigengene are the ones selected. In fact, if one uses a fix threshold by number of genes to include in these ranked lists, some modules may present only highly connected genes (such as the turquoise module) while others may contain a wide range of connectivities. On the other hand, the computation was not extended to the entire gene set as we were interested in comparing the resulting modules from this approach with clustering techniques, where the same selectivity was not applied. Therefore, for a fair comparison of methods, the subset of genes used in the next steps are only subgroups of the identified modules or clusters.

When considering only the most connected genes, two more clusters are identified as presenting genes with expression significantly different between ALS and control samples (Figure 4.49) that were not considered before reducing the number of genes (Figure 4.50).

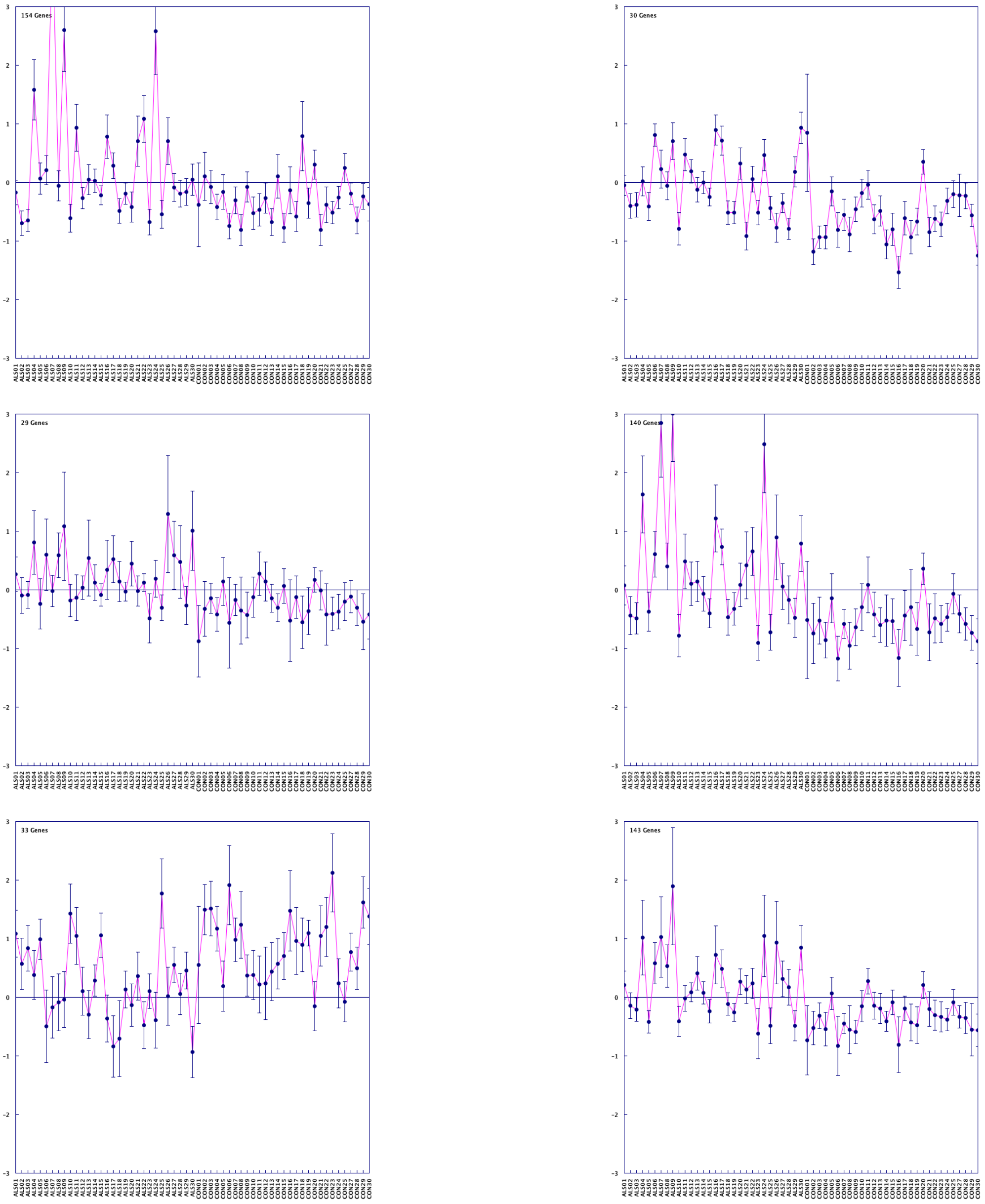


**Figure 4.49:** Centroid view of the clusters obtained for ALS1 with WGCNA (described left to right: black, brown, blue, pink, tan, yellow).

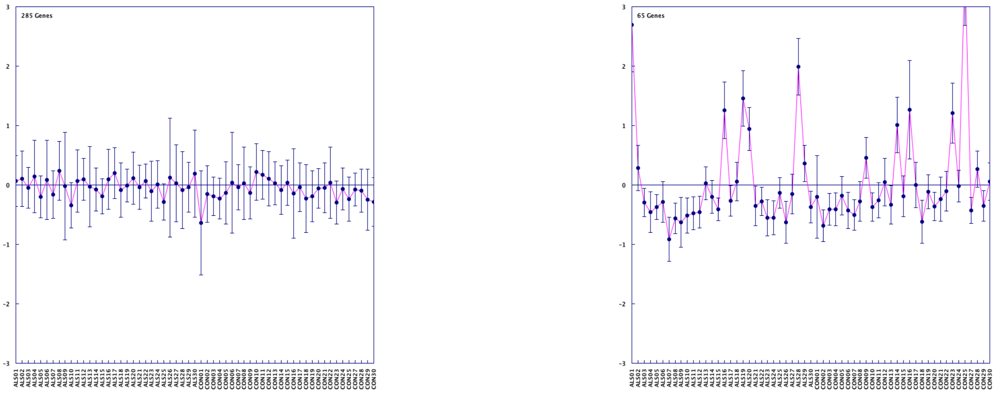


**Figure 4.50:** Centroid view of the 2 complete modules obtained for ALS with WGCNA: blue (left) and tan (right).

### 4.3.5 Functional annotation of clusters

Functional annotation of clusters was then performed by comparing each of them against the entire human genome and using Babelomics [50]. The results for modules in WGCNA are presented in Table 4.2, considering only the most connected genes, and for clusters obtained by HCL and K-means in Table 4.3 and Table 4.4, respectively. Whenever several GO terms are considered, only the further from the root of the graph are discriminated.

One module from the WGCNA method and one cluster from k-means (using both values of $k$) were identified with three neurodegenerative diseases (Alzheimer's disease, Parkinson's disease and Huntington's Disease), which may indicate the involvement of the same subset of genes in all of them. In depth study of the present results will be performed as future work.

| Module | Type of term | Term | p-value |
|---|---|---|---|
| Yellow | GO | regulation of protein metabolic process (GO:0051246) | 1.12e-3 |
| | | negative regulation of biosynthetic process (GO:0009890) | 3.78e-2 |
| | | small GTPase mediated signal transduction (GO:0007264) | 1.22e-3 |
| | | endocytosis (GO:006897) | 3.51e-2 |
| | | nucleocytoplasmic transport (GO:0006913) | 9.23e-3 |
| | | intracellular protein transport (GO:0006886) | 6.47e-5 |
| Salmon and Red | KEGG | Ribosome (hsa03010) | 6.03e-7 |
| | GO | translational elongation (GO: 0006414) | 1.57e-6 |
| MidnightBlue | GO | chemotaxis (GO:0006935) | 3.13e-2 |
| | | defense response to fungus (GO:00508332) | 1.04e-5 |
| | | defense response to Gram-negative bacterium (GO:0050829) | 2.33e-3 |
| | | response to virus (GO:0009615) | 6.43e-3 |
| | | xenobiotic metabolic process (GO:0006805) | 1.48e-4 |
| Purple | GO | DNA recombination (GO:0006310) | 1.99e-2 |
| Black | KEGG | Ribosome (hsa03010) | 1.76e-42 |
| | | Oxidative phosphorylation (hsa00190) | 5.76e-9 |
| | | Alzheimer's disease (hsa05010) | 5.76e-9 |
| | | Parkinson's disease (hsa05012) | 4.16e-8 |
| | | Huntington's disease (hsa05016) | 7.06e-8 |
| | | RNA degradation (hsa03018) | 2.38e-3 |
| | | Spliceosome (hsa03040) | 1.42e-2 |
| | | Cardiac muscle contraction (hsa04260) | 3.66e-2 |

**Table 4.2:** GO and KEGG significantly associated with WGCNA modules (considering the most connected genes only). For 3 modules (tan, pink and blue) no significant terms were identified, whilst in the turquoise module the relevant GO terms are presented in A.31.

| Cluster | Type of term | Term | p-value |
|---|---|---|---|
| Turquoise | KEGG | Phosphatidylinositol signaling system (hsa04070) | 4.12e-2 |
| | | Glycolysis / Gluconeogenesis (hsa00010) | 4.42e-2 |
| | | Fructose and mannose metabolism (hsa00051) | 4.42e-2 |
| | | RNA polymerase (hsa03020) | 4.42e-2 |
| | | Huntington's disease (hsa05016) | 4.16e-2 |
| | | T cell receptor signaling pathway (hsa04660) | 8.84e-3 |
| | | Natural killer cell mediated cytotoxicity (hsa04650) | 4.01e-2 |
| | | O-Mannosyl glycan biosynthesis (hsa00514) | 4.01e-2 |
| | GO | Several terms are annotated (refer to Figure A.33) | |
| Blue | KEGG | Ubiquitin mediated proteolysis (hsa04120) | 1.74e-2 |
| | | Long-term potentiation (hsa04720) | 1.74e-2 |
| | | RNA degradation (hsa03018) | 4.81e-2 |
| | GO | Several terms are annotated (refer to Figure A.32) | |

**Table 4.3:** GO and KEGG significantly associated with hierarchical clustering results.

| Cluster | Type of term | Term | p-value |
|---|---|---|---|
| Turquoise-k20 | KEGG | RNA degradation (hsa03018) | 2.29e-3 |
| | GO | Several terms are annotated (refer to Figure A.34) | |
| Blue-k20 | KEGG | Spliceosome (hsa03040) | 9.41e-3 |
| | GO | Several terms are annotated (refer to Figure A.35) | |
| Brown-k20 | KEGG | Ribosome (hsa03010) | 1.88e-71 |
| | | Oxidative phosphorylation (hsa00190) | 6.56e-23 |
| | | Cardiac muscle contraction (hsa04260) | 1.13e-5 |
| | | Alzheimer's disease (hsa05010) | 5.89e-16 |
| | | Parkinson's disease (hsa05012) | 1.18e-19 |
| | | Huntington's disease (hsa05016) | 1.59e-16 |
| | | RNA degradation (hsa03018) | 2.83e-2 |
| | | Protein export (hsa03060) | 3.8e-3 |
| | | Proteasome (hsa03050) | 1.36e-2 |
| | | Spliceosome (hsa03040) | 1.04e-4 |
| | GO | Several terms are annotated (refer to Figure A.36) | |
| Yellow-k20 | GO | Several terms are annotated (refer to Figure A.37) | |
| Turquoise-k50 | GO | Response to wounding (GO:0009611) | 1.42e-2 |
| Blue-k50 | KEGG | RNA degradation (hsa03018) | 3.01e-2 |
| | GO | protein targeting to membrane (GO:0006612) | 1.49e-2 |
| | | endosome to lysosome transport (GO:0008333) | 4.24e-2 |
| | | deoxyribonucleotide catabolic process (GO:0009264) | 9.89e-3 |
| | | histone modification (GO: 0016570) | 1.07e-2 |
| | | negative regulation of transcription (GO:0016481) | 4.76e-2 |
| | | mRNA processing (GO:0006397) | 4.22e-2 |
| | | RNA splicing (GO:0008380) | 4.24e-2 |
| Brown-k50 | KEGG | Ribosome (hsa03010) | 1.19e-38 |
| | | Oxidative phosphorylation (hsa00190) | 4.281e-12 |
| | | Alzheimer's disease (hsa05010) | 9.93e-9 |
| | | Parkinson's disease (hsa05012) | 2.40e-11 |
| | | Huntingont's disease (hsa05016) | 1.58e-7 |
| | | RNA degradation (hsa03018) | 8.62e-3 |
| | GO | Several terms are annotated (refer to Figure A.38) | |
| Green-k50 | KEGG | Neurotrophin signaling pathway (hsa04722) | 1.34e-4 |
| | | Graft-versus-host disease (hsa05332) | 3.16e-2 |
| | | Natural killer cell mediated cytotoxicity (hsa04650) | 3.16e-2 |
| | | Acute myeloid leukemia (hsa05221) | 3.73e-2 |
| | | Toll-like receptor signaling pathway (hsa04620) | 2.76e-2 |
| | | B cell receptor signaling pathway (hsa04662) | 8.76e-3 |
| | | Insulin signaling pathway (hsa04910) | 8.76e-4 |
| | GO | protein kinase cascade (GO:0007243) | 9.41e-3 |
| | | regulation of signal transduction (GO:0009966) | 9.41e-3 |
| | | small GTPase mediated signal transduction (GO:0007264) | 9.94e-3 |
| | | Ras protein signal transduction (GO:0007265) | 9.94e-3 |

**Table 4.4:** GO and KEGG significantly associated with k-means clustering results. Modules of $k = 20$ and $k = 50$ are respectively named '-k20' and '-k50'.

## 4.4 Conclusions and Future work

The first task of the present work consisted in data preprocessing and enable to identify the possible presence of batch effects that greatly influence the results when using standard clustering techniques such as k-means. Since WGCNA used Pearson correlation to extract the similarity features between genes in the dataset a method based on this measure is also more adequate to detect outliers as such, other selection procedures were used in this case. A more recent technique, also used in this work, that takes into consideration the network properties, such as connectivity and clustering coefficient, was shown to be more appropriate to this work. Moreover, it allows to better select not only outliers but also the dataset that would better discriminate between the classes under study.

Hierarchical clustering using average linkage was then applied to identify clusters with significant differences of gene expression profiles between ALS and control samples. Only two large clusters were identified with this method with the expected properties. K-means with four values of $k$ was then applied to all samples, dataset 1 and dataset 2. However, only in dataset 1 we observed the desirable characteristics of discrimination between ALS and controls, and only with $k = 20$ and $k = 50$. Four clusters were identified with the first and six with the higher value of $50$.

Finally, WGCNA method was applied to the four subsets of data, two with ALS samples and two with controls. A variation of the module generating algorithm was studied and network concepts, namely the clustering coefficient, was used to selected the set of modules with generally more cohesive nature. Different numbers of modules were obtained for each network. Therefore, a study of the overlapping between these modules was first performed with a cross-tabulation and then with network statistics based method. The second was considered more relevant in the context of studying the module preservation between WGCNA networks and showed a high confidence of module preservation between networks of the belonging to the same class.

The WGCNA method offers the possibility of using network concepts well studied and applied to complex networks that aid in the decision process and understanding of the structure of the results. For each module of dataset 1, a module eigengene was computed, that is considered the module representative as it describes the dimension that better explains most of the gene expression variation in that module. The most used connectivity measure in this type of networks, calculates the correlation of all genes in the module relatively to the module eigengene and was computed. A threshold of 0.85 was set on a list of ranked absolute gene's connectivity to select the most important, which were then considered for functional annotation.

A comparison of the modules identified with the latter module and the clusters obtained with the standard clustering techniques was then performed. The only set of genes that significantly intersects all other sets is the WGNA modules obtained for the ALS samples in dataset 1.

All these methods resulted in relevant groups of genes, which are functional enriched with several biological potentially interesting biological functions for the problem under study.

As future work, further study of the terms functional annotated in the resulting clusters and modules may result in interesting contributions to the understanding of this disorder. Further comparisons of these methods as well as improvement in the confidence of the results would be possible if a higher number of samples is considered.

# Bibliography

[1] K. Venkova-Hristova, A. Christov, Z. Kamaluddin, P. Kobalka, and K. Hensley, "Progress in Therapy Development for Amyotrophic Lateral Sclerosis." *Neurology research international*, vol. 2012, p. 187234, Jan. 2012. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3399448&tool=pmcentrez&rendertype=abstract

[2] J. Perry, "Amyotrophic lateral sclerosis: update and new developments," *Degenerative Neurological and Neuromuscular Disease*, vol. 2012, no. 2, p. 1, Feb. 2012. [Online]. Available: /pmc/articles/PMC3457793/?report=abstract

[3] J. Brettschneider, J. B. Toledo, V. M. Van Deerlin, L. Elman, L. McCluskey, V. M.-Y. Lee, and J. Q. Trojanowski, "Microglial activation correlates with disease progression and upper motor neuron clinical symptoms in amyotrophic lateral sclerosis." *PloS one*, vol. 7, no. 6, p. e39216, Jan. 2012. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3375234&tool=pmcentrez&rendertype=abstract

[4] K. Talbot and O. Ansorge, "Recent advances in the genetics of amyotrophic lateral sclerosis and frontotemporal dementia: common pathways in neurodegenerative disease." *Human molecular genetics*, vol. 15 Spec No, no. 2, pp. R182–7, Oct. 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16987882

[5] C. G. J. Saris, S. Horvath, P. W. J. van Vught, M. a. van Es, H. M. Blauw, T. F. Fuller, P. Langfelder, J. DeYoung, J. H. J. Wokke, J. H. Veldink, L. H. van den Berg, and R. a. Ophoff, "Weighted gene co-expression network analysis of the peripheral blood from Amyotrophic Lateral Sclerosis patients." *BMC genomics*, vol. 10, p. 405, Jan. 2009. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2743717&tool=pmcentrez&rendertype=abstract

[6] N. Kim, P. Andrews, F. W.Asselbergs, H. R. Frost, S. M.Williams, B. T. Harris, C. Read, K. Askland, and J. H. Moore, "Gene ontology analysis of pairwise genetic associations in two genome-wide studies of sporadic ALS," *BioData Mining*, vol. 5, no. 1, p. 9, 2012. [Online]. Available: http://www.biodatamining.org/content/5/1/9

[7] L. Zinman and M. Cudkowicz, "Emerging targets and treatments in amyotrophic lateral sclerosis," *The Lancet Neurology*, vol. 10, no. 5, pp. 481–490, 2011. [Online]. Available: http://dx.doi.org/10.1016/S1474-4422(11)70024-2

[8] A. Al-Chalabi, A. Jones, C. Troakes, A. King, S. Al-Sarraj, and L. H. van den Berg, “The genetics and neuropathology of amyotrophic lateral sclerosis.” *Acta neuropathologica*, vol. 124, no. 3, pp. 339–52, Sep. 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/22903397

[9] A. M. Strutt, J. Palcic, J. G. Wager, C. Titus, C. Macadam, J. Brown, B. M. Scott, Y. Harati, P. E. Schulz, and M. K. York, “Cognition, behavior, and respiratory function in amyotrophic lateral sclerosis.” *ISRN neurology*, vol. 2012, p. 912123, Jan. 2012. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3407622&tool=pmcentrez&rendertype=abstract

[10] M. C. Kiernan, S. Vucic, B. C. Cheah, M. R. Turner, A. Eisen, O. Hardiman, J. R. Burrell, and M. C. Zoing, “Amyotrophic lateral sclerosis.” *Lancet*, vol. 377, no. 9769, pp. 942–55, Mar. 2011. [Online]. Available: http://www.thelancet.com/journals/a/article/PIIS0140-6736(10)61156-7/fulltext

[11] C. L. Kolarcik and R. Bowser, “Retinoid signaling alterations in amyotrophic lateral sclerosis,” vol. 1, no. 2, pp. 130–145, 2012.

[12] “ALSoD: Amyotrophic Lateral Sclerosis Online Genetics Database.” [Online]. Available: http://alsod.iop.kcl.ac.uk/

[13] “ALSGene.” [Online]. Available: http://www.alsgene.org/methods.asp

[14] C. Genomics, “Finishing the euchromatic sequence of the human genome.” *Nature*, vol. 431, no. 7011, pp. 931–45, Oct. 2004. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15496913

[15] A. Faiz and J. K. Burgess, “How can microarrays unlock asthma?” *Journal of allergy*, vol. 2012, p. 241314, Jan. 2012. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3303677&tool=pmcentrez&rendertype=abstract

[16] A. J. Iafrate, L. Feuk, M. N. Rivera, M. L. Listewnik, P. K. Donahoe, Y. Qi, S. W. Scherer, and C. Lee, “Detection of large-scale variation in the human genome.” *Nature genetics*, vol. 36, no. 9, pp. 949–51, Sep. 2004. [Online]. Available: http://dx.doi.org/10.1038/ng1416

[17] S. Choudhuri, “Microarrays in biology and medicine.” *Journal of biochemical and molecular toxicology*, vol. 18, no. 4, pp. 171–9, Jan. 2004. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15452887

[18] M. C. Pirrung, “How to Make a DNA Chip,” *Angewandte Chemie International Edition*, vol. 41, no. 8, pp. 1276–1289, Apr. 2002. [Online]. Available: http://doi.wiley.com/10.1002/1521-3773(20020415)41:8⟨1276::AID-ANIE1276⟩3.0.CO;2-2

[19] M. J. Heller, “DNA microarray technology: devices, systems, and applications.” *Annual review of biomedical engineering*, vol. 4, pp. 129–53, Jan. 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12117754

[20] a. Schulze and J. Downward, “Navigating gene expression using microarrays–a technology review.” *Nature cell biology*, vol. 3, no. 8, pp. E190–5, Aug. 2001. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/11483980

[21] M. R. Dalman, A. Deeter, G. Nimishakavi, and Z.-H. Duan, “Fold change and p-value cutoffs significantly alter microarray interpretations.” p. S11, Jan. 2012. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3305783&tool=pmcentrez&rendertype=abstract

[22] I. Barbulovic-Nad, M. Lucente, Y. Sun, M. Zhang, A. R. Wheeler, and M. Bussmann, “Bio-microarray fabrication techniques–a review.” *Critical reviews in biotechnology*, vol. 26, no. 4, pp. 237–59, 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17095434

[23] S. Selvaraj and J. Natarajan, “Microarray data analysis and mining tools.” *Bioinformation*, vol. 6, no. 3, pp. 95–9, Jan. 2011. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3089881&tool=pmcentrez&rendertype=abstract

[24] J. Quackenbush, “Computational analysis of microarray data.” *Nature reviews. Genetics*, vol. 2, no. 6, pp. 418–27, Jun. 2001. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17051302

[25] G. C. Tseng, D. Ghosh, and E. Feingold, “Comprehensive literature review and statistical considerations for microarray meta-analysis.” *Nucleic acids research*, vol. 40, no. 9, pp. 3785–99, May 2012. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3351145&tool=pmcentrez&rendertype=abstract

[26] P. C. Boutros and A. B. Okey, “Unsupervised pattern recognition : An introduction to the whys and wherefores of clustering microarray data,” vol. 6, no. 4, pp. 331–344, 2005.

