TECHNICAL UNIVERSITY OF CRETE, SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING, DIGITAL SIGNAL & IMAGE PROCESSING LAB

Development and Analysis of Biological Interaction Networks

M.Sc. Thesis Tsakaneli Stavroula

Chania, December 2022

Thesis Committee
M.Sc Thesis Supervisor Professor Michael Zervakis,
Professor Athanasios Liavas
Professor Apostolos Dollas

Abstract

Multiple sclerosis (MS) is a chronic inflammatory demyelinating disease that affects approximately 2.8 million persons globally. While there is currently no cure for this neurodegenerative disease, MS has become a highly manageable disease through treatment options like disease-modifying medications, that can help to control the symptoms and slow disease progression. Among them, interferon beta (IFNβ) therapy is a first-line treatment for MS but has shown to be only partially effective. Information from extensive databases for large groups of multiple sclerosis patients indicates that the natural history of MS evolves in two stages: (i) in the focal inflammatory process with flares, and ii) in disability that progresses irrespective of the focal inflammation (lesion or relapse) Thus, it is important to identify biomarkers that aid in early identification of the disease as well as of IFNβ responders. A second aim of our study was to identify biomarkers that aid in early identification of MS stages, i.e. the relapsing-remitting form (RRMS), the secondary progressive phase (SPMS) and the primary progressive MS (PPMS).

Gene co-expression patterns for various phenotypes can be reveal with the aid of microarrays but the variation and heterogeneity of the disease act as limitations for the utility of gene-expression profiles. In addition, the different microarray platforms utilized, as well as the different experimental protocols followed, make difficult to combine gene-expression datasets from heterogeneous platforms and different studies. Another limitation is the great imbalance between the huge number of transcripts and genes (tens of thousands) and the relatively small number of available sample cases (hundreds). Furthermore, it is essential to combine feature-selection approaches and the 'biological validity' of the resulted gene biomarkers. Thus, our purpose in not only to focus on highly differential genes but combine different approaches in order to reach a gene signature after examining the relationships of gene signatures and deduce submodules of greater prognostic/diagnostic significance in relation to Multiple Sclerosis, the progression of the disease and future therapy.

In this study, based on gene expression profiles from untreated, interferon treated patients and healthy subjects from publicly available datasets, we performed differential expression analysis and Pigengene network association (weighted correlation network analysis (WGCNA) and Bayesian networks modeling) so as to construct a high-confidence protein-protein (PPI) interaction network. Subsequently, aiming to find the most significant clustering modules and hub genes, we applied several topological analysis methods (cytoHubba plugin) followed by MCODE clustering algorithm. Our approach resulted in highly connected hub genes generating four highly reliable hub-gene-signatures. Finally, we approached the topic of drug repurposing by examining the drug-gene relationships through different databases.

Περίληψη

Η σκλήρυνση κατά πλάκας (ΣΚΠ) είναι μια χρόνια φλεγμονώδης απομυελινωτική νόσος που επηρεάζει περίπου 2,8 εκατομμύρια άτομα παγκοσμίως. Ενώ επί του παρόντος δεν υπάρχει θεραπεία για αυτή τη νευροεκφυλιστική νόσο, η σκλήρυνση κατά πλάκας έχει γίνει μια εξαιρετικά διαχειρίσιμη ασθένεια μέσω επιλογών θεραπείας όπως τα τροποποιητικά της νόσου φάρμακα, που μπορούν να βοηθήσουν στον έλεγχο των συμπτωμάτων και στην επιβράδυνση της εξέλιξης της νόσου. Μεταξύ αυτών, η θεραπεία με ιντερφερόνη βήτα (IFNβ) είναι μια θεραπεία πρώτης γραμμής για τη σκλήρυνση κατά πλάκας, αλλά έχει αποδειχθεί μόνο μερικώς αποτελεσματική. Πληροφορίες από εκτεταμένες βάσεις δεδομένων για μεγάλες ομάδες ασθενών με σκλήρυνση κατά πλάκας δείχνουν ότι η φυσική ιστορία της ΣΚΠ εξελίσσεται σε δύο στάδια: (i) στην εστιακή φλεγμονώδη διαδικασία με εξάρσεις και ii) στην αναπηρία που εξελίσσεται ανεξάρτητα από την εστιακή φλεγμονή (βλάβη ή υποτροπή). Επομένως, είναι σημαντικό να εντοπιστούν βιοδείκτες που βοηθούν στην έγκαιρη αναγνώριση της νόσου καθώς και των αποκρίσεων στην IFNβ. Ένας δεύτερος στόχος της μελέτης μας είναι ο εντοπισμός βιοδεικτών που βοηθούν στην πρώιμη αναγνώριση των μορφών της ΣΚΠ, δηλαδή της υποτροπιάζουσας-διαλείπουσας μορφής πολλαπλής σκλήρυνσης (RRMS), της δευτεροπαθώς προϊούσας μορφής πολλαπλής σκλήρυνσης (SPMS) και της πρωτοπαθώς προϊούσας μορφής πολλαπλής σκλήρυνσης (PPMS).