[27] W. Zhao, P. Langfelder, T. Fuller, J. Dong, A. Li, and S. Hovarth, “Weighted gene coexpression network analysis: state of the art.” *Journal of Biopharmaceutical Statistics*, vol. 20, no. 2, pp. 281–300, 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20309759

[28] P. Valarmathie, “Survey on Clustering Algorithms for Microarray Gene Expression Data,” vol. 69, no. 1, pp. 5–20, 2012.

[29] A. de la Fuente, “From ’differential expression’ to ’differential networking’ - identification of dysfunctional regulatory networks in diseases.” *Trends in genetics : TIG*, vol. 26, no. 7, pp. 326–33, Jul. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.tig.2010.05.001

[30] W. Shannon, R. Culverhouse, and J. Duncan, “Analyzing microarray data using cluster analysis.” *Pharmacogenomics*, vol. 4, no. 1, pp. 41–52, Jan. 2003. [Online]. Available: http://www.futuremedicine.com/doi/abs/10.1517/phgs.4.1.41.22581?journalCode=pgs

[31] H. Chipman, T. J. Hastie, and R. Tibshirani, “Clustering Microarray Data,” pp. 161–204, 2001.

[32] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, “A gene-coexpression network for global discovery of conserved genetic modules.” *Science (New York, N.Y.)*, vol. 302, no. 5643, pp. 249–55, Oct. 2003. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12934013

[33] P. D’haeseleer, “How does gene expression clustering work?” *Nature biotechnology*, vol. 23, no. 12, pp. 1499–501, Dec. 2005. [Online]. Available: http://dx.doi.org/10.1038/nbt1205-1499

[34] C. J. Wolfe, I. S. Kohane, and A. J. Butte, “Systematic survey reveals general applicability of ”guilt-by-association” within gene coexpression networks.” *BMC bioinformatics*, vol. 6, p. 227, Jan. 2005. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1239911&tool=pmcentrez&rendertype=abstract

[35] M. C. Oldham, G. Konopka, K. Iwamoto, P. Langfelder, T. Kato, S. Horvath, and D. H. Geschwind, “Functional organization of the transcriptome in human brain.” *Nature neuroscience*, vol. 11, no. 11, pp. 1271–82, Nov. 2008. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2756411&tool=pmcentrez&rendertype=abstract

[36] G. Stolovitzky, R. J. Prill, and A. Califano, “Lessons from the DREAM2 Challenges.” *Annals of the New York Academy of Sciences*, vol. 1158, pp. 159–95, Mar. 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19348640

[37] L. F. Costa and F. A. Rodrigues, “Characterization of Complex Networks : A Survey of measurements,” 2008.

[38] B. Zhang and S. Horvath, “A General Framework for Weighted Gene Co-Expression Network Analysis A General Framework for Weighted Gene Co-Expression Network Analysis,” vol. 4, no. 1, 2005.

[39] J. D. Allen, Y. Xie, M. Chen, L. Girard, and G. Xiao, “Comparing statistical methods for constructing large scale gene networks.” *PloS one*, vol. 7, no. 1, p. e29348, Jan. 2012. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0029348

[40] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabási, “Hierarchical organization of modularity in metabolic networks.” *Science (New York, N.Y.)*, vol. 297, no. 5586, pp. 1551–5, Aug. 2002. [Online]. Available: http://www.sciencemag.org/content/297/5586/1551.abstract

[41] P. Langfelder and S. Horvath, “WGCNA: an R package for weighted correlation network analysis.” *BMC bioinformatics*, vol. 9, p. 559, Jan. 2008. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2631488&tool=pmcentrez&rendertype=abstract

[42] “The R Project for Statistical Computing.” [Online]. Available: http://www.r-project.org/index.html

[43] J. Dong and S. Horvath, “Understanding network concepts in modules.” *BMC systems biology*, vol. 1, no. 1, p. 24, Jan. 2007. [Online]. Available: http://www.biomedcentral.com/1752-0509/1/24

[44] M. R. J. Carlson, B. Zhang, Z. Fang, P. S. Mischel, S. Horvath, and S. F. Nelson, “Gene connectivity, function, and sequence conservation: predictions from modular yeast co-expression networks.” *BMC genomics*, vol. 7, p. 40, Jan. 2006. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1413526&tool=pmcentrez&rendertype=abstract

[45] P. Langfelder and S. Horvath, "Eigengene networks for studying the relationships between co-expression modules." *BMC systems biology*, vol. 1, no. 1, p. 54, Jan. 2007. [Online]. Available: http://www.biomedcentral.com/1752-0509/1/54

[46] M. J. Mason, G. Fan, K. Plath, Q. Zhou, and S. Horvath, "Signed weighted gene co-expression network analysis of transcriptional regulation in murine embryonic stem cells." *BMC genomics*, vol. 10, p. 327, Jan. 2009. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2727539&tool=pmcentrez&rendertype=abstract

[47] J. a. Miller, S. Horvath, and D. H. Geschwind, "Divergence of human and mouse brain transcriptome highlights Alzheimer disease pathways," *Proceedings of the National Academy of Sciences*, vol. 107, no. 28, pp. 12 698–12 703, Jun. 2010. [Online]. Available: http://www.pnas.org/cgi/doi/10.1073/pnas.0914257107

[48] "The Gene Ontology." [Online]. Available: http://www.geneontology.org/

[49] "KEGG Overview." [Online]. Available: http://www.genome.jp/kegg/kegg1a.html

[50] "Babelomics4, gene expression and functional profiling analysis suite." [Online]. Available: http://babelomics.bioinfo.cipf.es/

[51] "Bioinformatics Graz - Software." [Online]. Available: http://genome.tugraz.at/genesisclient/genesisclient_description.shtml

[52] E. Freyhult, M. Landfors, J. Önskog, T. R. Hvidsten, and P. Rydén, "Challenges in microarray class discovery: a comprehensive examination of normalization, gene selection and clustering." *BMC bioinformatics*, vol. 11, no. 1, p. 503, Jan. 2010. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3098084&tool=pmcentrez&rendertype=abstract

[53] "Ailun: Platform Annotation." [Online]. Available: http://ailun.stanford.edu/platformAnnotation.php

[54] Q. Li, N. J. Birkbak, B. Gyorffy, Z. Szallasi, and A. C. Eklund, "Jetset: selecting the optimal microarray probe set to represent a gene," *BMC Bioinformatics*, vol. 12, no. 1, p. 474, 2011. [Online]. Available: http://www.biomedcentral.com/1471-2105/12/474

[55] M. C. Oldham, P. Langfelder, and S. Horvath, "Network methods for describing sample relationships in genomic datasets: application to Huntington's disease." *BMC systems biology*, vol. 6, no. 1, p. 63, Jan. 2012. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3441531&tool=pmcentrez&rendertype=abstract

[56] A. Sturn, "Cluster Analysis for Large Scale Gene Expression Studies," Ph.D. dissertation, 2000.

[57] P. Langfelder, B. Zhang, and S. Horvath, "Dynamic Tree Cut : in-depth description , tests and applications," pp. 1–11, 2009.

[58] Z. Huang, “Clustering large data sets with mixed numerical and categorical values,” ser. Proceedings of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining. Word Scientific.

[59] ——, “A fast clustering algorithm to cluster very large categorical data sets in data mining,” ser. In Proc. SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997.

[60] A. M. Yip and S. Horvath, “Gene network interconnectedness and the generalized topological overlap measure.” *BMC bioinformatics*, vol. 8, p. 22, Jan. 2007. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1797055&tool=pmcentrez&rendertype=abstract

[61] S. Horvath, B. Zhang, M. Carlson, K. V. Lu, S. Zhu, R. M. Felciano, M. F. Laurance, W. Zhao, S. Qi, Z. Chen, Y. Lee, a. C. Scheck, L. M. Liau, H. Wu, D. H. Geschwind, P. G. Febbo, H. I. Kornblum, T. F. Cloughesy, S. F. Nelson, and P. S. Mischel, “Analysis of oncogenic signaling networks in glioblastoma identifies ASPM as a molecular target.” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 46, pp. 17 402–7, Nov. 2006. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1635024&tool=pmcentrez&rendertype=abstract

[62] P. Langfelder, R. Luo, M. C. Oldham, and S. Horvath, “Is my network module preserved and reproducible?” *PLoS computational biology*, vol. 7, no. 1, p. e1001057, Jan. 2011. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3024255&tool=pmcentrez&rendertype=abstract

# A

# Experimental Results Appendix

## A.1 Clustering of Genes

### A.1.1 Hierarchical Clustering

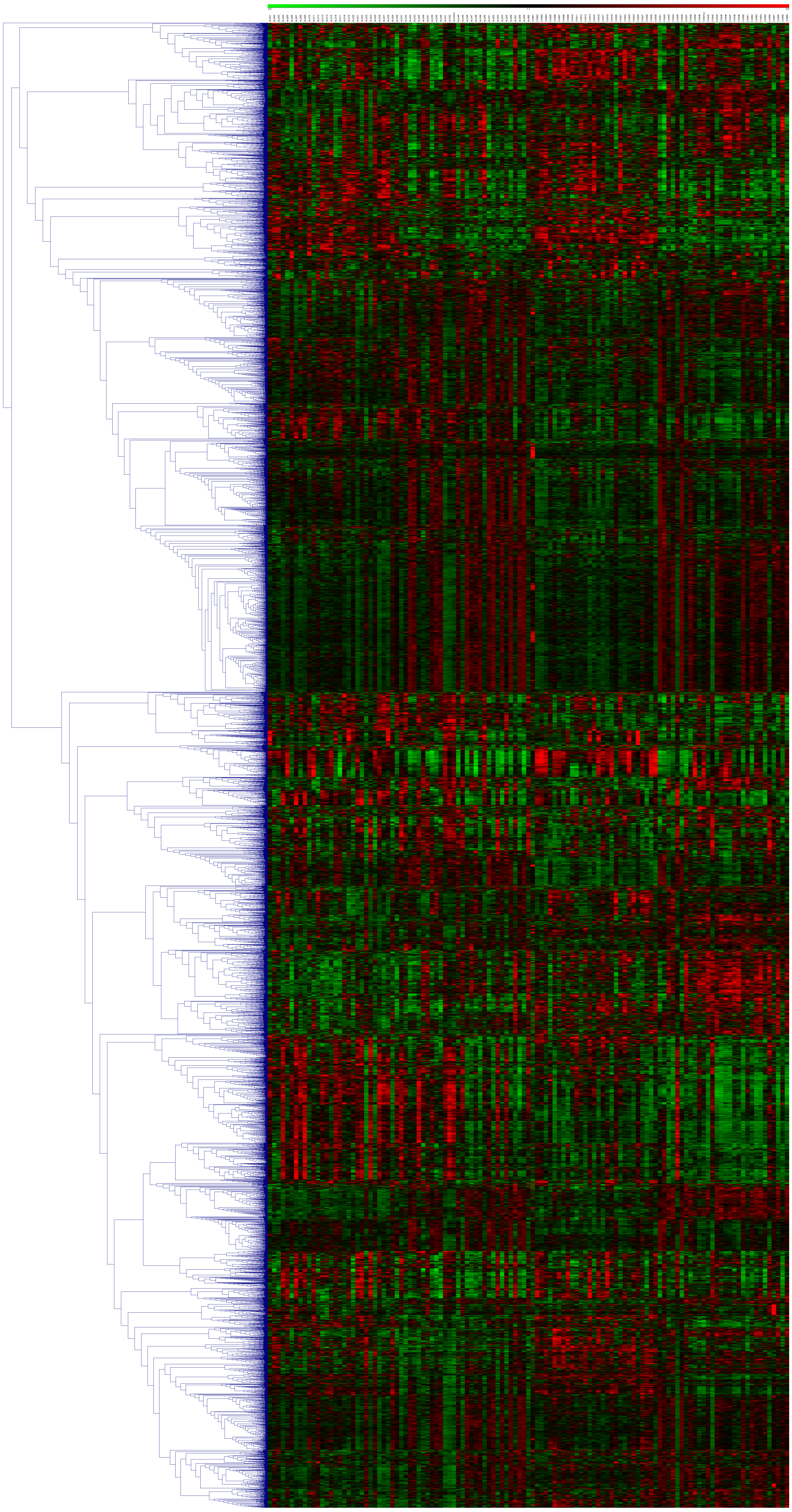


Figure A.1: Average Linkage Hierarchical clustering based on Euclidean distance applied to cluster genes in all samples. Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width).

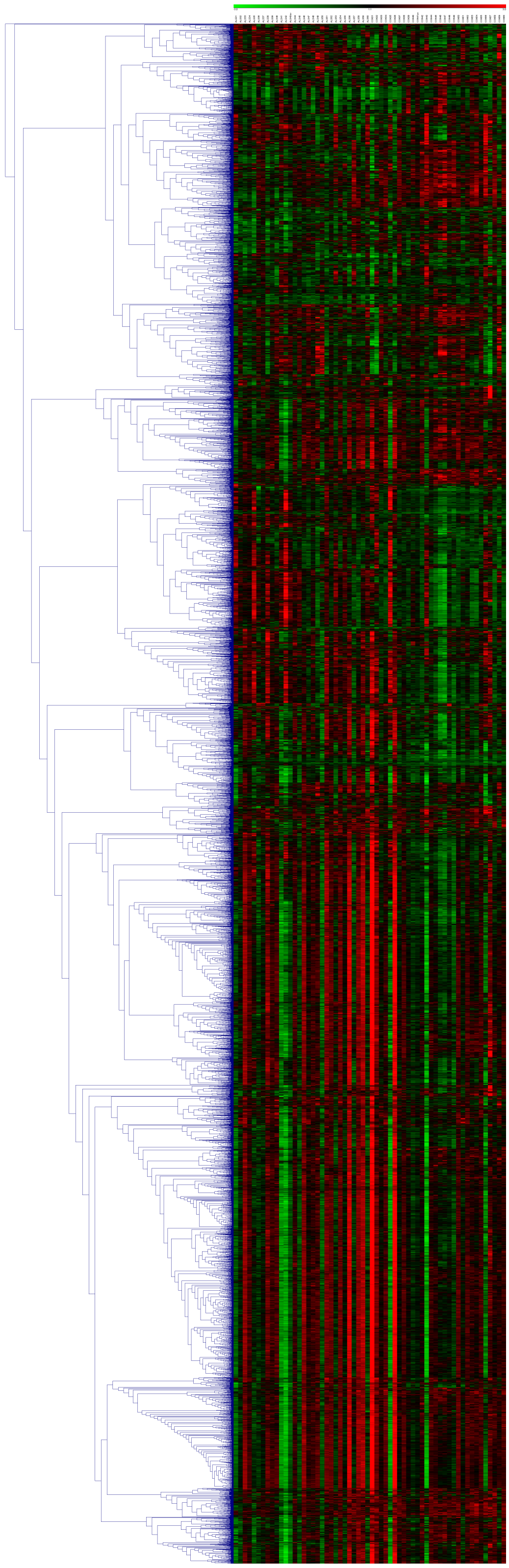


**Figure A.2:** Average linkage hierarchical clustering based on Euclidean distance applied to cluster genes in the combination of ALS2 and C2. Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width).

## A.1.2 K-Means

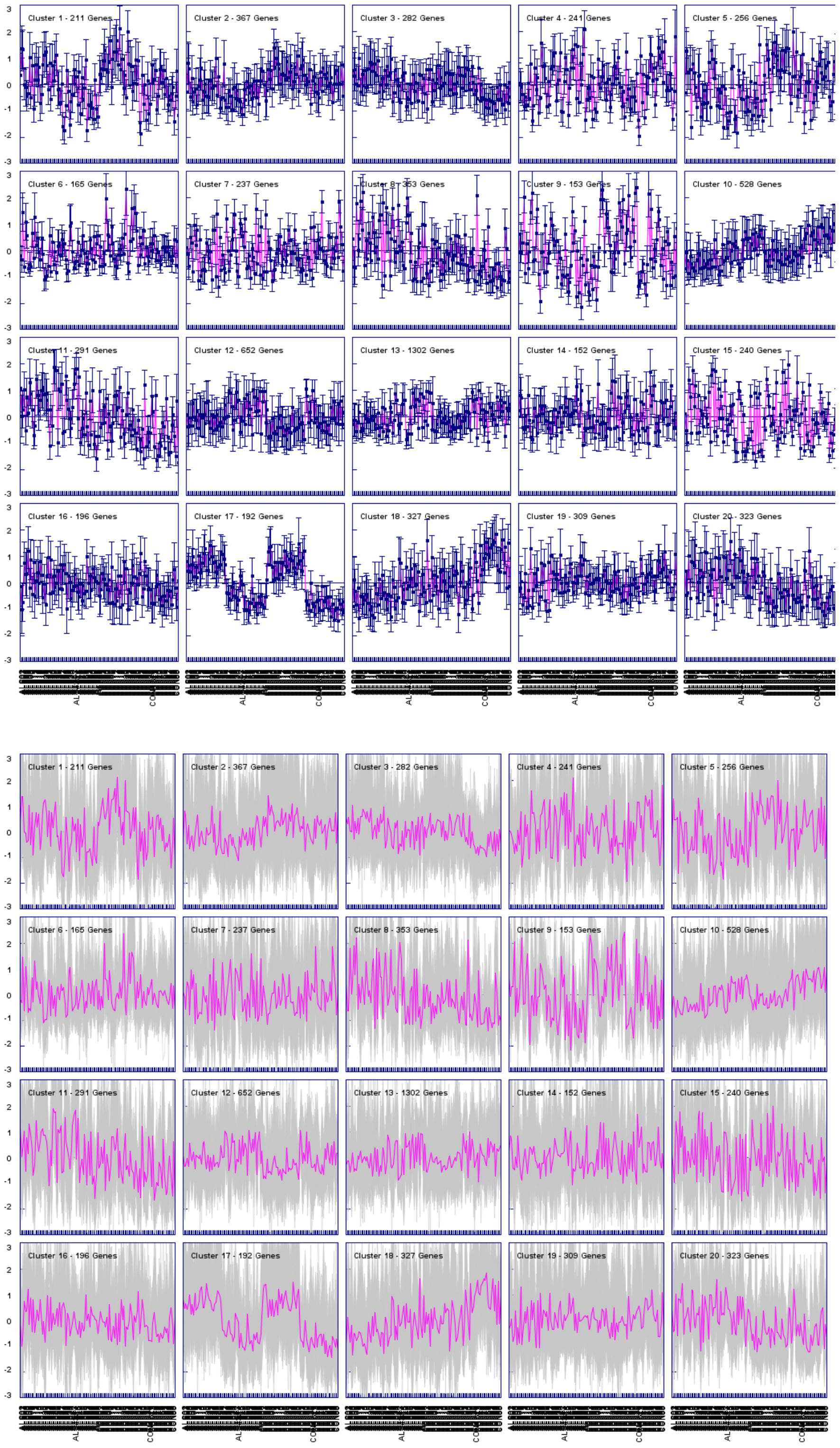


**Figure A.3:** Resulting clustering from applying the K-Means algorithm with $k = 20$ and using Euclidean distance to cluster genes in the entire dataset after sample and gene normalization and with batch effects correction. Centroid view of the clusters (top) and their mean expression (bottom). Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width).

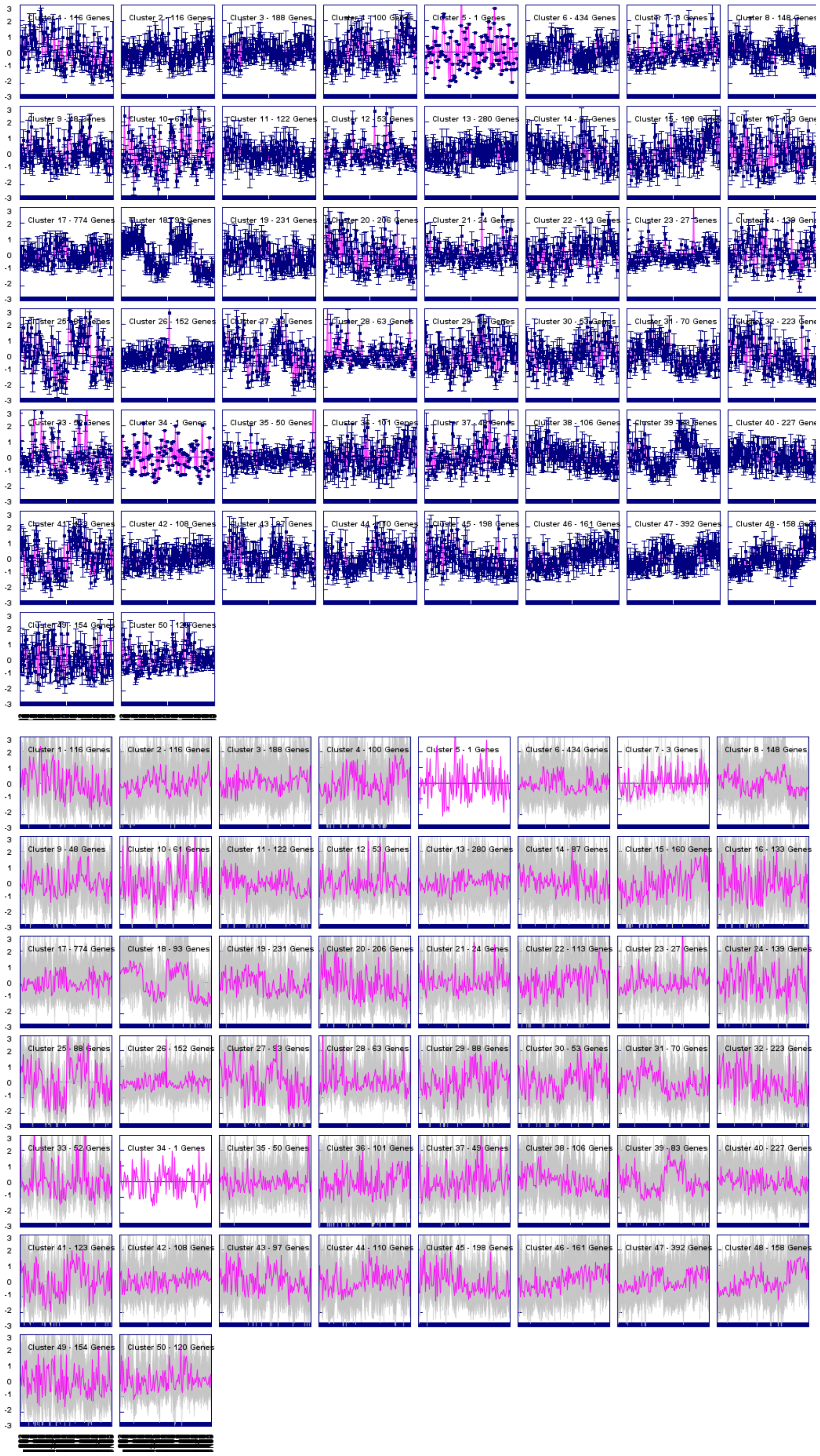


**Figure A.4:** Centroid view of the clusters (left) and their mean expression pattern (right) of applying the k-means algorithm with $k = 50$ and euclidean distance to cluster genes in all datasets. Dataset order from left to right: ALS1, ALS2, C1, C2 (each one corresponds to 1/4 of the image's width).

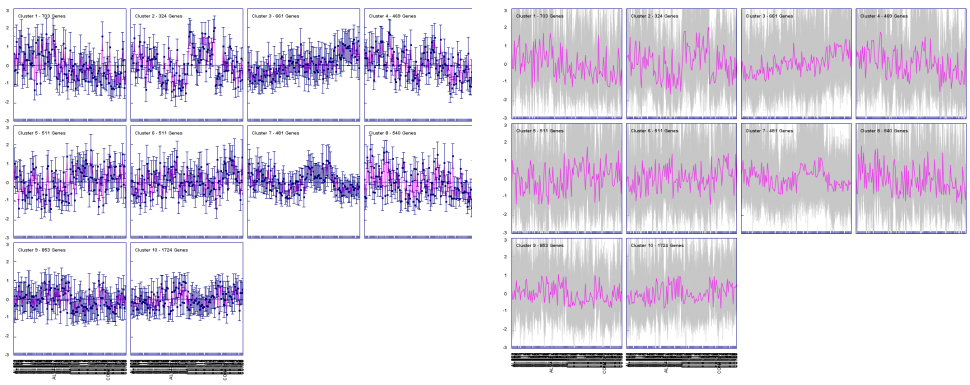


**Figure A.5:** Centroid view of the clusters (left) and their mean expression pattern (right) of applying the k-means algorithm with $k = 10$ and euclidean distance to cluster genes in ALS1 and AL2. Dataset order from left to right: ALS1, ALS2 (each one corresponds to 1/2 of the image's width).

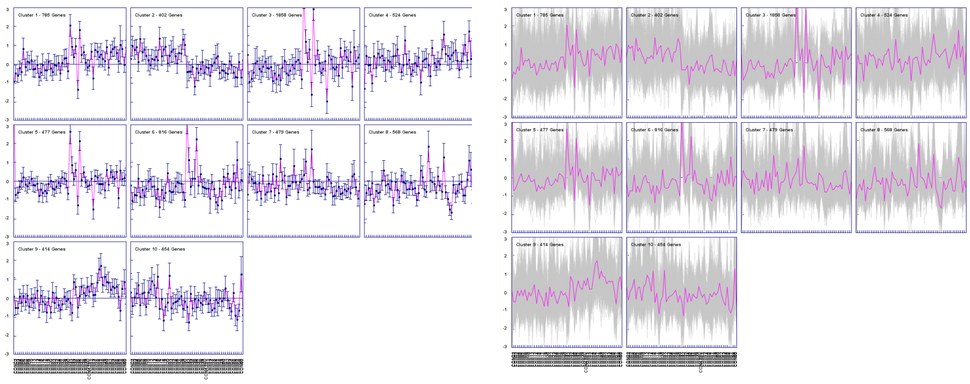


**Figure A.6:** Centroid view of the clusters (left) and their mean expression pattern (right) of applying the k-means algorithm with $k = 10$ and euclidean distance to cluster genes in C1 and C2. Dataset order from left to right: C1, C2 (each one corresponds to 1/2 of the image's width).

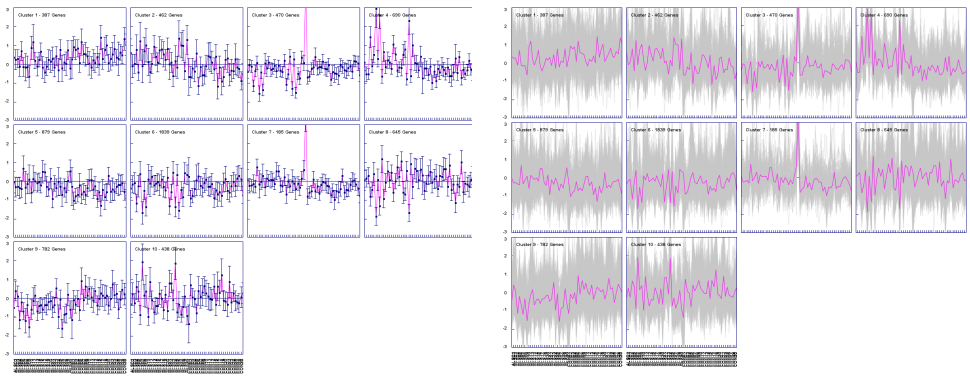


**Figure A.7:** Centroid view of the clusters (left) and their mean expression pattern (right) of applying the k-means algorithm with $k = 10$ and euclidean distance to cluster genes in ALS1 and C1. Dataset order from left to right: ALS1, C1 (each one corresponds to 1/2 of the image's width).

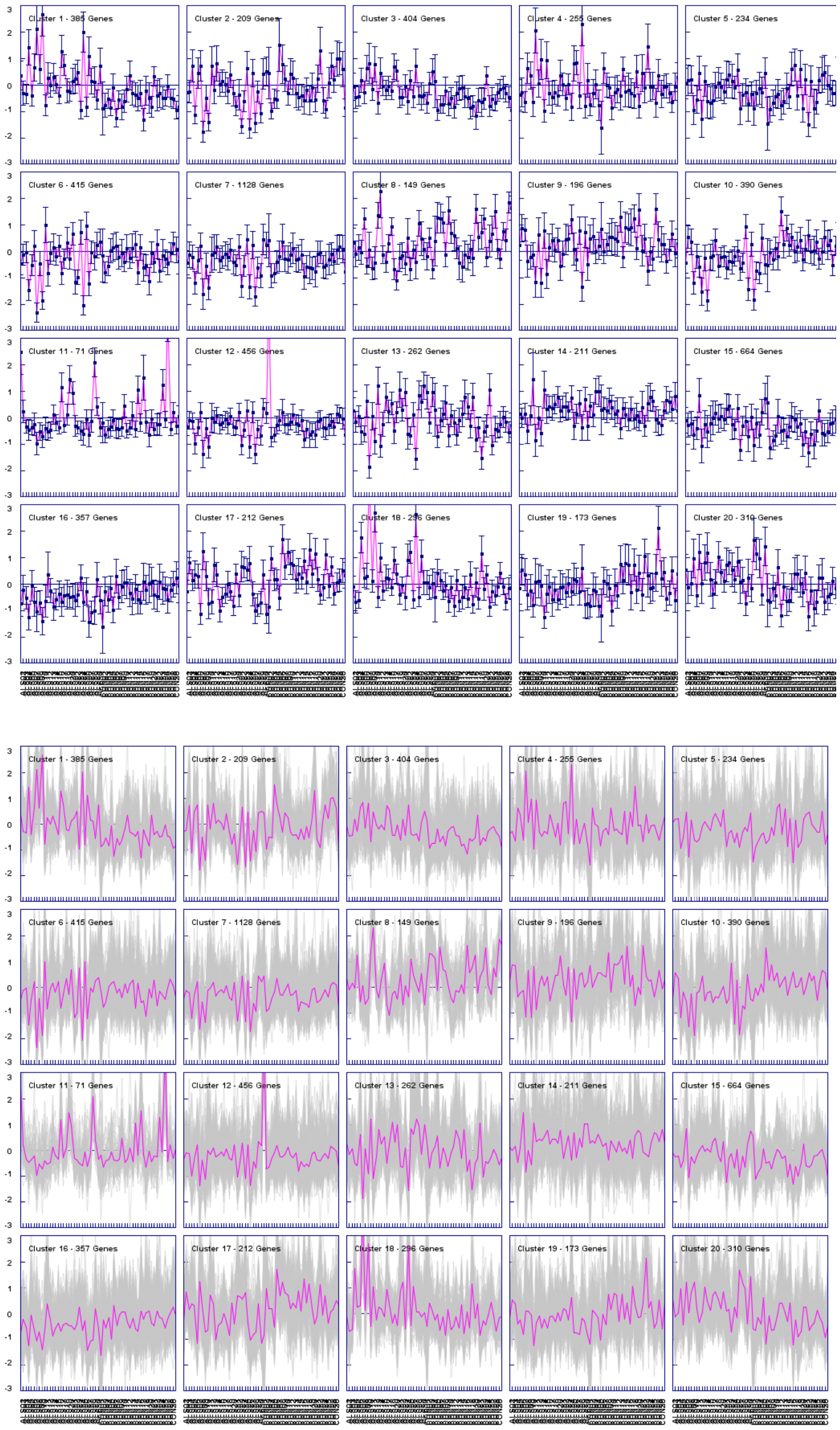


**Figure A.8:** Resulting clustering from applying the K-Means algorithm with $k = 20$ and using Euclidean distance to cluster genes in the first dataset (ALS1 with C1). Centroid view of the clusters (top) and their mean expression (bottom). Dataset order from left to right: ALS1, C1 (each one corresponds to 1/2 of the image's width).

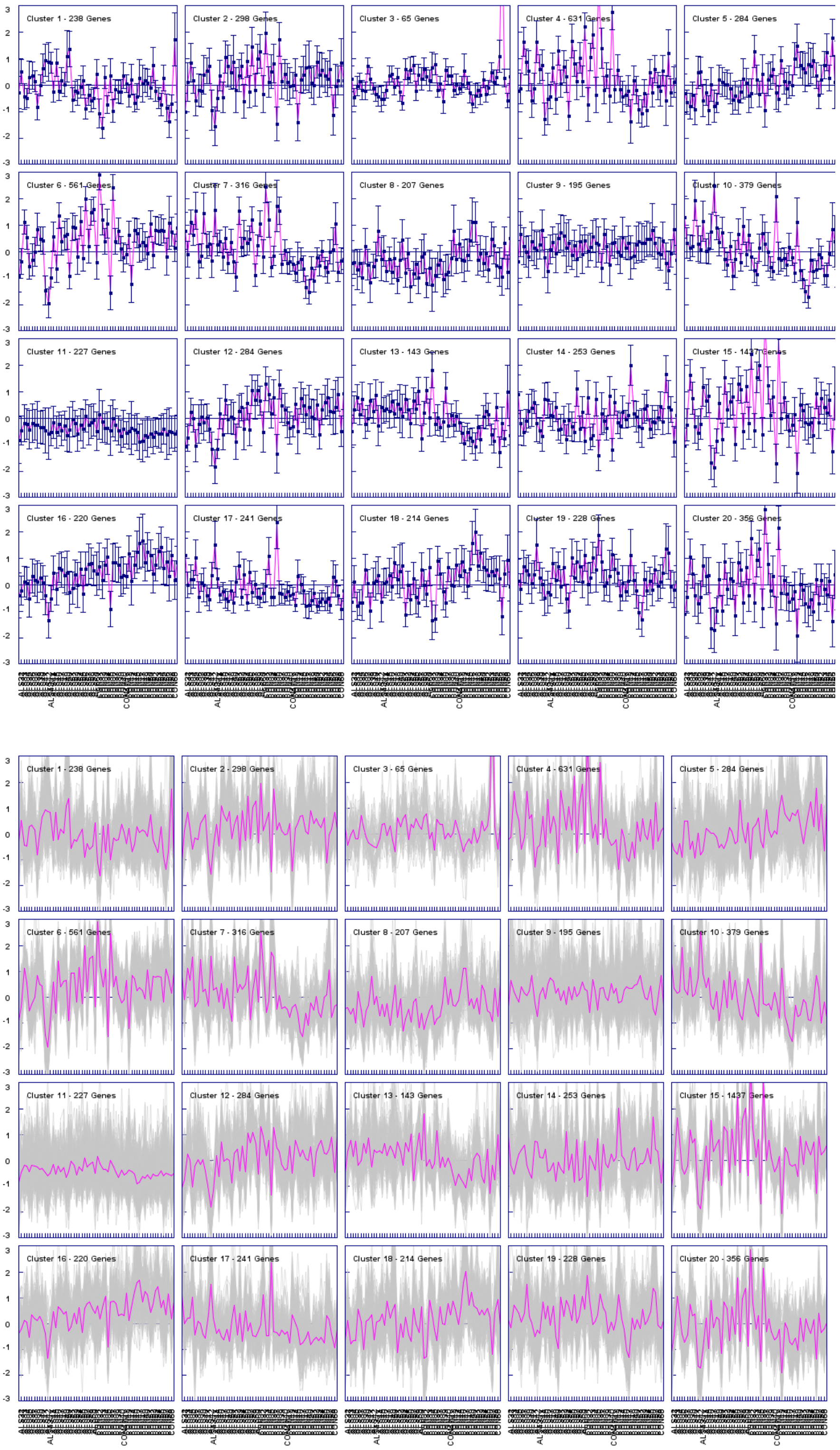


**Figure A.9:** Resulting clustering from applying the K-Means algorithm with $k = 20$ and using Euclidean distance to cluster genes in the second dataset (ALS2 with C2) after sample and gene normalization. Centroid view of the clusters (top) and their mean expression (bottom). Dataset order from left to right: ALS2, C2 (each one corresponds to 1/2 of the image's width).

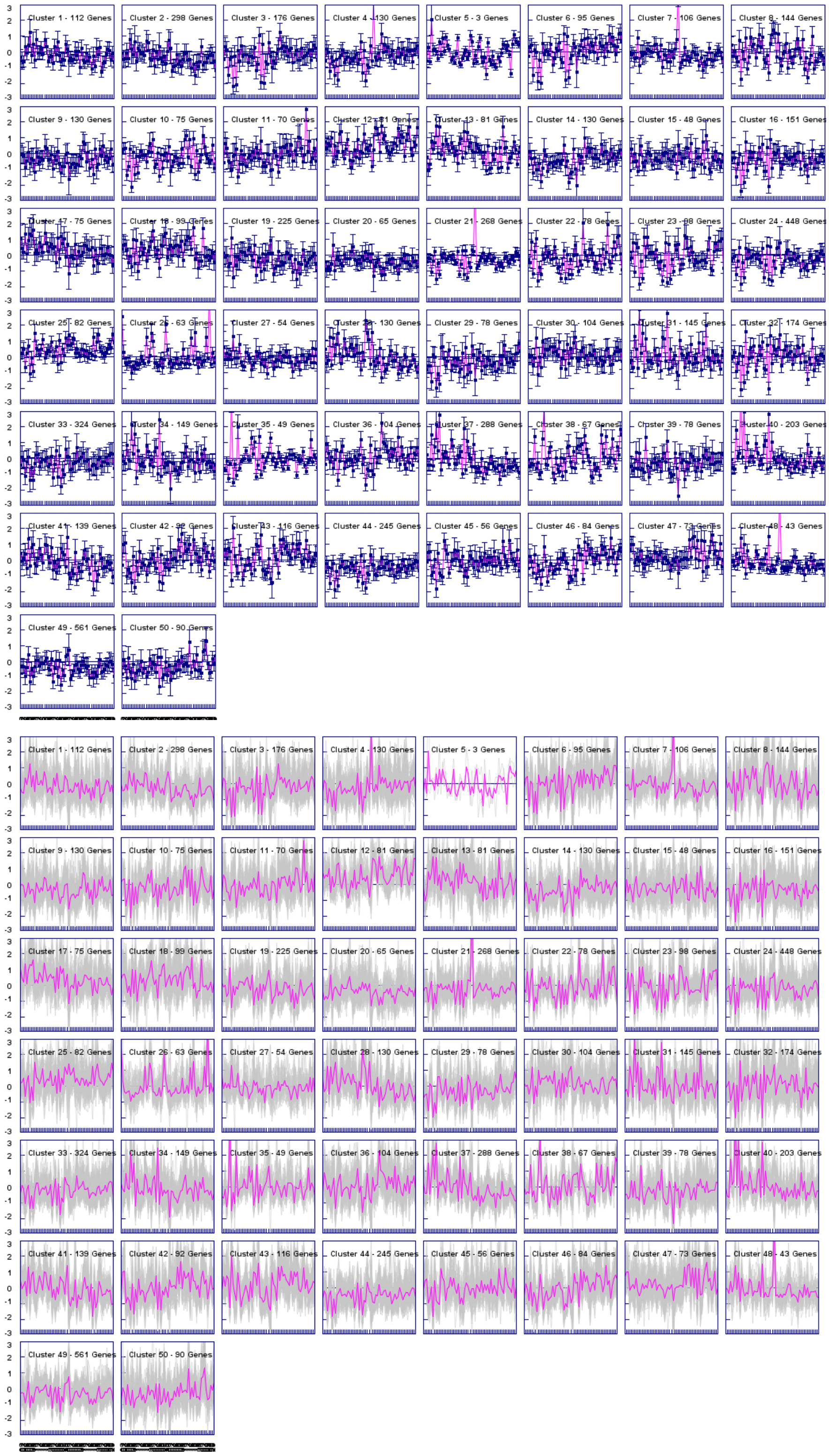


**Figure A.10:** Centroid view of the clusters (left) and their mean expression pattern (right) of applying the k-means algorithm with $k = 50$ and euclidean distance to cluster genes in ALS1 and C1. Dataset order from left to right: ALS1, C1 (each one corresponds to 1/2 of the image's width).

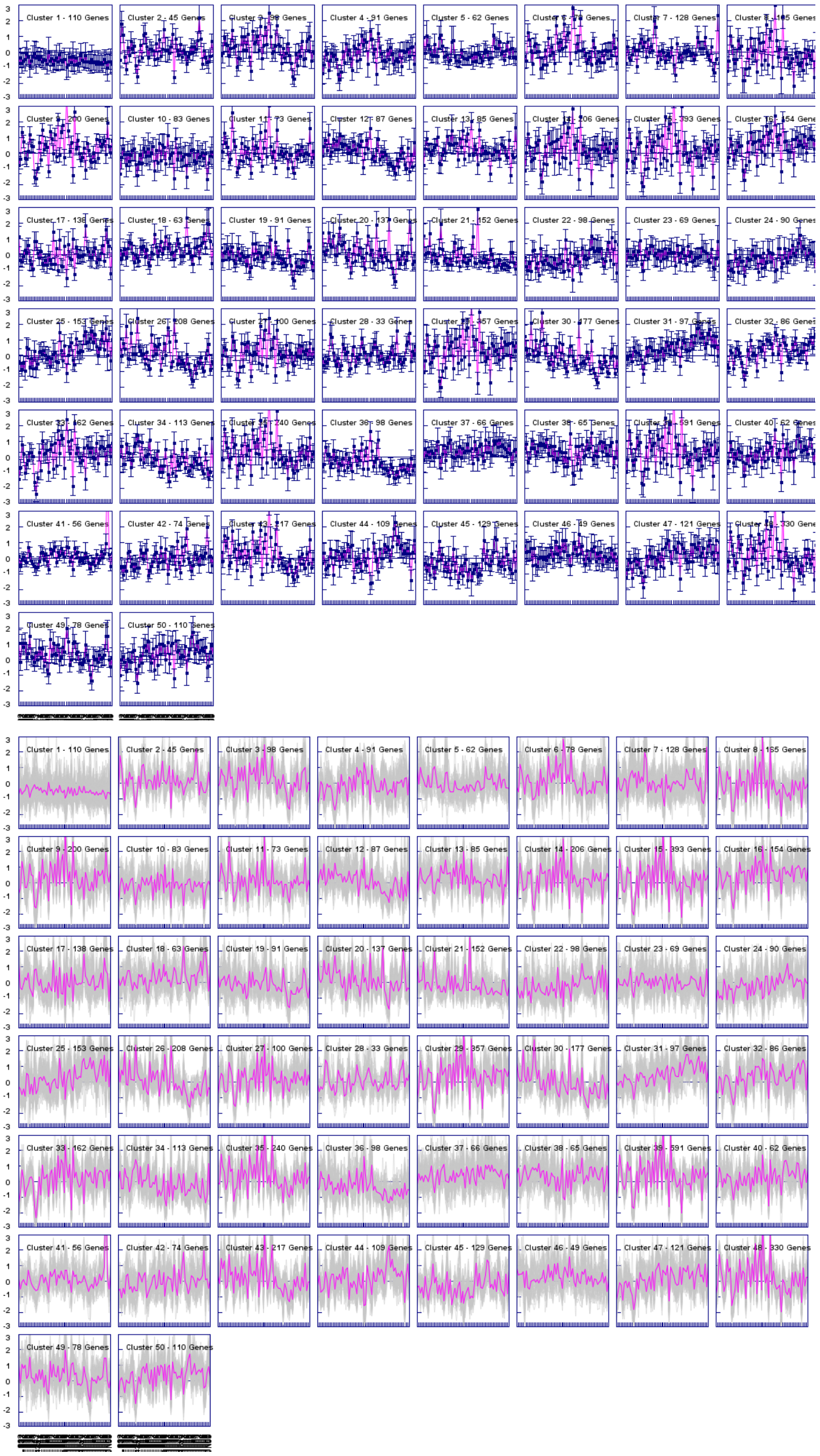


**Figure A.11:** Centroid view of the clusters (top) and their mean expression pattern (bottom) of applying the k-means algorithm with $k = 50$ and euclidean distance to cluster genes in ALS2 and C2. Dataset order from left to right: ALS2, C2 (each one corresponds to 1/2 of the image's width).