Μοτίβα συνέκφρασης γονιδίων για διάφορους φαινοτύπους μπορούν να αποκαλυφθούν με τη βοήθεια μικροσυστοιχιών, αλλά η ποικιλία και η ετερογένεια της νόσου λειτουργούν ως περιορισμοί για τη χρησιμότητα των προφίλ γονιδιακής έκφρασης. Επιπλέον, οι διαφορετικές πλατφόρμες μικροσυστοιχιών που χρησιμοποιούνται, καθώς και τα διαφορετικά πειραματικά πρωτόκολλα που ακολουθούνται καθιστούν δύσκολο τον συνδυασμό δεδομένων γονιδιακής έκφρασης από ετερογενείς πλατφόρμες και διαφορετικές μελέτες. Ένας άλλος περιορισμός είναι η μεγάλη ανισορροπία μεταξύ του τεράστιου αριθμού των μεταγράφων και γονιδίων (δεκάδες χιλιάδες) και του σχετικά μικρού αριθμού διαθέσιμων δειγμάτων (εκατοντάδες). Επιπλέον, είναι σημαντικό να συνδυαστούν οι προσεγγίσεις επιλογής χαρακτηριστικών και η «βιολογική εγκυρότητα» των γονιδιακών βιοδεικτών που προέκυψαν. Έτσι, ο σκοπός μας είναι όχι μόνο να επικεντρωθούμε σε σημαντικά διαφορικά εκφρασμένα γονίδια, αλλά να συνδυάσουμε διαφορετικές προσεγγίσεις για να εξάγουμε μια γονιδιακή υπογραφή με υψηλή προγνωστική/διαγνωστική αξία για την Σκλήρυνση κατά Πλάκας, την εξέλιξη της νόσου και τη μελλοντική θεραπεία, αφού εξετάσουμε τις σχέσεις μεταξύ των γονιδιακών υπογραφών και συνάγουμε υποδίκτυα.

Σε αυτή τη μελέτη, χρησιμοποιήσαμε δημόσια διαθέσιμα σύνολα δεδομένων και με βάση τα προφίλ γονιδιακής έκφρασης από: α) ασθενείς που δεν υποβλήθηκαν σε θεραπεία με ιντερφερόνη και άτομα που υποβλήθηκαν σε θεραπεία με ιντερφερόνη, β) άτομα που δεν υποβλήθηκαν σε θεραπεία με ιντερφερόνη και υγιή άτομα, και γ) άτομα που ανήκουν σε μια από τις τρεις κύριες μορφές πολλαπλής σκλήρυνσης και δεν έχουν υποβληθεί σε θεραπεία και υγιή άτομα, πραγματοποιήσαμε ανάλυση διαφορικής έκφρασης και συσχέτιση δικτύου Pigengene (σταθμισμένη ανάλυση δικτύου συσχέτισης (WGCNA) και μοντελοποίηση δικτύων Bayes), έτσι ώστε να κατασκευάσουμε ένα ισχυρό δίκτυο πρωτεϊνικών αλληλεπιδράσεων (PPI). Στη συνέχεια, με στόχο την εύρεση των πιο σημαντικών μονάδων ομαδοποίησης και σημαντικών γονιδίων, εφαρμόσαμε διάφορες μεθόδους τοπολογικής ανάλυσης (cytoHubba) ακολουθούμενες από τον αλγόριθμο ομαδοποίησης MCODE. Η προσέγγισή μας είχε ως αποτέλεσμα υψηλά συνδεδεμένα γονίδια (hub) που παράγουν τέσσερις εξαιρετικά εύρωστες 'γονιδιακές υπογραφές κόμβων' με υψηλή απόδοση ταξινόμησης. Τέλος, προσεγγίσαμε το θέμα της επαναχρησιμοποίησης φαρμάκων εξετάζοντας τις σχέσεις φαρμάκου-γονιδίου μέσα από διαφορετικές βάσεις δεδομένων.