# A.2 WGCNA

## A.2.1 Soft Power

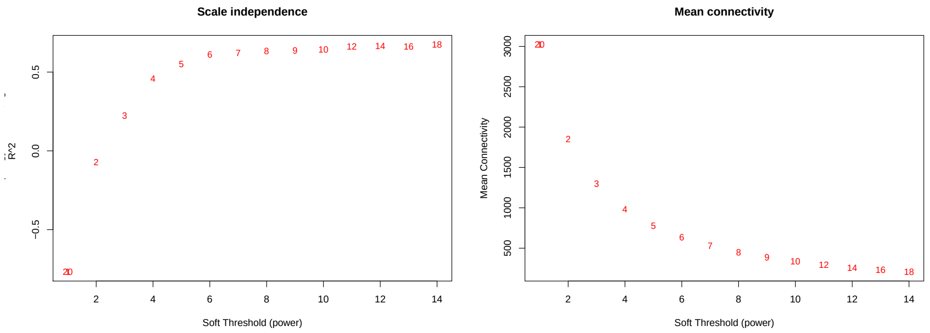


**Figure A.12:** Fitting of the network to the scale-free topology accordingly to the beta parameter used in the soft-thresholding procedure for dataset ALS2. Dataset order from left to right: ALS2, C2 (each one corresponds to 1/2 of the image's width).

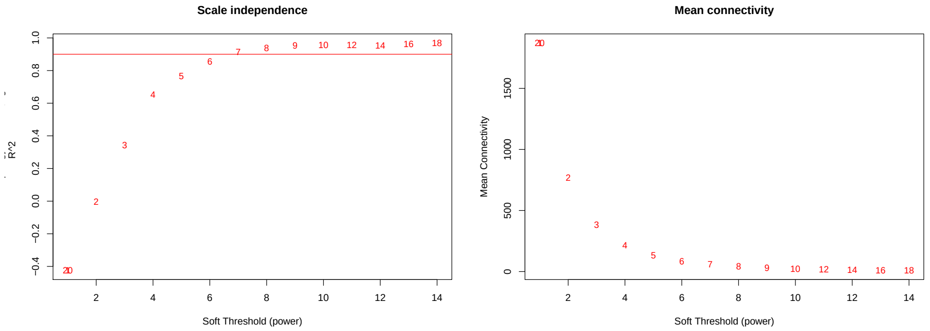


**Figure A.13:** Fitting of the network to the scale-free topology accordingly to the beta parameter used in the soft-thresholding procedure for dataset C1.

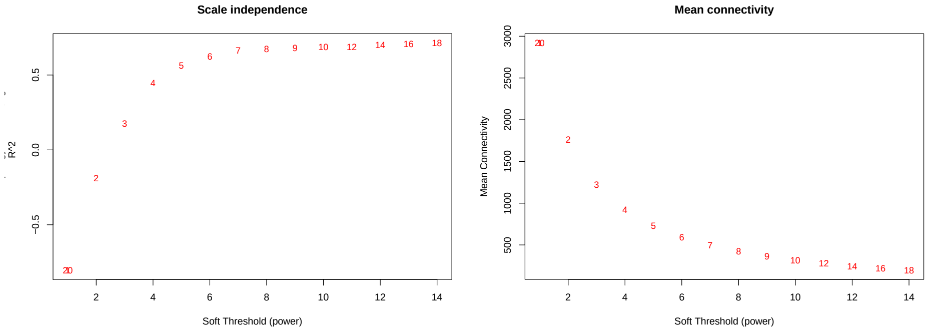


**Figure A.14:** Fitting of the network to the scale-free topology accordingly to the beta parameter used in the soft-thresholding procedure for dataset C2.

### A.2.2 Branch Cutting Parameter Variation

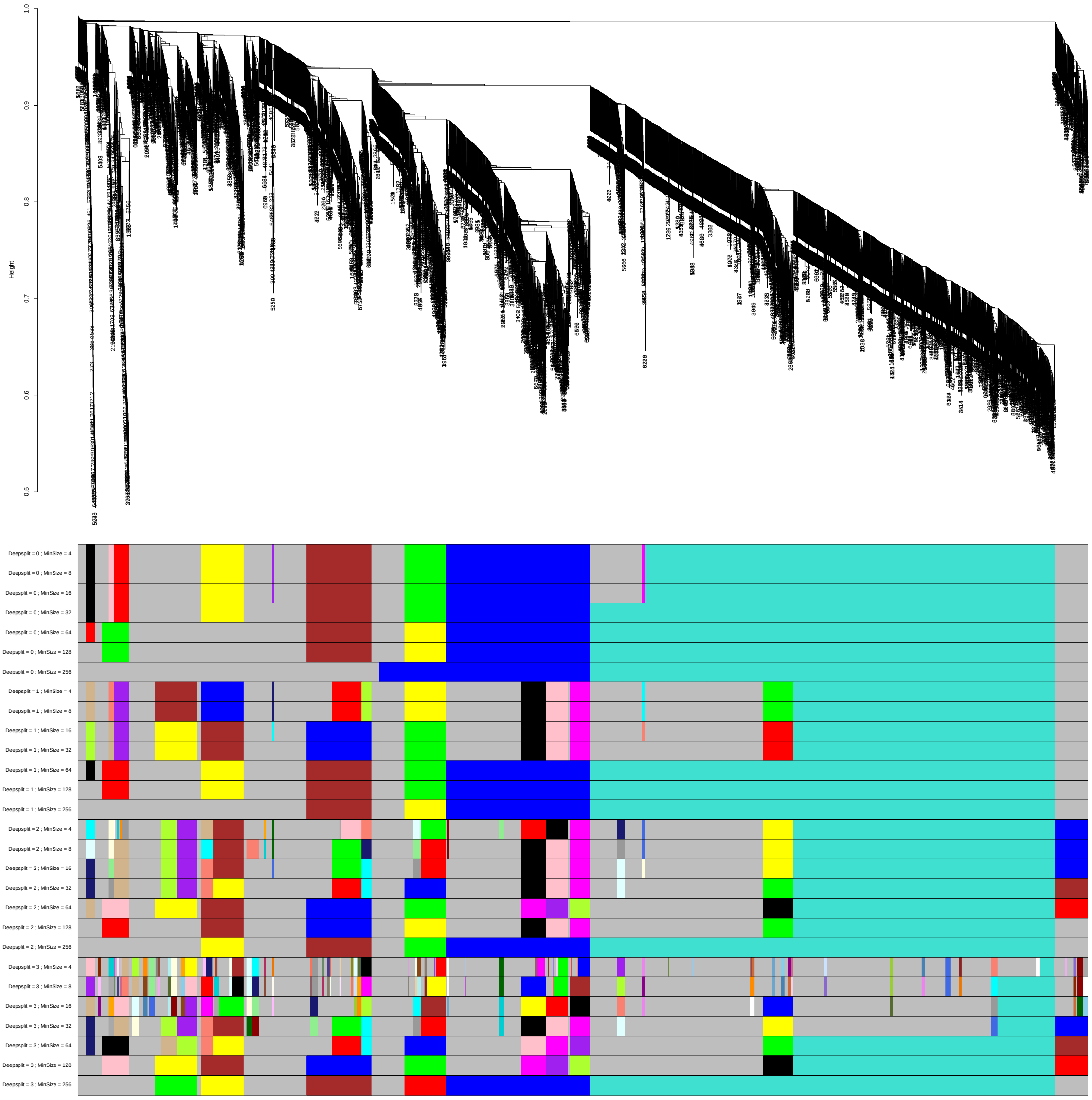


**Figure A.15:** Dendrogram resulting from applying WGCNA with a soft-threshold of 6 to dataset ALS1. Different cuts of this hierarchical tree are presented by variation of two parameters of the *cutreeHybrid()*: *minimum cluster size* (4, 8, 16, 32, 64, 128 and 256) and *deepSplit* (0,1,2,3).

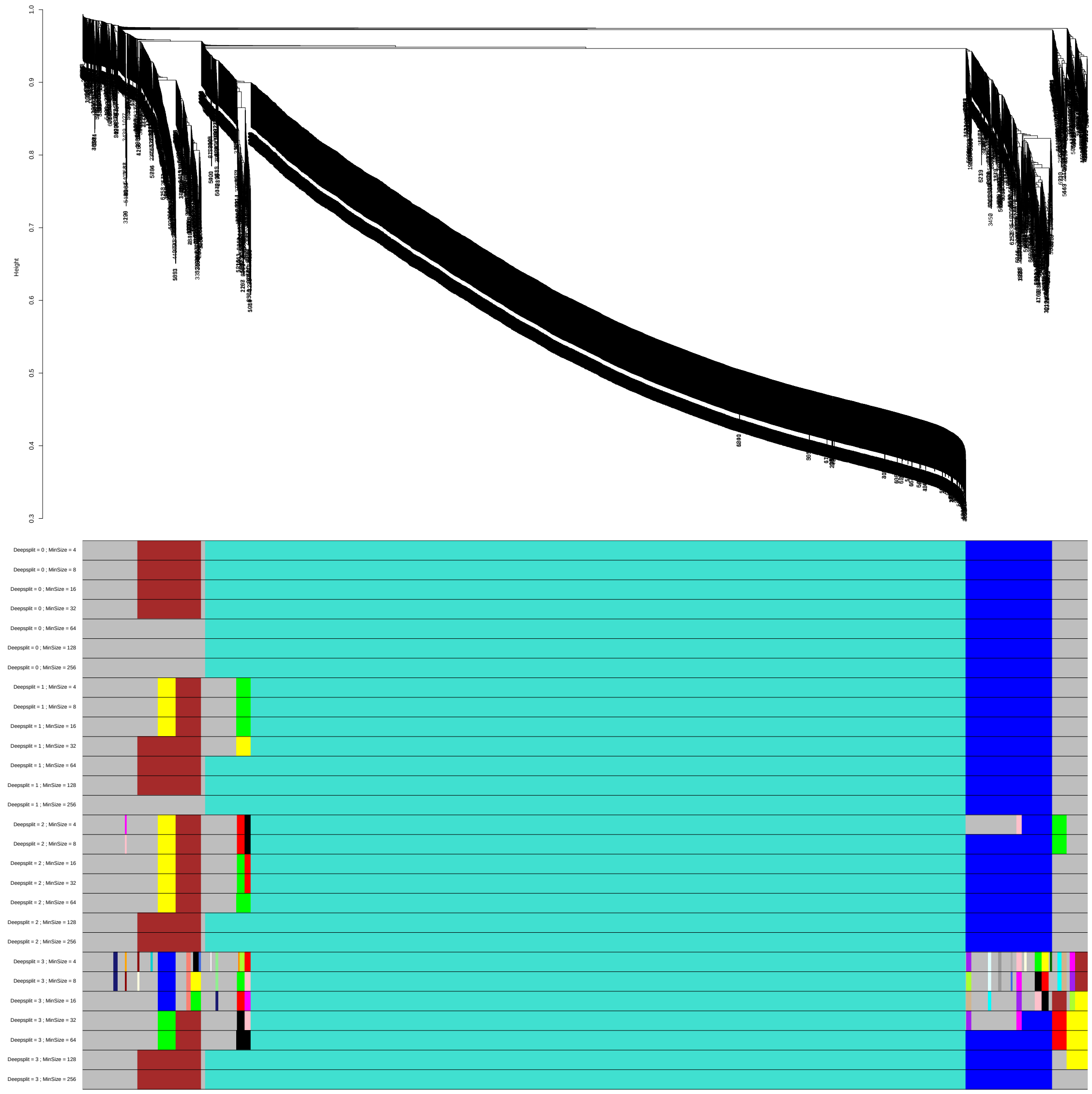


**Figure A.16:** Dendrogram resulting from applying WGCNA with a soft-threshold of 6 to dataset ALS2. Different cuts of this hierarchical tree are presented by variation of two parameters of the *cutreeHybrid()*: *minimum cluster size* (4, 8, 16, 32, 64, 128 and 256) and *deepSplit* (0,1,2,3).

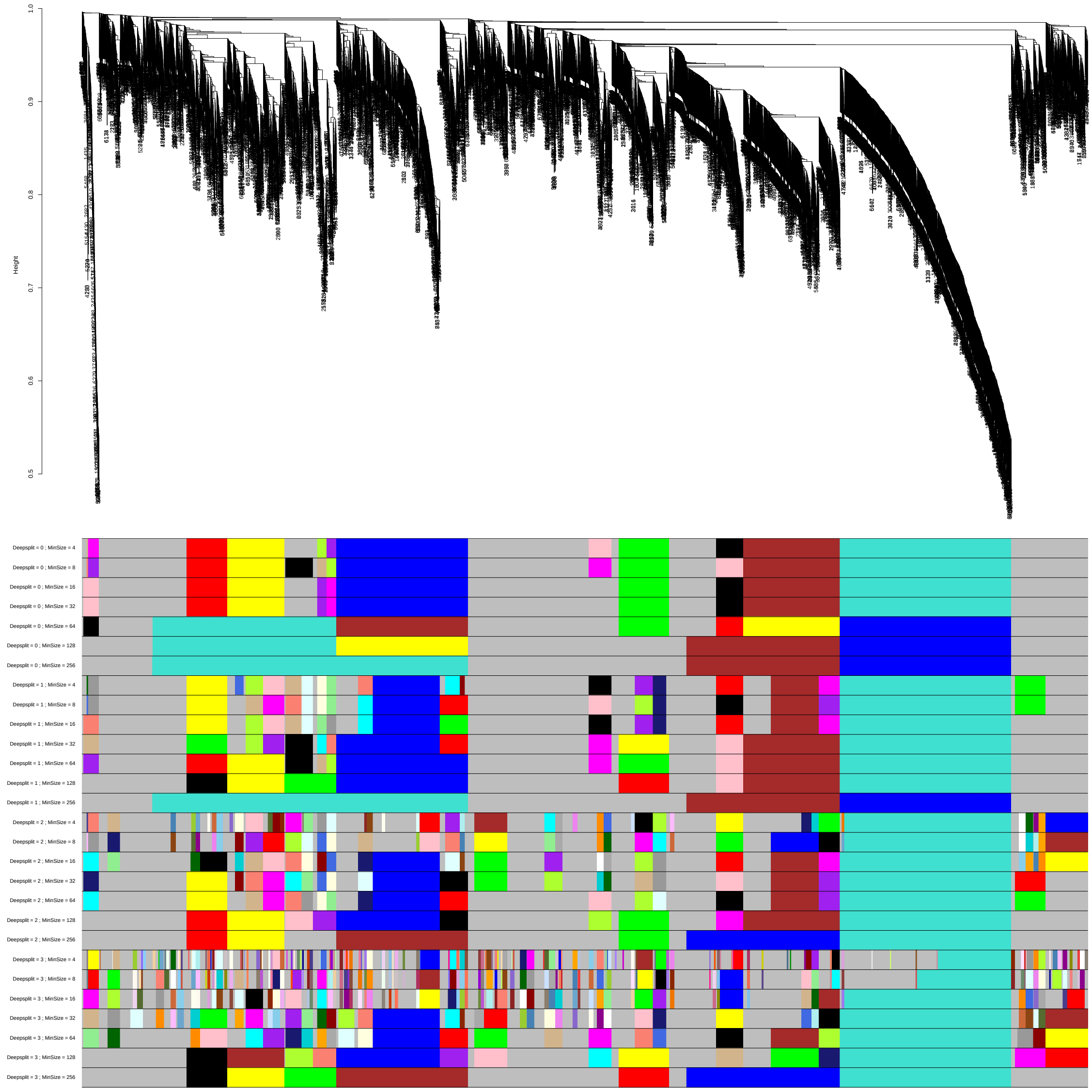


**Figure A.17:** Dendrogram resulting from applying WGCNA with a soft-threshold of 6 to dataset ALS1. Different cuts of this hierarchical tree are presented by variation of two parameters of the *cutreeHybrid()*: *minimum cluster size* (4, 8, 16, 32, 64, 128 and 256) and *deepSplit* (0,1,2,3).

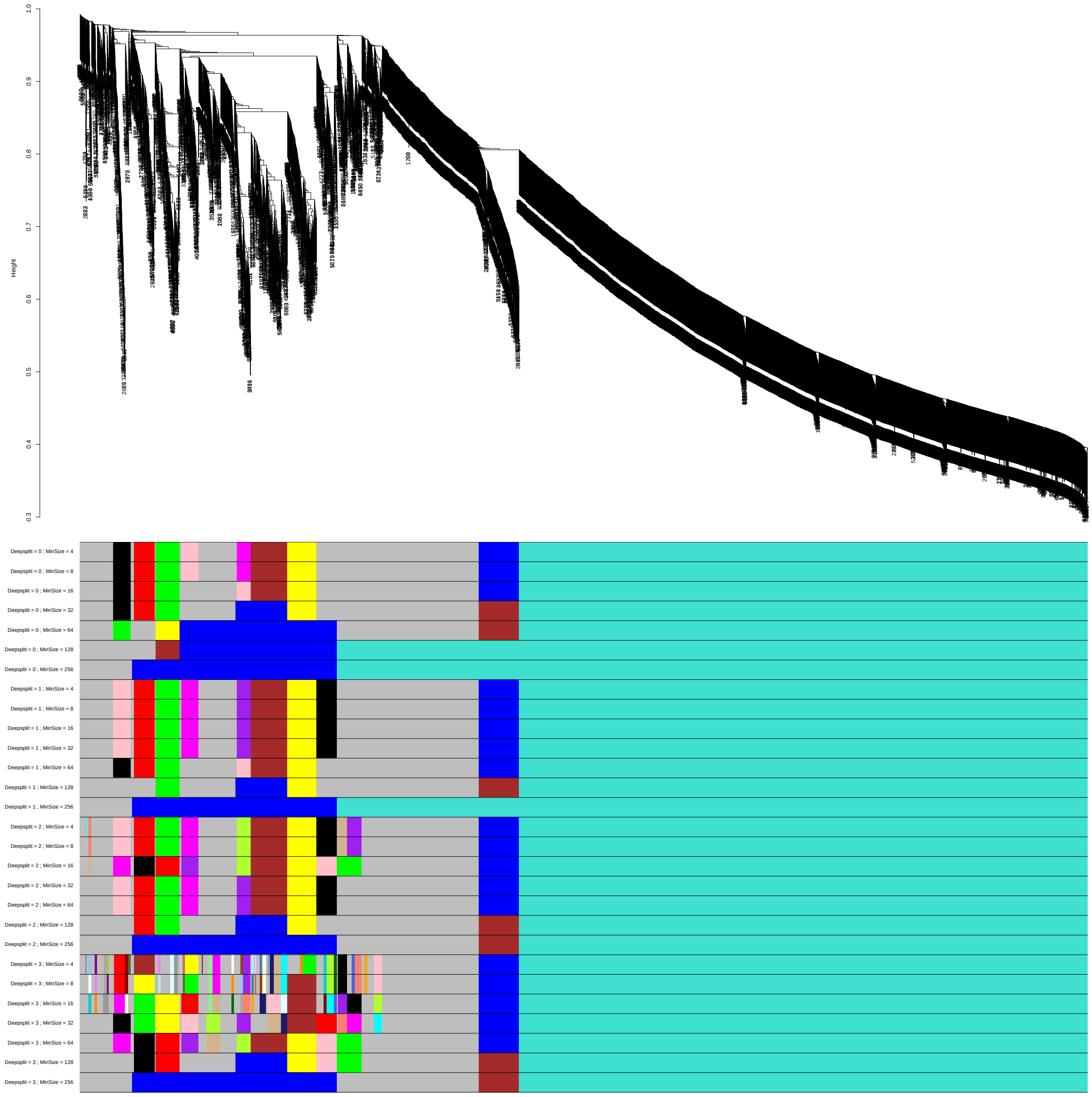
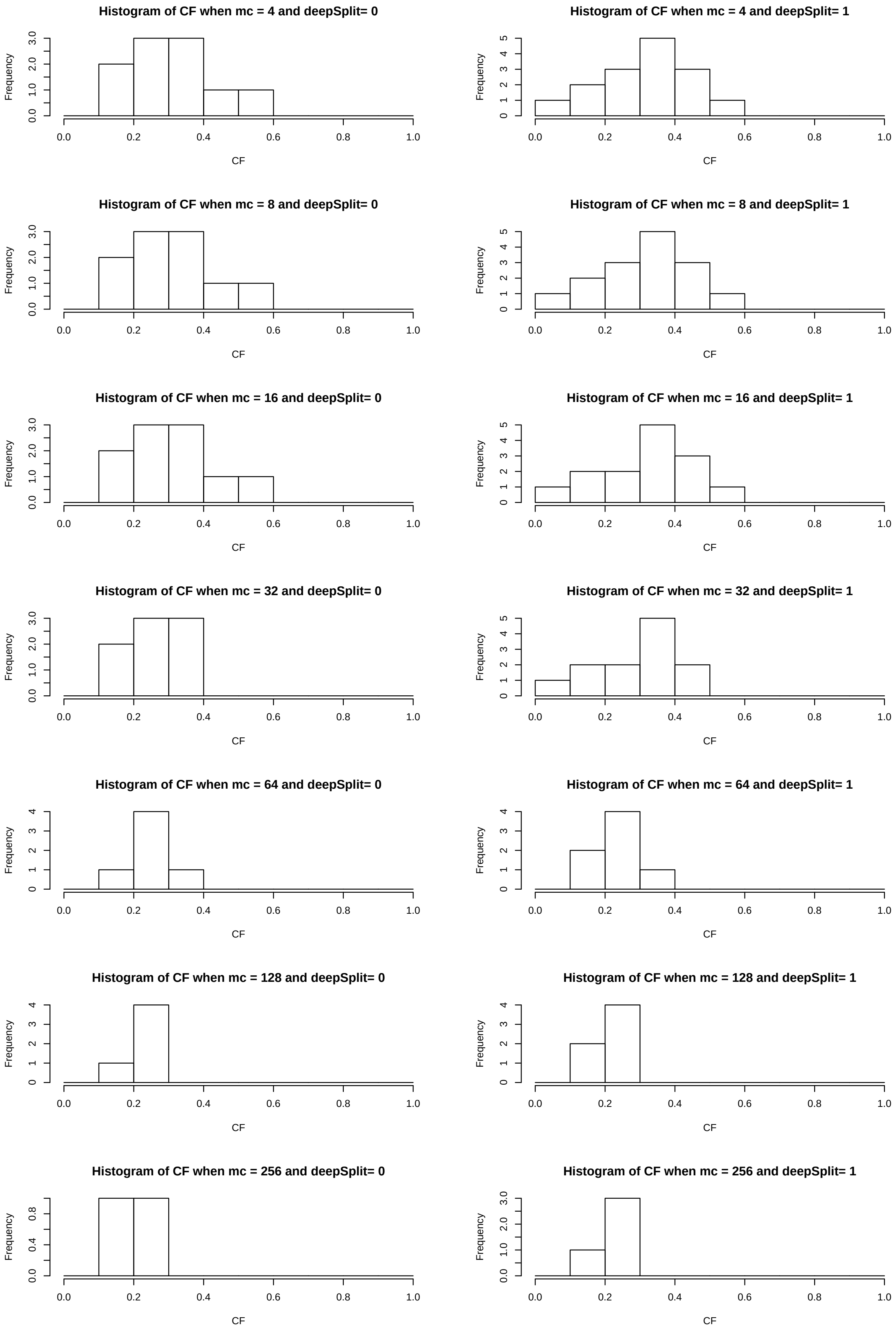


**Figure A.18:** Dendrogram resulting from applying WGCNA with a soft-threshold of 6 to dataset ALS1. Different cuts of this hierarchical tree are presented by variation of two parameters of the *cutreeHybrid()*: *minimum cluster size* (4, 8, 16, 32, 64, 128 and 256) and *deepSplit* (0,1,2,3).

## A.2.3 Network Concepts

### A.2.3.A Clustering Coefficient Histograms



Histogram of CF when mc = 4 and deepSplit= 0

Histogram of CF when mc = 4 and deepSplit= 1

Histogram of CF when mc = 8 and deepSplit= 0

Histogram of CF when mc = 8 and deepSplit= 1

Histogram of CF when mc = 16 and deepSplit= 0

Histogram of CF when mc = 16 and deepSplit= 1

Histogram of CF when mc = 32 and deepSplit= 0

Histogram of CF when mc = 32 and deepSplit= 1

Histogram of CF when mc = 64 and deepSplit= 0

Histogram of CF when mc = 64 and deepSplit= 1

Histogram of CF when mc = 128 and deepSplit= 0

Histogram of CF when mc = 128 and deepSplit= 1

Histogram of CF when mc = 256 and deepSplit= 0

Histogram of CF when mc = 256 and deepSplit= 1

**Figure A.19:** Clustering coefficient histogram for each cluster set for ALS1, varying the *minimum cluster size* in brach cutting algorithm in 4, 8, 16, 32, 64, 128 and 256 and *deepSplit* equals to 0 (left) and 1 (right).

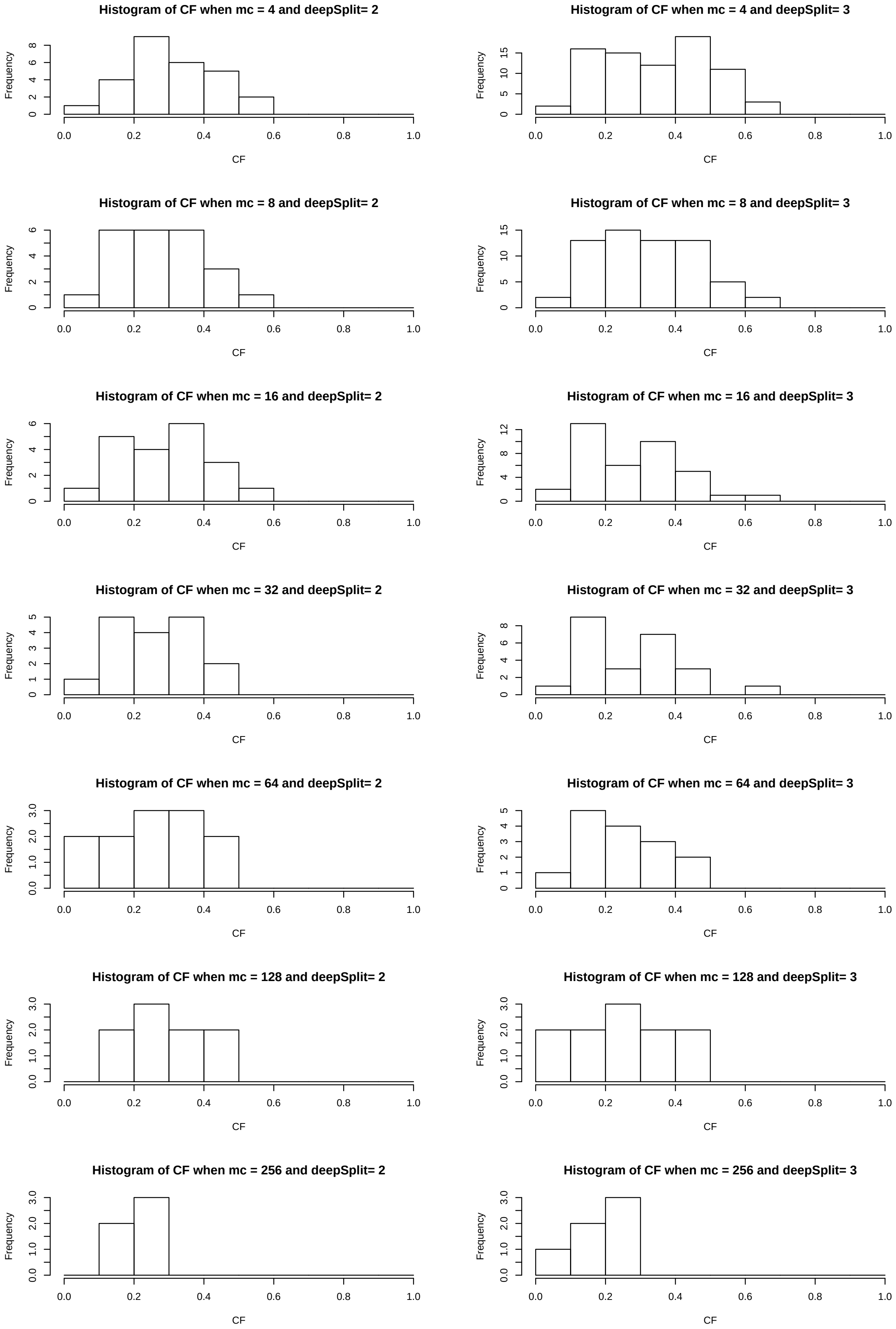
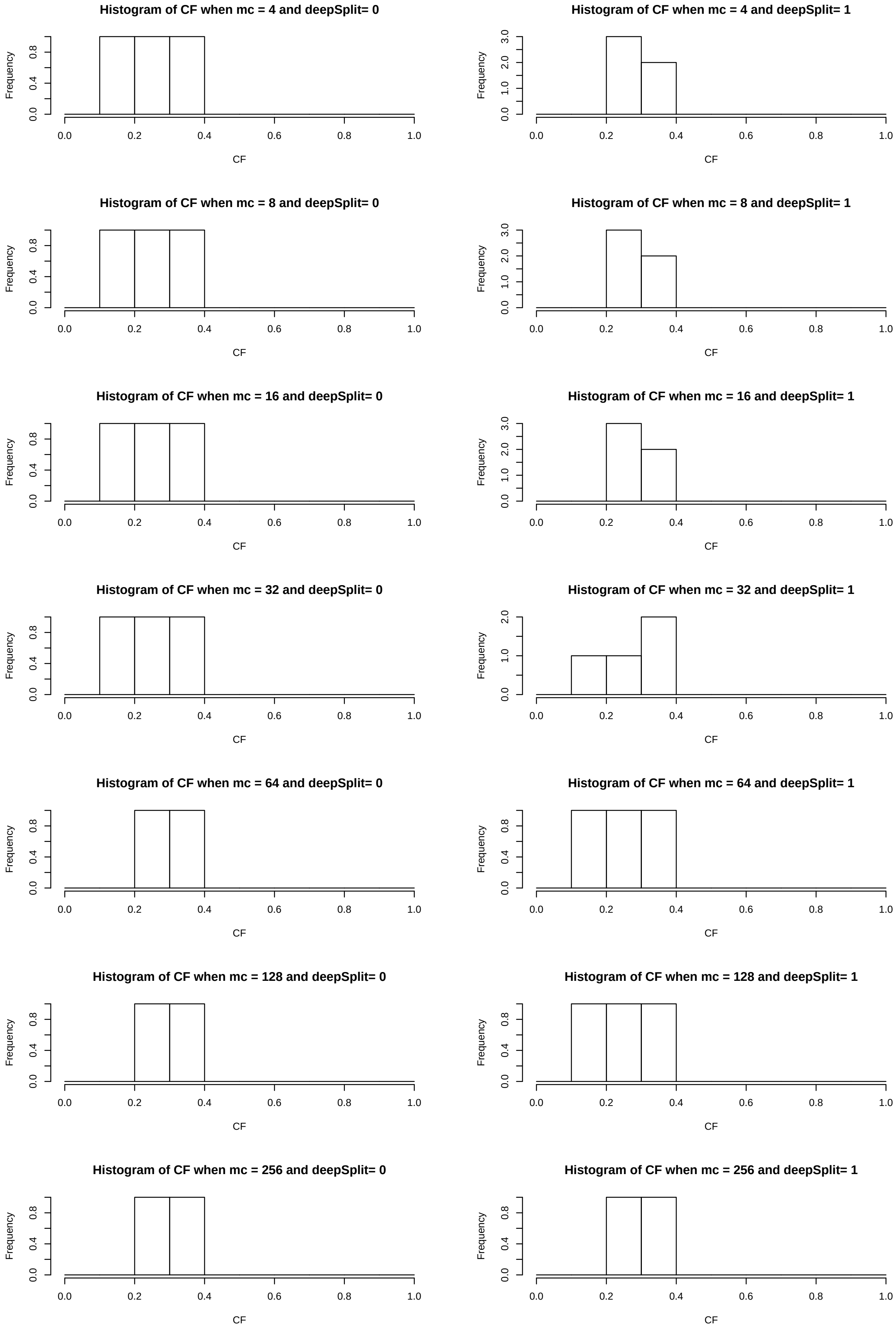


**Figure A.20:** Clustering coefficient histogram for each cluster set for ALS1, varying the *minimum cluster size* in brach cutting algorithm in 4, 8, 16, 32, 64, 128 and 256 and *deepSplit* equals to 2 (left) and 3 (right).


Histogram of CF when mc = 4 and deepSplit= 0

Histogram of CF when mc = 4 and deepSplit= 1

Histogram of CF when mc = 8 and deepSplit= 0

Histogram of CF when mc = 8 and deepSplit= 1

Histogram of CF when mc = 16 and deepSplit= 0

Histogram of CF when mc = 16 and deepSplit= 1

Histogram of CF when mc = 32 and deepSplit= 0

Histogram of CF when mc = 32 and deepSplit= 1

Histogram of CF when mc = 64 and deepSplit= 0

Histogram of CF when mc = 64 and deepSplit= 1

Histogram of CF when mc = 128 and deepSplit= 0

Histogram of CF when mc = 128 and deepSplit= 1

Histogram of CF when mc = 256 and deepSplit= 0

Histogram of CF when mc = 256 and deepSplit= 1

**Figure A.21:** Clustering coefficient histogram for each cluster set for ALS2, varying the *minimum cluster size* in brach cutting algorithm in 4, 8, 16, 32, 64, 128 and 256 and *deepSplit* equals to 0 (left) and 1 (right).

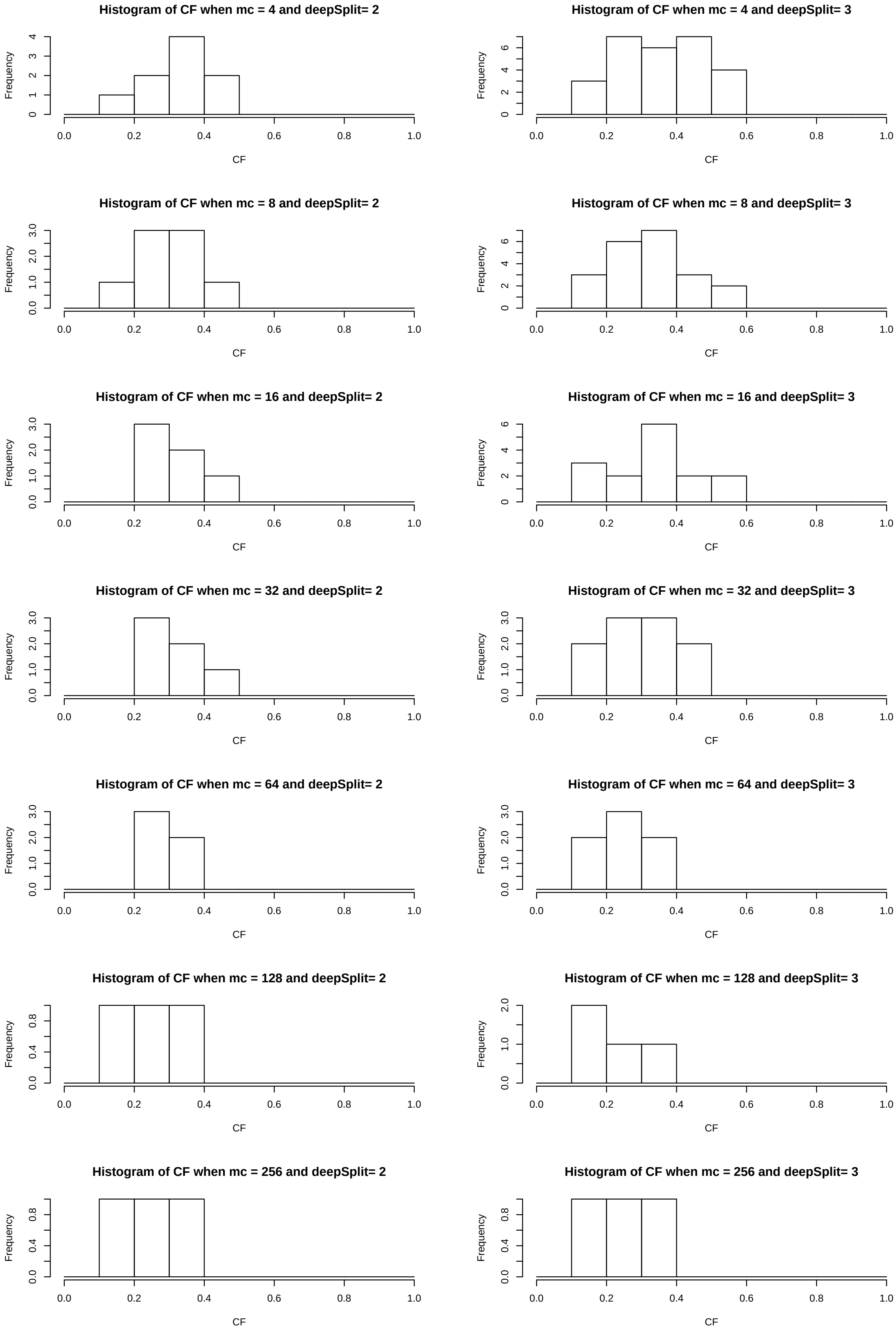
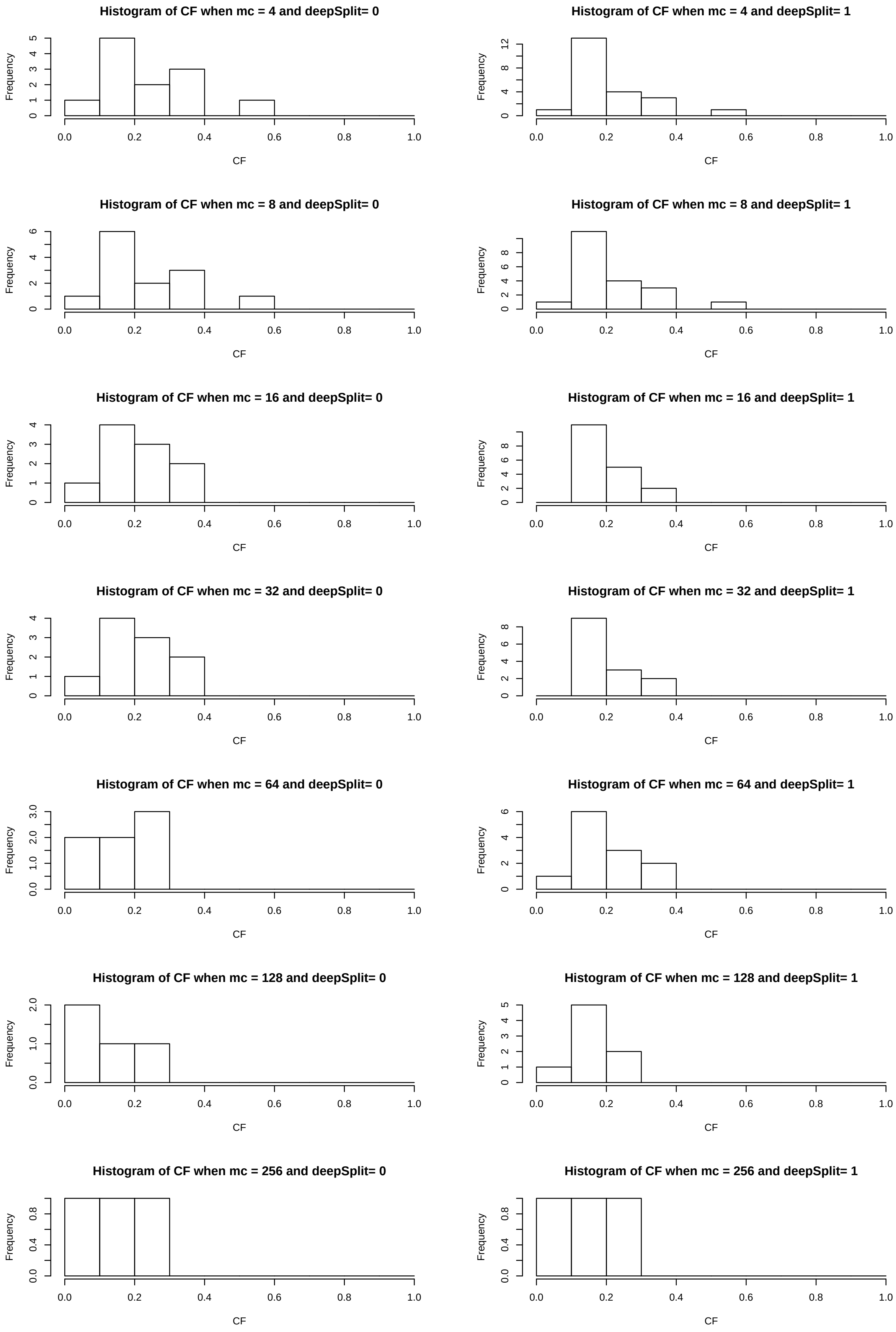


**Figure A.22:** Clustering coefficient histogram for each cluster set for ALS2, varying the *minimum cluster size* in brach cutting algorithm in 4, 8, 16, 32, 64, 128 and 256 and *deepSplit* equals to 2 (left) and 3 (right).



Histogram of CF when mc = 4 and deepSplit= 0

Histogram of CF when mc = 4 and deepSplit= 1

Histogram of CF when mc = 8 and deepSplit= 0

Histogram of CF when mc = 8 and deepSplit= 1

Histogram of CF when mc = 16 and deepSplit= 0

Histogram of CF when mc = 16 and deepSplit= 1

Histogram of CF when mc = 32 and deepSplit= 0

Histogram of CF when mc = 32 and deepSplit= 1

Histogram of CF when mc = 64 and deepSplit= 0

Histogram of CF when mc = 64 and deepSplit= 1

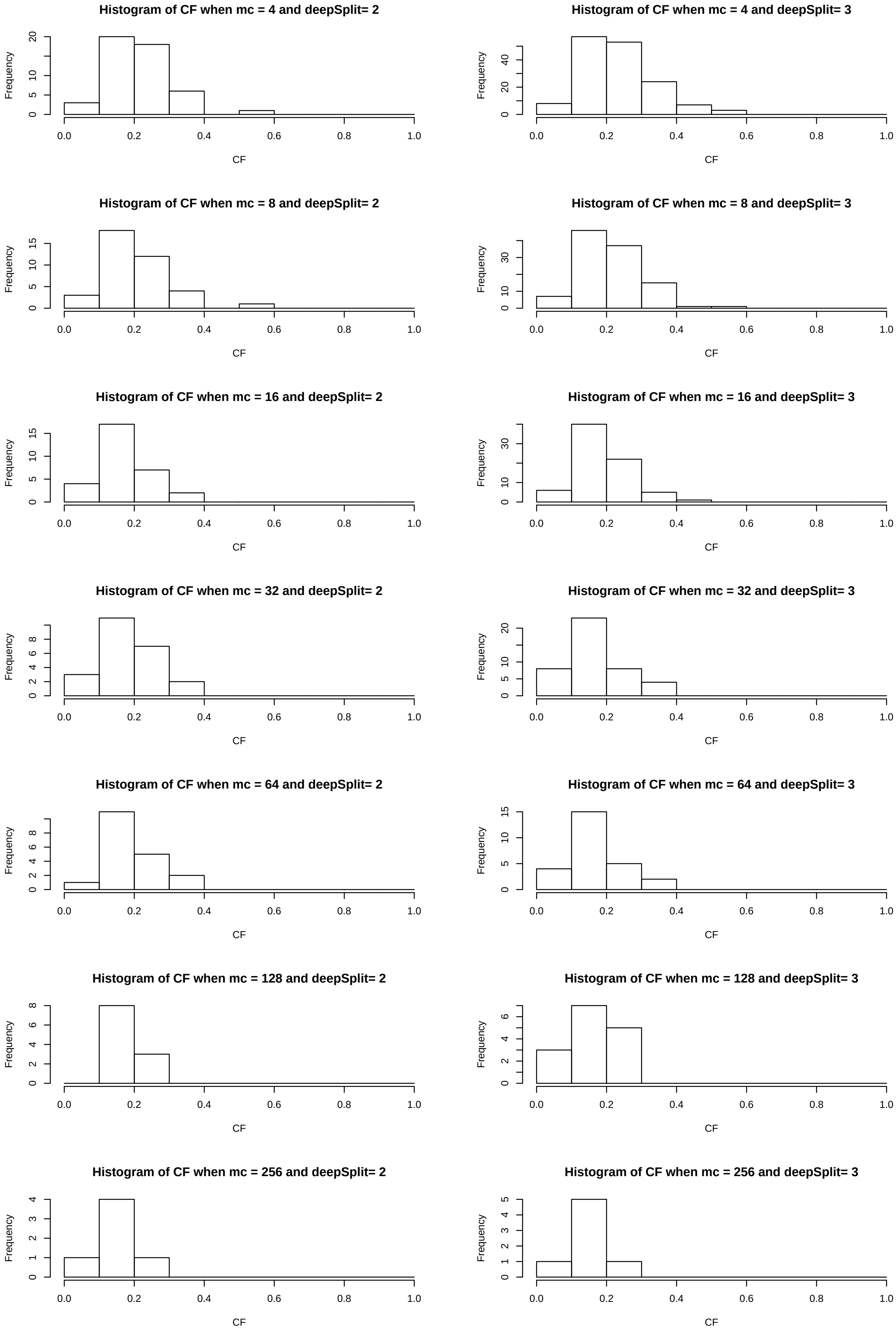
Histogram of CF when mc = 128 and deepSplit= 0

Histogram of CF when mc = 128 and deepSplit= 1

Histogram of CF when mc = 256 and deepSplit= 0

Histogram of CF when mc = 256 and deepSplit= 1

**Figure A.23:** Clustering coefficient histogram for each cluster set for C1, varying the *minimum cluster size* in brach cutting algorithm in 4, 8, 16, 32, 64, 128 and 256 and *deepSplit* equals to 0 (left) and 1 (right).


Histogram of CF when mc = 4 and deepSplit= 2

Histogram of CF when mc = 4 and deepSplit= 3

Histogram of CF when mc = 8 and deepSplit= 2

Histogram of CF when mc = 8 and deepSplit= 3

Histogram of CF when mc = 16 and deepSplit= 2

Histogram of CF when mc = 16 and deepSplit= 3

Histogram of CF when mc = 32 and deepSplit= 2

Histogram of CF when mc = 32 and deepSplit= 3

Histogram of CF when mc = 64 and deepSplit= 2

Histogram of CF when mc = 64 and deepSplit= 3

Histogram of CF when mc = 128 and deepSplit= 2

Histogram of CF when mc = 128 and deepSplit= 3

Histogram of CF when mc = 256 and deepSplit= 2

Histogram of CF when mc = 256 and deepSplit= 3

**Figure A.24:** Clustering coefficient histogram for each cluster set for C1, varying the *minimum cluster size* in brach cutting algorithm in 4, 8, 16, 32, 64, 128 and 256 and *deepSplit* equals to 2 (left) and 3 (right).

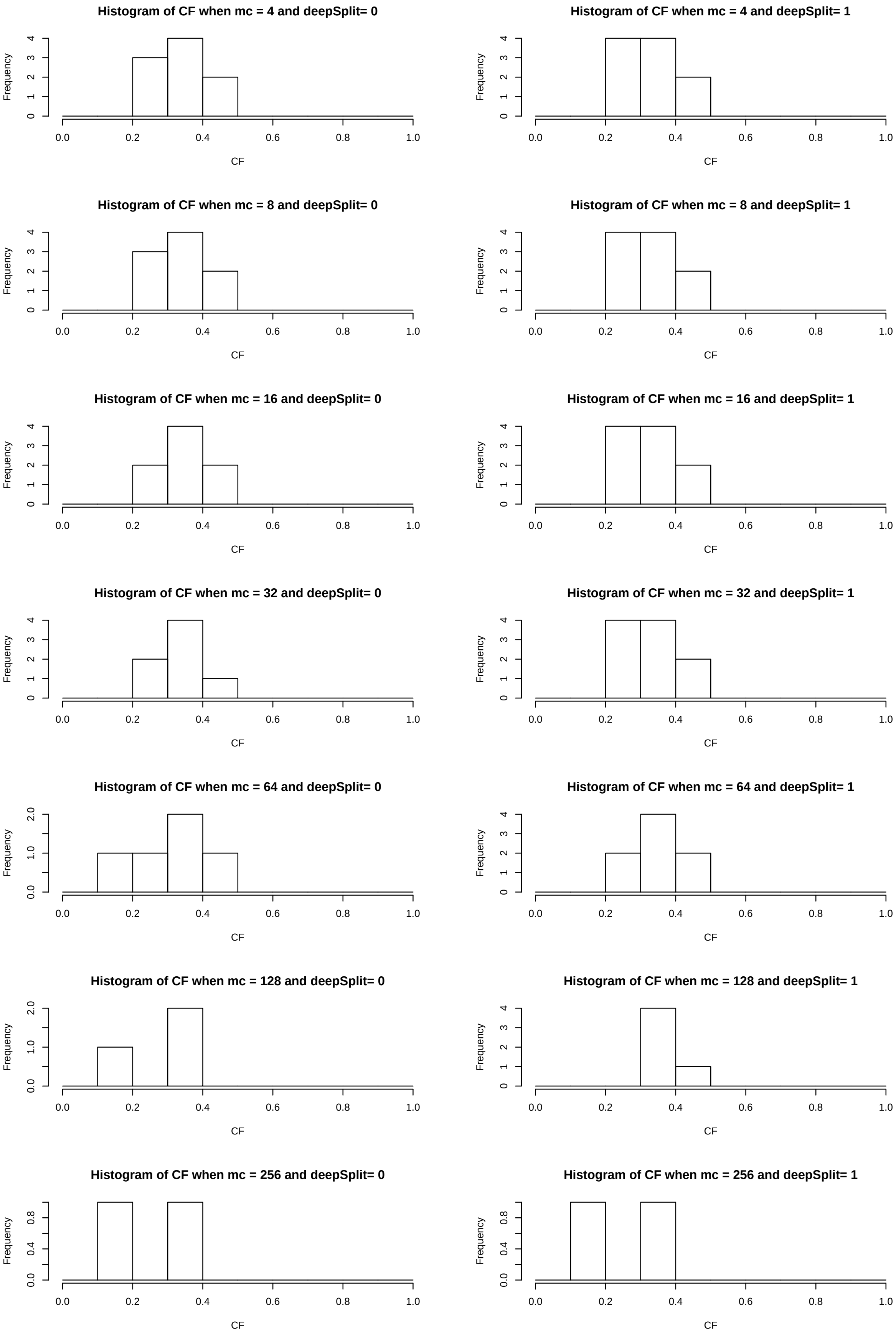
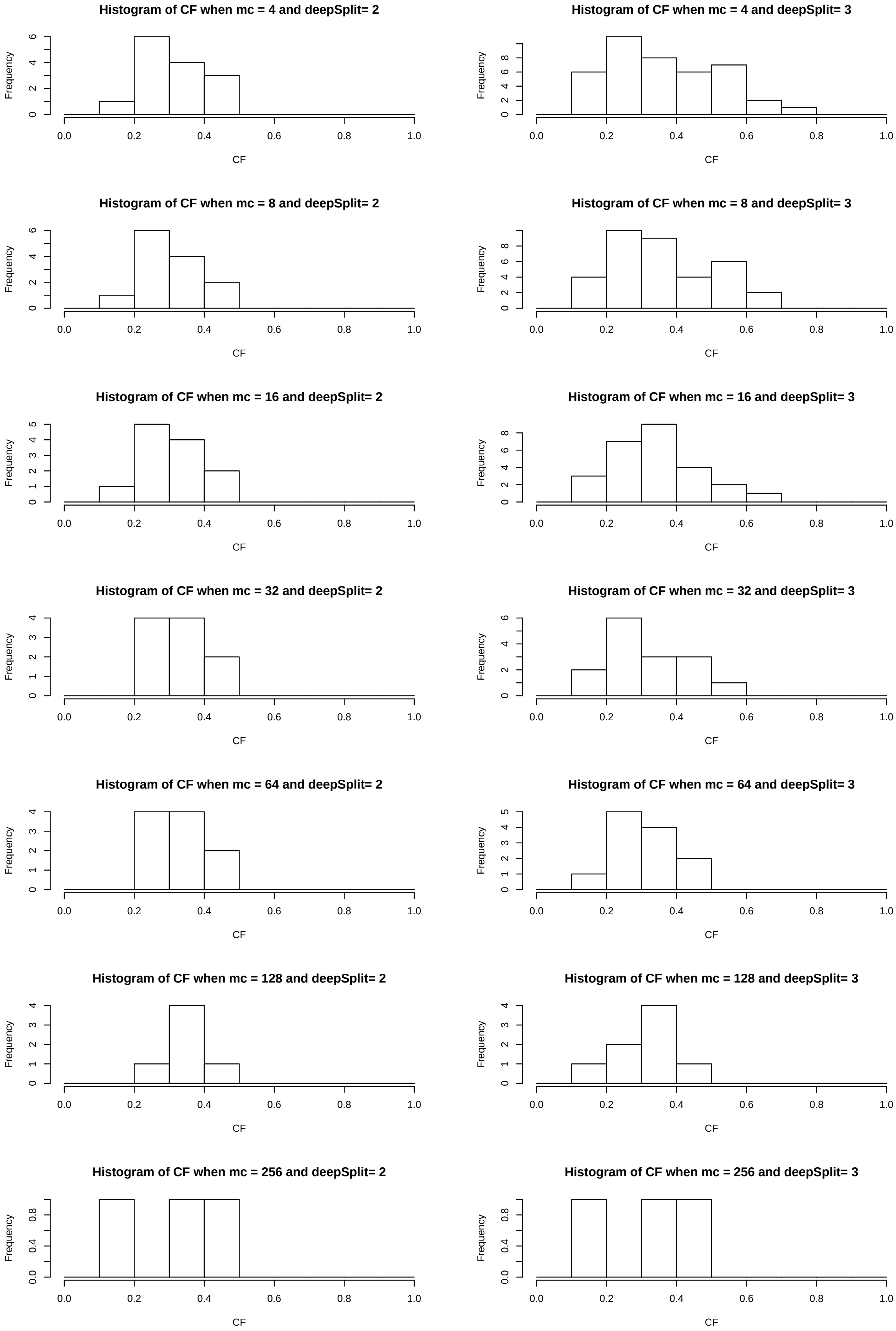


**Figure A.25:** Clustering coefficient histogram for each cluster set for C2, varying the *minimum cluster size* in brach cutting algorithm in 4, 8, 16, 32, 64, 128 and 256 and *deepSplit* equals to 0 (left) and 1 (right).



Histogram of CF when mc = 4 and deepSplit= 2

Histogram of CF when mc = 4 and deepSplit= 3

Histogram of CF when mc = 8 and deepSplit= 2

Histogram of CF when mc = 8 and deepSplit= 3

Histogram of CF when mc = 16 and deepSplit= 2

Histogram of CF when mc = 16 and deepSplit= 3

Histogram of CF when mc = 32 and deepSplit= 2

Histogram of CF when mc = 32 and deepSplit= 3

Histogram of CF when mc = 64 and deepSplit= 2

Histogram of CF when mc = 64 and deepSplit= 3

Histogram of CF when mc = 128 and deepSplit= 2

Histogram of CF when mc = 128 and deepSplit= 3

Histogram of CF when mc = 256 and deepSplit= 2

Histogram of CF when mc = 256 and deepSplit= 3

**Figure A.26:** Clustering coefficient histogram for each cluster set for C2, varying the *minimum cluster size* in brach cutting algorithm in 4, 8, 16, 32, 64, 128 and 256 and *deepSplit* equals to 2 (left) and 3 (right).

### A.2.4 Network Based Statistics to study WGCNA Module Preservation

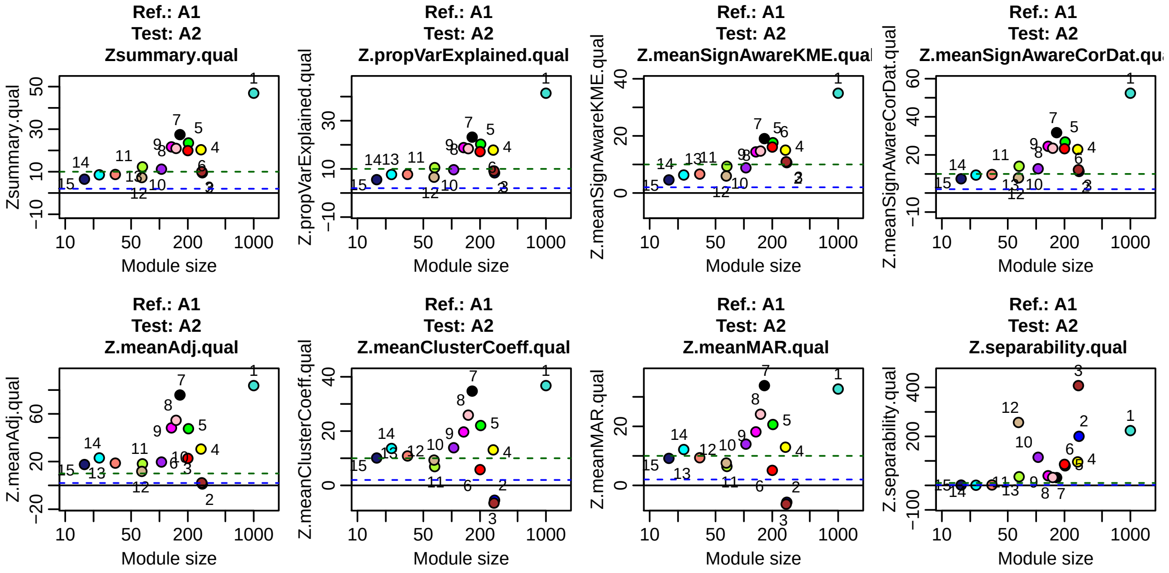


**Figure A.27:** Network based statistics summaries accordingly to Langfelder et. al [62] using as reference network dataset ALS1 and as testing set ALS2. The measure $Z_{summary}$ is the one translating the general behaviour of these network statistics. The green line in each subplot defines the empirical determined significance threshold (=10) obtained in [62] and above each a module is considered preserved. The blue line is the lower limit of significance (=2), below this line it is considered that there is no enough evidence of that module being preserved on both networks.

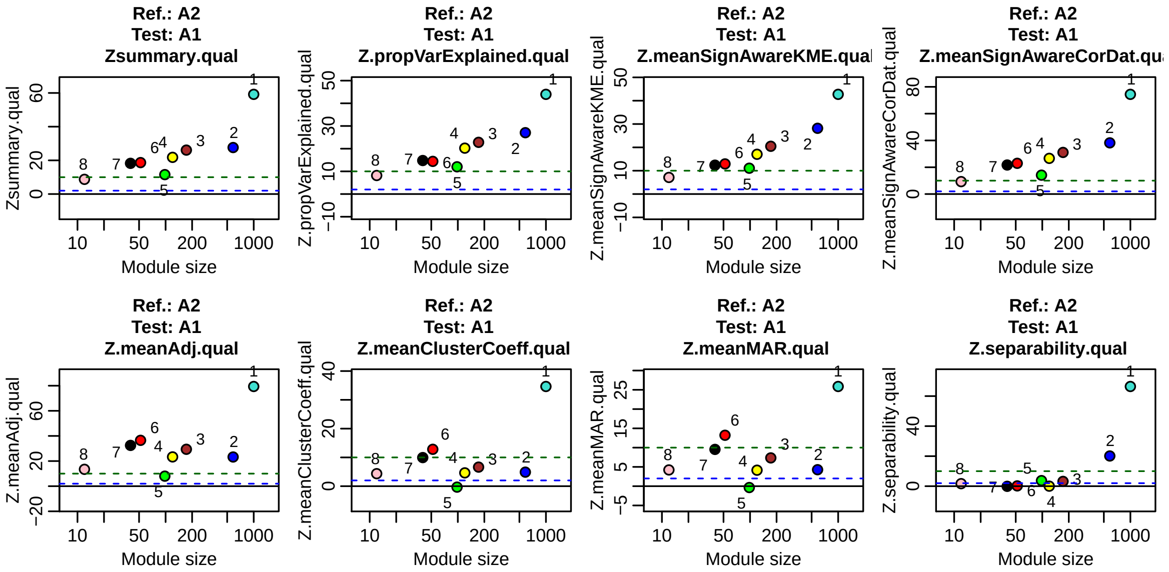


**Figure A.28:** Network based statistics summaries accordingly to Langfelder et. al [62] using as reference network dataset ALS2 and as testing set ALS1. The measure $Z_{summary}$ is the one translating the general behaviour of these network statistics. The green line in each subplot defines the empirical determined significance threshold (=10) obtained in [62] and above each a module is considered preserved. The blue line is the lower limit of significance (=2), below this line it is considered that there is no enough evidence of that module being preserved on both networks.

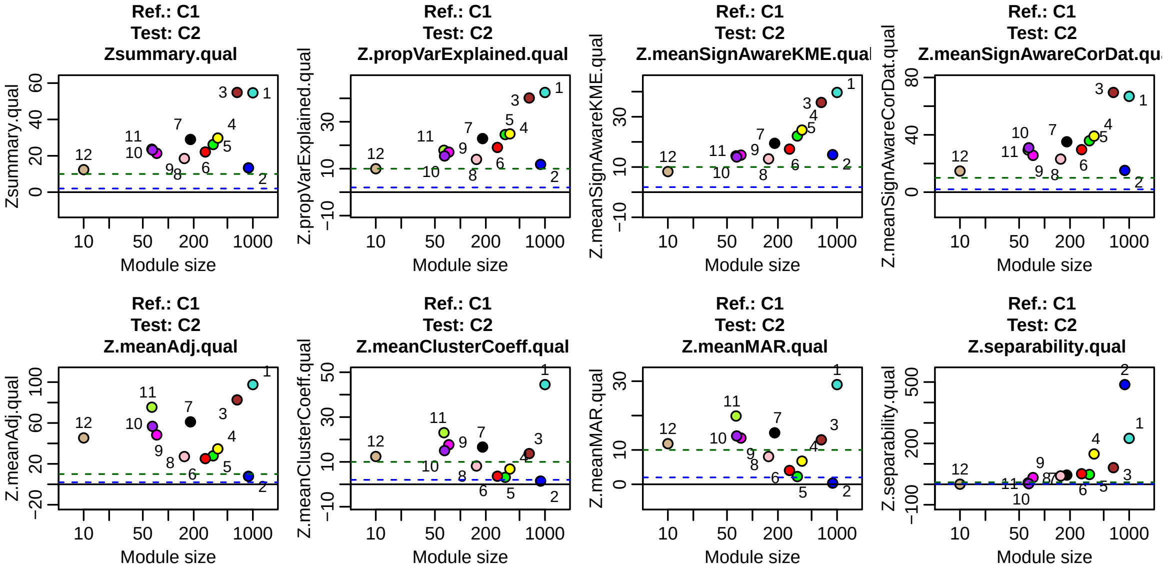


**Figure A.29:** Network based statistics summaries accordingly to Langfelder et. al [62] using as reference network dataset C1 and as testing set C2. The measure $Z_{summary}$ is the one translating the general behaviour of these network statistics. The green line in each subplot defines the empirical determined significance threshold (=10) obtained in [62] and above each a module is considered preserved. The blue line is the lower limit of significance (=2), below this line it is considered that there is no enough evidence of that module being preserved on both networks.

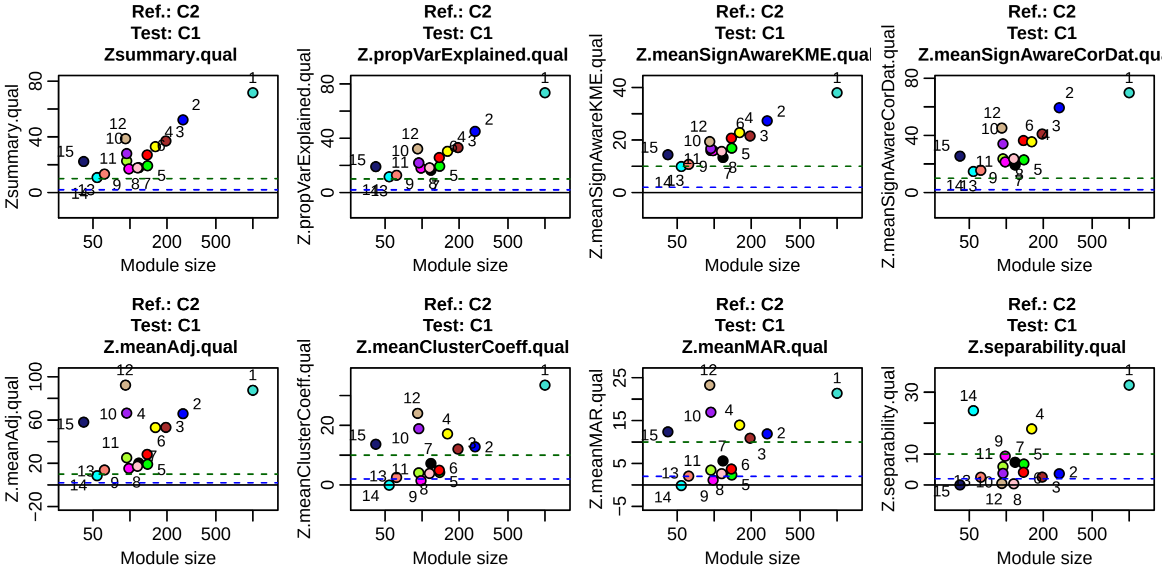


**Figure A.30:** Network based statistics summaries accordingly to Langfelder et. al [62] using as reference network dataset C2 and as testing set C1. The measure $Z_{summary}$ is the one translating the general behaviour of these network statistics. The green line in each subplot defines the empirical determined significance threshold (=10) obtained in [62] and above each a module is considered preserved. The blue line is the lower limit of significance (=2), below this line it is considered that there is no enough evidence of that module being preserved on both networks.

## A.3 Functional annotation of clusters

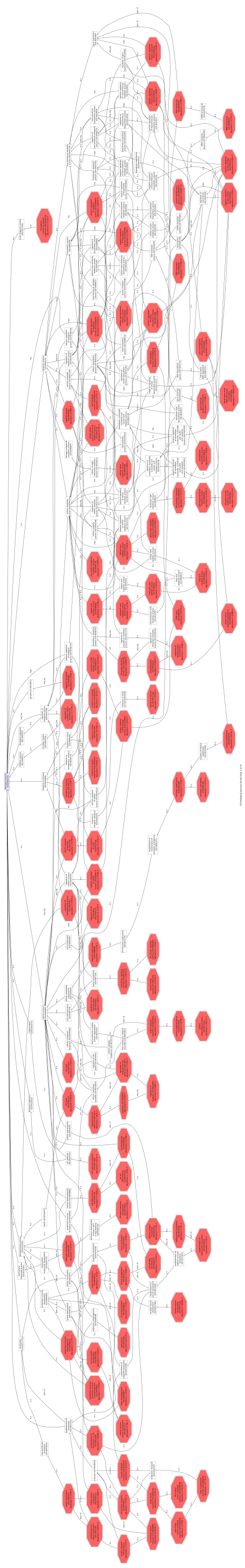


**Figure A.31:** GO terms of the WGCNA Turquoise module.

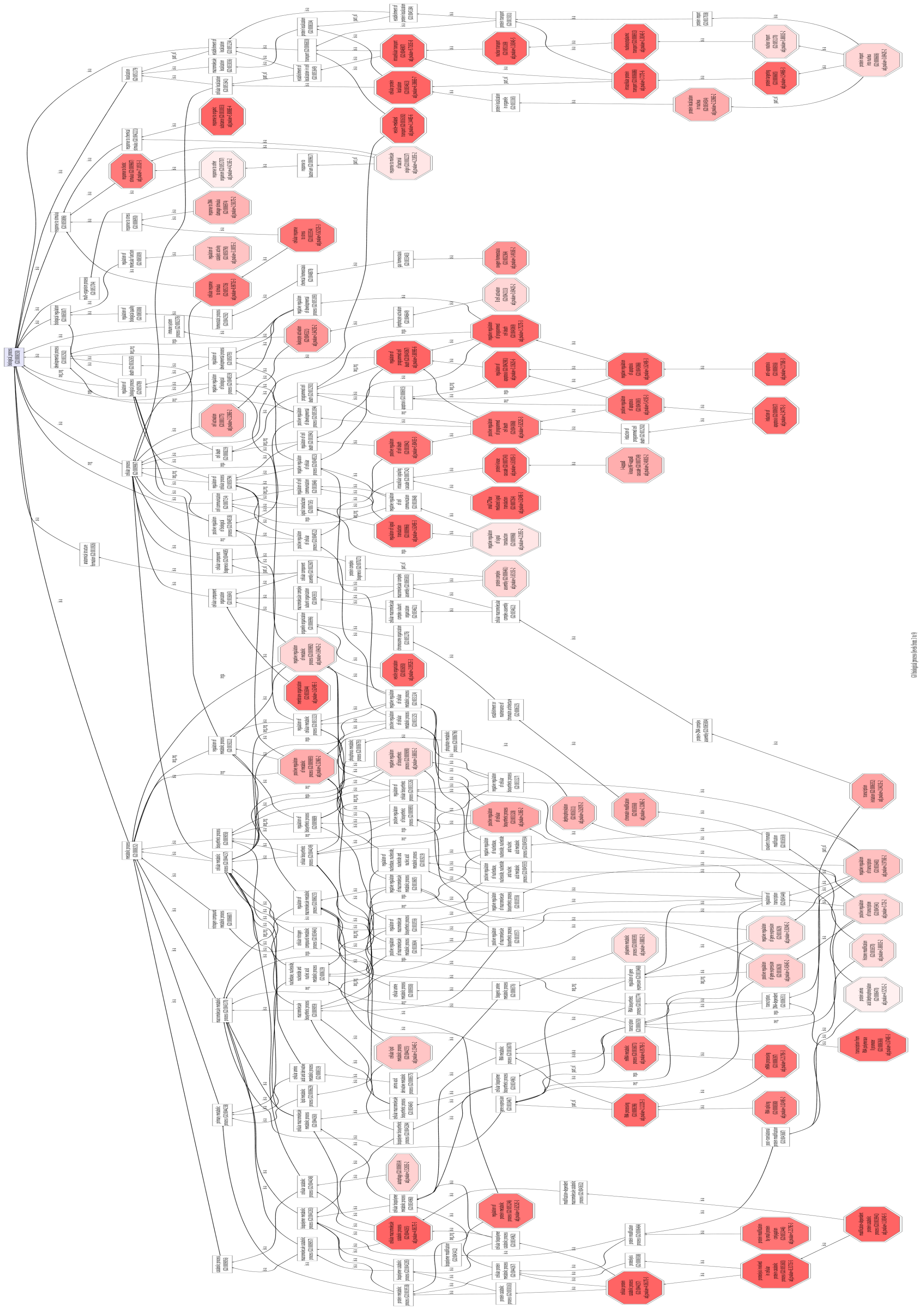


**Figure A.32:** GO terms of Blue HCL cluster.

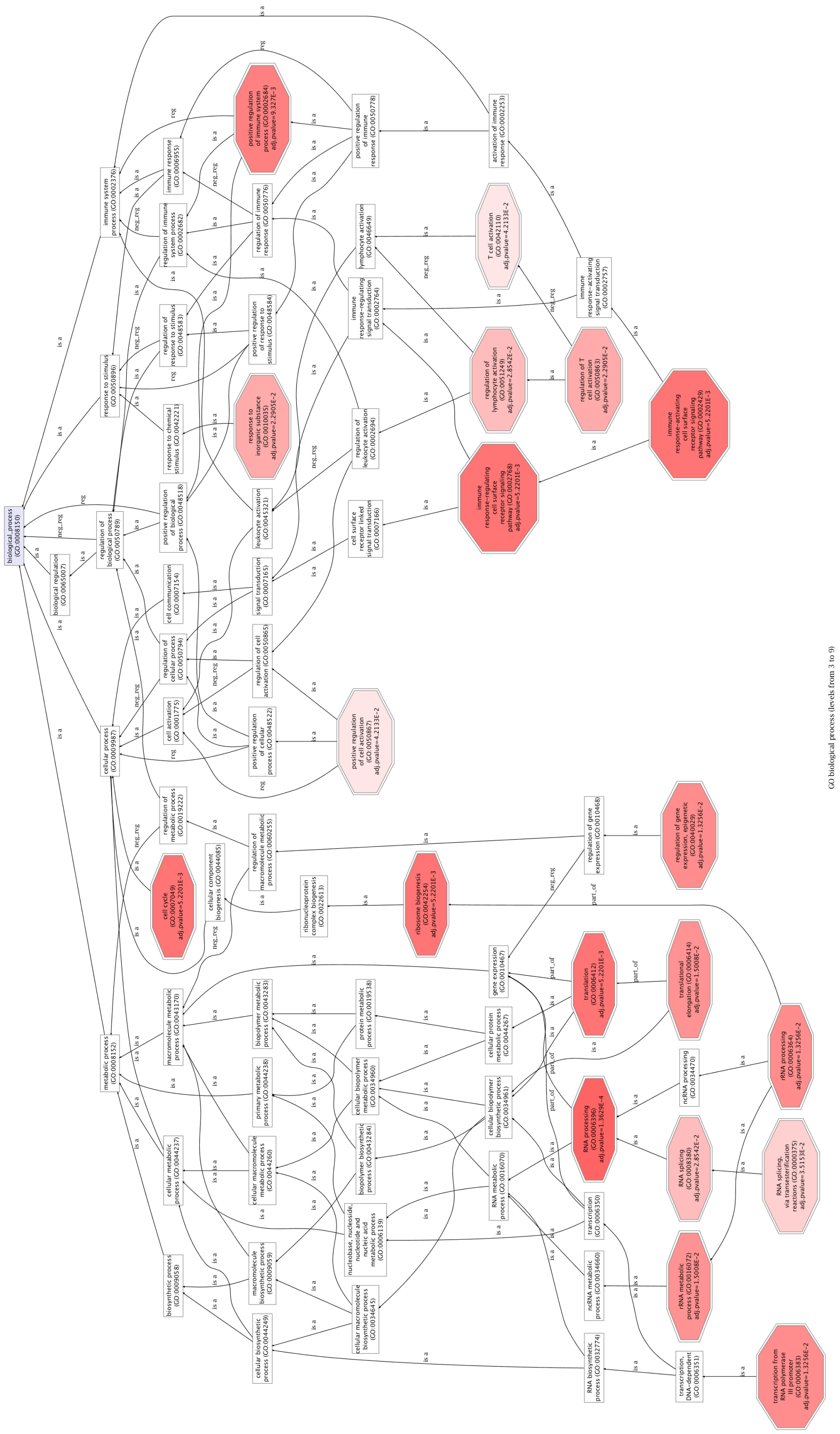

GO biological process (levels from 3 to 9)
biological_process (GO:0008150)
biological regulation (GO:0065007)
regulation of biological process (GO:0050789)
immune system process (GO:0002376)
response to stimulus (GO:0050896)
cellular process (GO:0009987)
metabolic process (GO:0008152)
immune response (GO:0006955)
regulation of immune system process (GO:0002682)
regulation of response to stimulus (GO:0048583)
response to chemical stimulus (GO:0042221)
positive regulation of biological process (GO:0048518)
cell communication (GO:0007154)
regulation of cellular process (GO:0050794)
cell activation (GO:0001775)
regulation of metabolic process (GO:0019222)
cell cycle (GO:0007049) adj.pvalue=5.2201E-3
cellular component biogenesis (GO:0044085)
macromolecule metabolic process (GO:0043170)
cellular metabolic process (GO:0044237)
biosynthetic process (GO:0009058)
positive regulation of immune system process (GO:0002684) adj.pvalue=9.327E-3
regulation of immune response (GO:0050776)
positive regulation of response to stimulus (GO:0048584)
response to inorganic substance (GO:0010035) adj.pvalue=2.2905E-2
leukocyte activation (GO:0045321)
signal transduction (GO:0007165)
regulation of cell activation (GO:0050865)
positive regulation of cellular process (GO:0048522)
regulation of macromolecule metabolic process (GO:0060255)
biopolymer metabolic process (GO:0043283)
primary metabolic process (GO:0044238)
cellular macromolecule metabolic process (GO:0044260)
macromolecule biosynthetic process (GO:0009059)
cellular biosynthetic process (GO:0044249)
positive regulation of immune response (GO:0050778)
lymphocyte activation (GO:0046649)
immune response-regulating signal transduction (GO:0002764)
regulation of leukocyte activation (GO:0002694)
cell surface receptor linked signal transduction (GO:0007166)
positive regulation of cell activation (GO:0050867) adj.pvalue=4.2133E-2
ribonucleoprotein complex biogenesis (GO:0022613)
protein metabolic process (GO:0019538)
cellular biopolymer metabolic process (GO:0034960)
biopolymer biosynthetic process (GO:0043284)
nucleobase, nucleoside, nucleotide and nucleic acid metabolic process (GO:0006139)
cellular macromolecule biosynthetic process (GO:0034645)
ribosome biogenesis (GO:0042254) adj.pvalue=5.2201E-3
activation of immune response (GO:0002253)
T cell activation (GO:0042110) adj.pvalue=4.2133E-2
regulation of lymphocyte activation (GO:0051249) adj.pvalue=2.8542E-2
immune response-regulating cell surface receptor signaling pathway (GO:0002768) adj.pvalue=5.2201E-3
gene expression (GO:0010467)
cellular protein metabolic process (GO:0044267)
cellular biopolymer biosynthetic process (GO:0034961)
RNA metabolic process (GO:0016070)
immune response-activating signal transduction (GO:0002757)
regulation of T cell activation (GO:0050863) adj.pvalue=2.2905E-2
regulation of gene expression (GO:0010468)
translation (GO:0006412) adj.pvalue=5.2201E-3
RNA processing (GO:0006396) adj.pvalue=1.3629E-4
transcription (GO:0006350)
ncRNA metabolic process (GO:0034660)
RNA biosynthetic process (GO:0032774)
immune response-activating cell surface receptor signaling pathway (GO:0002429) adj.pvalue=5.2201E-3
regulation of gene expression, epigenetic (GO:0040029) adj.pvalue=1.3256E-2
translational elongation (GO:0006414) adj.pvalue=1.5008E-2
ncRNA processing (GO:0034470)
RNA splicing (GO:0008380) adj.pvalue=2.8542E-2
rRNA metabolic process (GO:0016072) adj.pvalue=1.5008E-2
transcription, DNA-dependent (GO:0006351)
rRNA processing (GO:0006364) adj.pvalue=1.3256E-2
RNA splicing, via transesterification reactions (GO:0000375) adj.pvalue=3.5153E-2
transcription from RNA polymerase III promoter (GO:0006383) adj.pvalue=1.3256E-2
is a
reg
neg_reg
part_of

**Figure A.33:** GO terms of Turquoise HCL cluster.

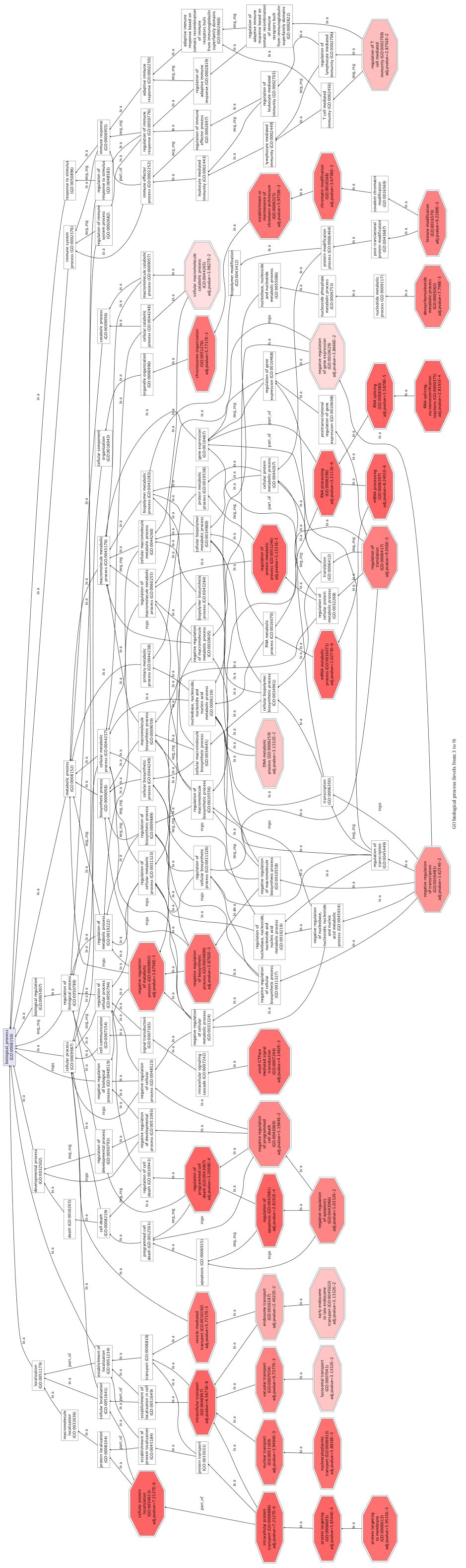


**Figure A.34:** GO terms of Turquoise module of k-means using $k = 20$.

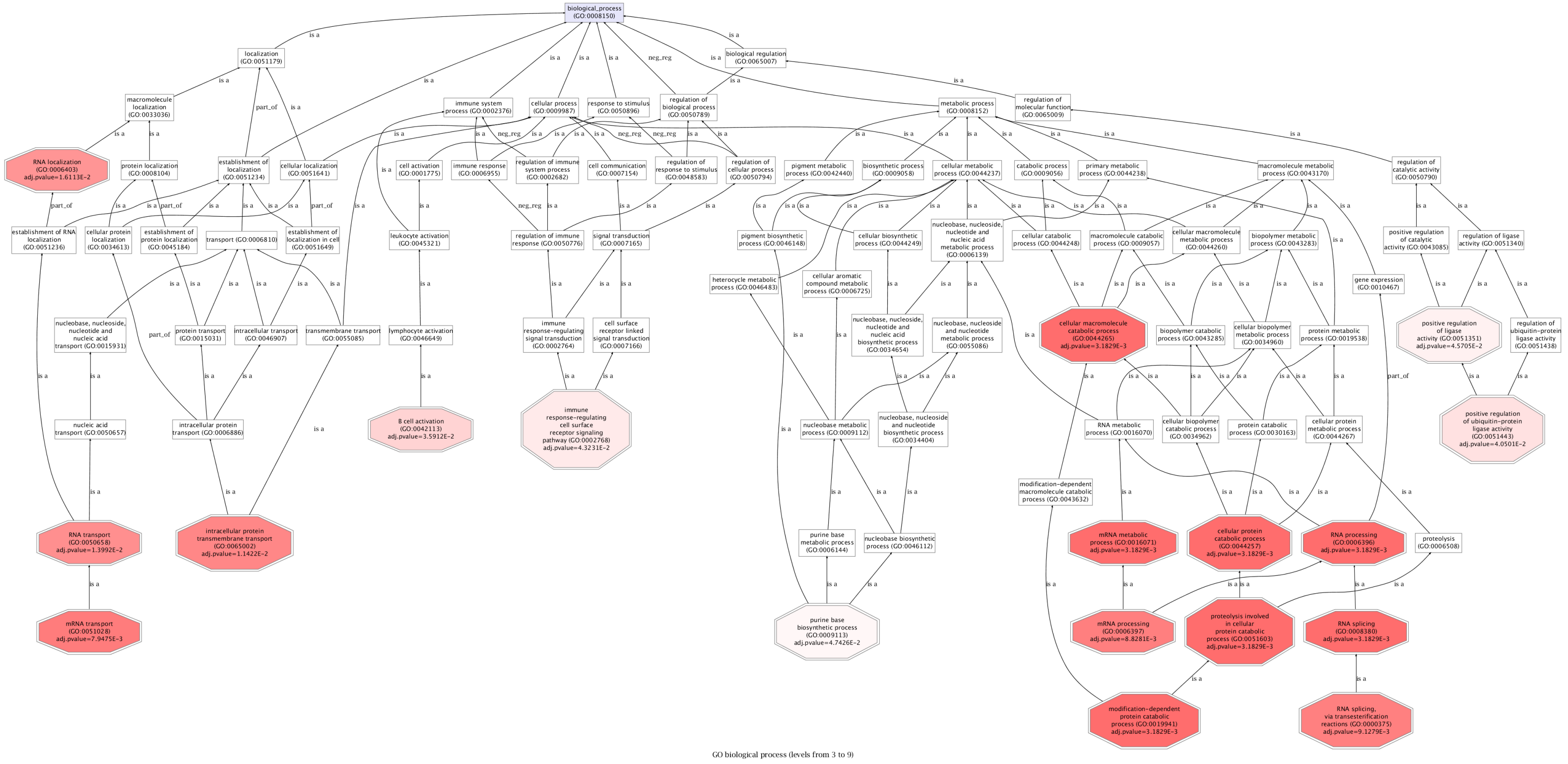

biological_process (GO:0008150)
localization (GO:0051179)
biological regulation (GO:0065007)
macromolecule localization (GO:0033036)
immune system process (GO:0002376)
cellular process (GO:0009987)
response to stimulus (GO:0050896)
regulation of biological process (GO:0050789)
metabolic process (GO:0008152)
regulation of molecular function (GO:0065009)
RNA localization (GO:0006403) adj.pvalue=1.6113E-2
protein localization (GO:0008104)
establishment of localization (GO:0051234)
cellular localization (GO:0051641)
cell activation (GO:0001775)
immune response (GO:0006955)
regulation of immune system process (GO:0002682)
cell communication (GO:0007154)
regulation of response to stimulus (GO:0048583)
regulation of cellular process (GO:0050794)
pigment metabolic process (GO:0042440)
biosynthetic process (GO:0009058)
cellular metabolic process (GO:0044237)
catabolic process (GO:0009056)
primary metabolic process (GO:0044238)
macromolecule metabolic process (GO:0043170)
regulation of catalytic activity (GO:0050790)
establishment of RNA localization (GO:0051236)
cellular protein localization (GO:0034613)
establishment of protein localization (GO:0045184)
transport (GO:0006810)
establishment of localization in cell (GO:0051649)
leukocyte activation (GO:0045321)
regulation of immune response (GO:0050776)
signal transduction (GO:0007165)
pigment biosynthetic process (GO:0046148)
cellular biosynthetic process (GO:0044249)
nucleobase, nucleoside, nucleotide and nucleic acid metabolic process (GO:0006139)
cellular catabolic process (GO:0044248)
macromolecule catabolic process (GO:0009057)
cellular macromolecule metabolic process (GO:0044260)
biopolymer metabolic process (GO:0043283)
positive regulation of catalytic activity (GO:0043085)
regulation of ligase activity (GO:0051340)
heterocycle metabolic process (GO:0046483)
cellular aromatic compound metabolic process (GO:0006725)
gene expression (GO:0010467)
nucleobase, nucleoside, nucleotide and nucleic acid transport (GO:0015931)
protein transport (GO:0015031)
intracellular transport (GO:0046907)
transmembrane transport (GO:0055085)
lymphocyte activation (GO:0046649)
immune response-regulating signal transduction (GO:0002764)
cell surface receptor linked signal transduction (GO:0007166)
nucleobase, nucleoside, nucleotide and nucleic acid biosynthetic process (GO:0034654)
nucleobase, nucleoside and nucleotide metabolic process (GO:0055086)
cellular macromolecule catabolic process (GO:0044265) adj.pvalue=3.1829E-3
biopolymer catabolic process (GO:0043285)
cellular biopolymer metabolic process (GO:0034960)
protein metabolic process (GO:0019538)
positive regulation of ligase activity (GO:0051351) adj.pvalue=4.5705E-2
regulation of ubiquitin-protein ligase activity (GO:0051438)
nucleic acid transport (GO:0050657)
intracellular protein transport (GO:0006886)
B cell activation (GO:0042113) adj.pvalue=3.5912E-2
immune response-regulating cell surface receptor signaling pathway (GO:0002768) adj.pvalue=4.3231E-2
nucleobase metabolic process (GO:0009112)
nucleobase, nucleoside and nucleotide biosynthetic process (GO:0034404)
RNA metabolic process (GO:0016070)
cellular biopolymer catabolic process (GO:0034962)
protein catabolic process (GO:0030163)
cellular protein metabolic process (GO:0044267)
positive regulation of ubiquitin-protein ligase activity (GO:0051443) adj.pvalue=4.0501E-2
modification-dependent macromolecule catabolic process (GO:0043632)
RNA transport (GO:0050658) adj.pvalue=1.3992E-2
intracellular protein transmembrane transport (GO:0065002) adj.pvalue=1.1422E-2
purine base metabolic process (GO:0006144)
nucleobase biosynthetic process (GO:0046112)
mRNA metabolic process (GO:0016071) adj.pvalue=3.1829E-3
cellular protein catabolic process (GO:0044257) adj.pvalue=3.1829E-3
RNA processing (GO:0006396) adj.pvalue=3.1829E-3
proteolysis (GO:0006508)
mRNA transport (GO:0051028) adj.pvalue=7.9475E-3
purine base biosynthetic process (GO:0009113) adj.pvalue=4.7426E-2
mRNA processing (GO:0006397) adj.pvalue=8.8281E-3
proteolysis involved in cellular protein catabolic process (GO:0051603) adj.pvalue=3.1829E-3
RNA splicing (GO:0008380) adj.pvalue=3.1829E-3
modification-dependent protein catabolic process (GO:0019941) adj.pvalue=3.1829E-3
RNA splicing, via transesterification reactions (GO:0000375) adj.pvalue=9.1279E-3
is a
part_of
neg_reg
GO biological process (levels from 3 to 9)

**Figure A.35:** GO terms of Blue module of k-means using $k = 20$.

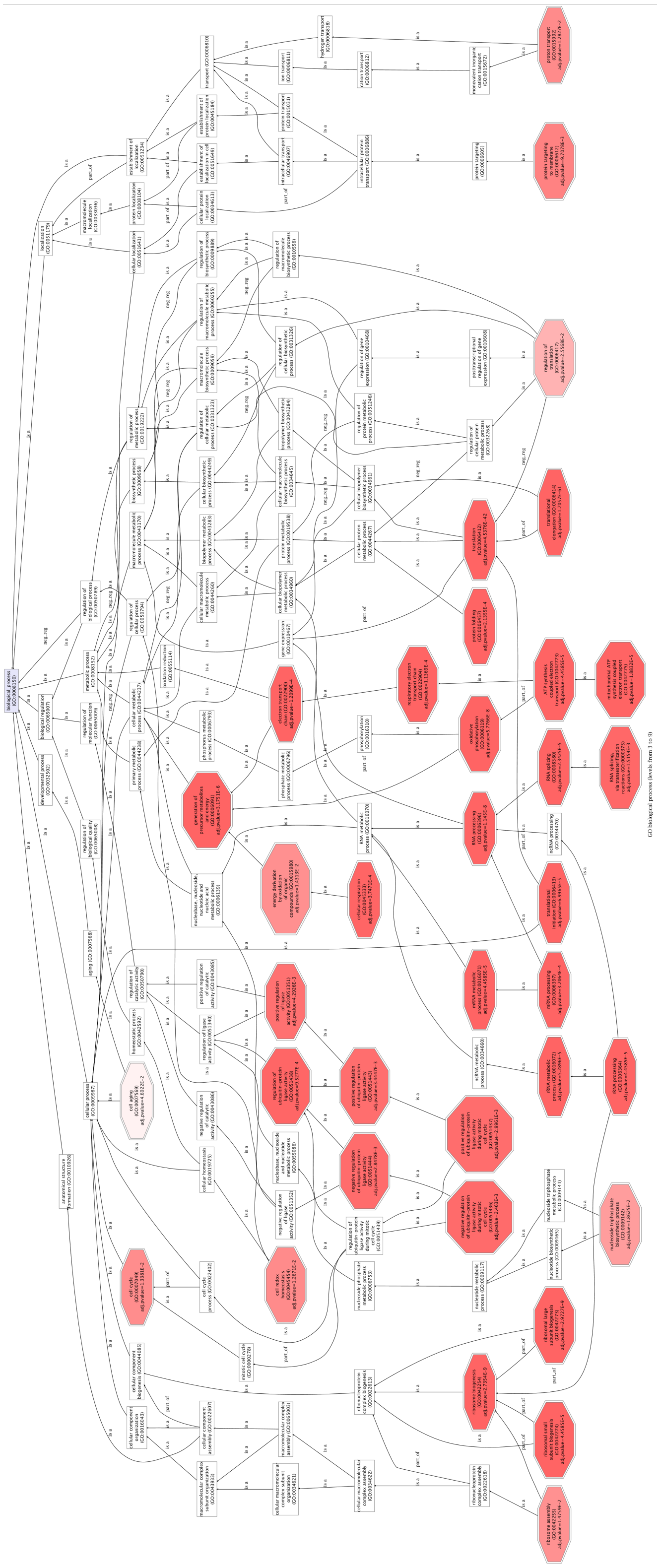


**Figure A.36:** GO terms of Brown module of k-means using $k = 20$.

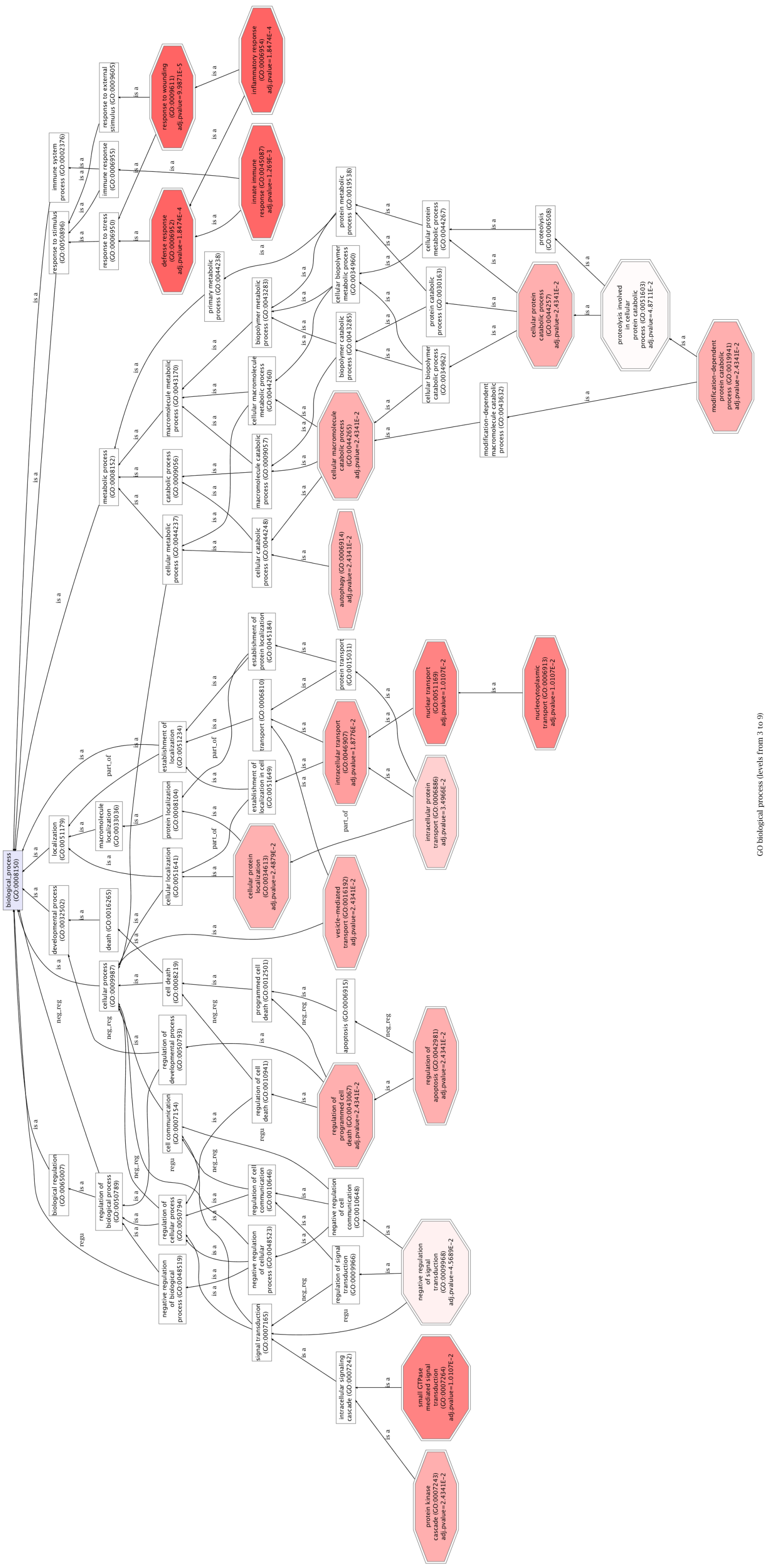

GO biological process (levels from 3 to 9)
biological_process (GO:0008150)
immune system process (GO:0002376)
response to stimulus (GO:0050896)
response to external stimulus (GO:0009605)
immune response (GO:0006955)
response to stress (GO:0006950)
response to wounding (GO:0009611) adj.pvalue=9.9871E-5
defense response (GO:0006952) adj.pvalue=1.8474E-4
inflammatory response (GO:0006954) adj.pvalue=1.8474E-4
innate immune response (GO:0045087) adj.pvalue=1.2619E-3
primary metabolic process (GO:0044238)
metabolic process (GO:0008152)
macromolecule metabolic process (GO:0043170)
catabolic process (GO:0009056)
cellular metabolic process (GO:0044237)
biopolymer metabolic process (GO:0043283)
cellular macromolecule metabolic process (GO:0044260)
macromolecule catabolic process (GO:0009057)
cellular catabolic process (GO:0044248)
protein metabolic process (GO:0019538)
cellular biopolymer metabolic process (GO:0034960)
biopolymer catabolic process (GO:0043285)
cellular macromolecule catabolic process (GO:0044265) adj.pvalue=2.4341E-2
autophagy (GO:0006914) adj.pvalue=2.4341E-2
cellular protein metabolic process (GO:0044267)
protein catabolic process (GO:0030163)
cellular biopolymer catabolic process (GO:0034962)
modification-dependent macromolecule catabolic process (GO:0043632)
proteolysis (GO:0006508)
cellular protein catabolic process (GO:0044257) adj.pvalue=2.4341E-2
proteolysis involved in cellular protein catabolic process (GO:0051603) adj.pvalue=4.8711E-2
modification-dependent protein catabolic process (GO:0019941) adj.pvalue=2.4341E-2
localization (GO:0051179)
macromolecule localization (GO:0033036)
establishment of localization (GO:0051234)
protein localization (GO:0008104)
cellular localization (GO:0051641)
establishment of protein localization (GO:0045184)
transport (GO:0006810)
establishment of localization in cell (GO:0051649)
cellular protein localization (GO:0034613) adj.pvalue=2.4879E-2
protein transport (GO:0015031)
intracellular transport (GO:0046907) adj.pvalue=1.8776E-2
vesicle-mediated transport (GO:0016192) adj.pvalue=2.4341E-2
nuclear transport (GO:0051169) adj.pvalue=1.0107E-2
nucleocytoplasmic transport (GO:0006913) adj.pvalue=1.0107E-2
intracellular protein transport (GO:0006886) adj.pvalue=3.4966E-2
developmental process (GO:0032502)
death (GO:0016265)
cellular process (GO:0009987)
cell death (GO:0008219)
programmed cell death (GO:0012501)
apoptosis (GO:0006915)
regulation of developmental process (GO:0050793)
regulation of cell death (GO:0010941)
regulation of programmed cell death (GO:0043067) adj.pvalue=2.4341E-2
regulation of apoptosis (GO:0042981) adj.pvalue=2.4341E-2
cell communication (GO:0007154)
biological regulation (GO:0065007)
regulation of biological process (GO:0050789)
regulation of cellular process (GO:0050794)
regulation of cell communication (GO:0010646)
negative regulation of biological process (GO:0048519)
negative regulation of cellular process (GO:0048523)
negative regulation of cell communication (GO:0010648)
regulation of signal transduction (GO:0009966)
negative regulation of signal transduction (GO:0009968) adj.pvalue=4.5689E-2
signal transduction (GO:0007165)
intracellular signaling cascade (GO:0007242)
small GTPase mediated signal transduction (GO:0007264) adj.pvalue=1.0107E-2
protein kinase cascade (GO:0007243) adj.pvalue=2.4341E-2
is a
part_of
neg_reg
regu

**Figure A.37:** GO terms of Yellow module of k-means using $k = 20$.

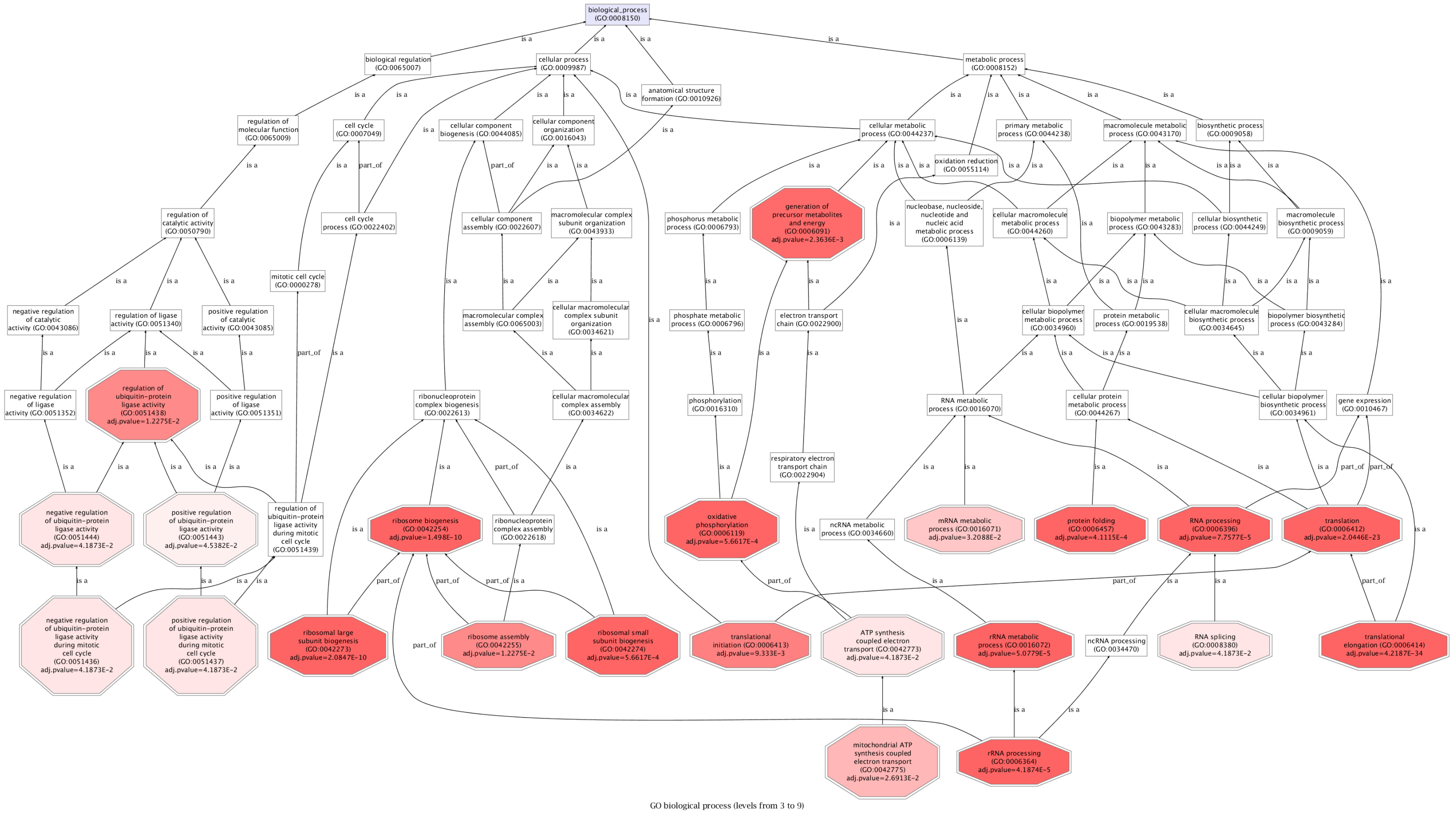

biological_process (GO:0008150)
biological regulation (GO:0065007)
cellular process (GO:0009987)
anatomical structure formation (GO:0010926)
metabolic process (GO:0008152)
regulation of molecular function (GO:0065009)
cell cycle (GO:0007049)
cellular component biogenesis (GO:0044085)
cellular component organization (GO:0016043)
cellular metabolic process (GO:0044237)
primary metabolic process (GO:0044238)
macromolecule metabolic process (GO:0043170)
biosynthetic process (GO:0009058)
oxidation reduction (GO:0055114)
regulation of catalytic activity (GO:0050790)
cell cycle process (GO:0022402)
cellular component assembly (GO:0022607)
macromolecular complex subunit organization (GO:0043933)
phosphorus metabolic process (GO:0006793)
generation of precursor metabolites and energy (GO:0006091) adj.pvalue=2.3636E-3
nucleobase, nucleoside, nucleotide and nucleic acid metabolic process (GO:0006139)
cellular macromolecule metabolic process (GO:0044260)
biopolymer metabolic process (GO:0043283)
cellular biosynthetic process (GO:0044249)
macromolecule biosynthetic process (GO:0009059)
mitotic cell cycle (GO:0000278)
negative regulation of catalytic activity (GO:0043086)
regulation of ligase activity (GO:0051340)
positive regulation of catalytic activity (GO:0043085)
macromolecular complex assembly (GO:0065003)
cellular macromolecular complex subunit organization (GO:0034621)
phosphate metabolic process (GO:0006796)
electron transport chain (GO:0022900)
cellular biopolymer metabolic process (GO:0034960)
protein metabolic process (GO:0019538)
cellular macromolecule biosynthetic process (GO:0034645)
biopolymer biosynthetic process (GO:0043284)
negative regulation of ligase activity (GO:0051352)
regulation of ubiquitin-protein ligase activity (GO:0051438) adj.pvalue=1.2275E-2
positive regulation of ligase activity (GO:0051351)
ribonucleoprotein complex biogenesis (GO:0022613)
cellular macromolecular complex assembly (GO:0034622)
phosphorylation (GO:0016310)
RNA metabolic process (GO:0016070)
cellular protein metabolic process (GO:0044267)
cellular biopolymer biosynthetic process (GO:0034961)
gene expression (GO:0010467)
respiratory electron transport chain (GO:0022904)
negative regulation of ubiquitin-protein ligase activity (GO:0051444) adj.pvalue=4.1873E-2
positive regulation of ubiquitin-protein ligase activity (GO:0051443) adj.pvalue=4.5382E-2
regulation of ubiquitin-protein ligase activity during mitotic cell cycle (GO:0051439)
ribosome biogenesis (GO:0042254) adj.pvalue=1.498E-10
ribonucleoprotein complex assembly (GO:0022618)
oxidative phosphorylation (GO:0006119) adj.pvalue=5.6617E-4
ncRNA metabolic process (GO:0034660)
mRNA metabolic process (GO:0016071) adj.pvalue=3.2088E-2
protein folding (GO:0006457) adj.pvalue=4.1115E-4
RNA processing (GO:0006396) adj.pvalue=7.7577E-5
translation (GO:0006412) adj.pvalue=2.0446E-23
negative regulation of ubiquitin-protein ligase activity during mitotic cell cycle (GO:0051436) adj.pvalue=4.1873E-2
positive regulation of ubiquitin-protein ligase activity during mitotic cell cycle (GO:0051437) adj.pvalue=4.1873E-2
ribosomal large subunit biogenesis (GO:0042273) adj.pvalue=2.0847E-10
ribosome assembly (GO:0042255) adj.pvalue=1.2275E-2
ribosomal small subunit biogenesis (GO:0042274) adj.pvalue=5.6617E-4
translational initiation (GO:0006413) adj.pvalue=9.333E-3
ATP synthesis coupled electron transport (GO:0042773) adj.pvalue=4.1873E-2
rRNA metabolic process (GO:0016072) adj.pvalue=5.0779E-5
ncRNA processing (GO:0034470)
RNA splicing (GO:0008380) adj.pvalue=4.1873E-2
translational elongation (GO:0006414) adj.pvalue=4.2187E-34
mitochondrial ATP synthesis coupled electron transport (GO:0042775) adj.pvalue=2.6913E-2
rRNA processing (GO:0006364) adj.pvalue=4.1874E-5
GO biological process (levels from 3 to 9)

**Figure A.38:** GO terms of Brown of k-means using $k = 50$.