Acknowledgements

Firstly, I would like to thank my thesis supervisor, Professor Michalis Zervakis for his guidance, patience, support and for giving me the chance to expand my knowledge in the exciting field of bioinformatics. Dr. Katerina Bei for her support, devoted time and biological insight that she offered me.

Moreover, Professor Athanasios Liavas and Professor Apostolos Dollas for their contribution as members of the thesis committee.

Last but not least, I would like to give my warmest thanks, to my family and Michael for their love, support and encouragement. Nothing would be possible without them.

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Multiple sclerosis (MS) is the most common autoimmune disease, a potentially disabling disease of the brain and spinal cord, the central nervous system (CNS). It is characterized by the infiltration of autoreactive immune cells into the CNS, which target the myelin sheath, leading to the loss of neuronal function. Eventually, the disease can cause permanent damage or deterioration of the nerves. Signs and symptoms of MS vary widely and depend on the amount of nerve damage and which nerves are affected. Some people with severe MS may lose the ability to walk independently or at all, while others may experience long periods of remission without any new symptoms. Although it is accepted that MS is a multifactorial disorder with both genetic and environmental factors influencing its development and course, the molecular pathogenesis of multiple sclerosis (MS) has not yet been fully elucidated. There's currently no cure for multiple sclerosis. However, the growing arsenal of disease-modifying therapies offers opportunities to reduce disability and extend survival of people with multiple sclerosis (MS). [1]

According to 2020 data on multiple sclerosis (MS), the number of people suffering from the disease worldwide amount to 2.8 million [1]. MS is the most common demyelinating disease that affects the central and peripheral nervous system. This autoimmune disorder shows a significant variation in prevalence, reaching high levels in Europe (lower in the South, higher in the North). Although the etiology of this multifactorial disease remains unknown, the implications of environmental and immunogenetic factors appear to be major [2]. Information from extensive databases for large groups of multiple sclerosis patients indicates that the natural history of MS evolves in two stages: (i) in the focal inflammatory process with flares, and ii) in disability that progresses irrespective of the focal inflammation (lesion or relapse) [2]. Despite its impact and increasing rates on the global population, there is still no cure for MS. Among the available treatments, disease-modifying therapies such as interferon beta (IFNβ) are designed to help patients by reducing the relapse rates and delaying the onset of disability [3]. Although IFNβ is used as first-line therapy, many MS patients do not benefit from this treatment.

Chapter 1 proceeds with a brief presentation of the disease, the biological and bioinformatics perspectives, the related work and thesis outline and innovation. In Chapter 2 the biological and mathematical knowledge in bioinformatics, needed for our study is presented. Our Methodology pipeline is explained in Chapter 3. The process of our data, integration, differential expression analysis and network construction, are explained in detail. After generating the subnetworks and extracting the final gene signatures, we examine the nature of the involved pathways as well as the relationships between genes. Furthermore, we evaluate our results in a new independent dataset, after applying a classification algorithm, SVM, and also taking into consideration their biological significance. Finally, we examine the potential of drug repurposing based on our results which are presented in Chapter 4.

1.1 Multiple Sclerosis

Multiple sclerosis (MS) is an inflammatory demyelinating disease of the central nervous system (CNS) with varied clinical presentations and heterogeneous histopathological features (Figure1.1). The underlying immunological abnormalities in MS lead to various neurological and autoimmune manifestations. There is strong evidence that MS is, at least in part, an immune-mediated disease. Immunogenetic markers have been identified and, in particular thanks to studies of genome-wide associations, more than 100 genetic variants have been reported. Most of these are involved in the immune response and often associated with other autoimmune diseases. Studies of the natural history of MS suggest it is a two-phase disease: in the first phase, inflammation is focal with flares; and in the second phase, disability progresses independently of focal inflammation. This has clear implications for therapy. [2]

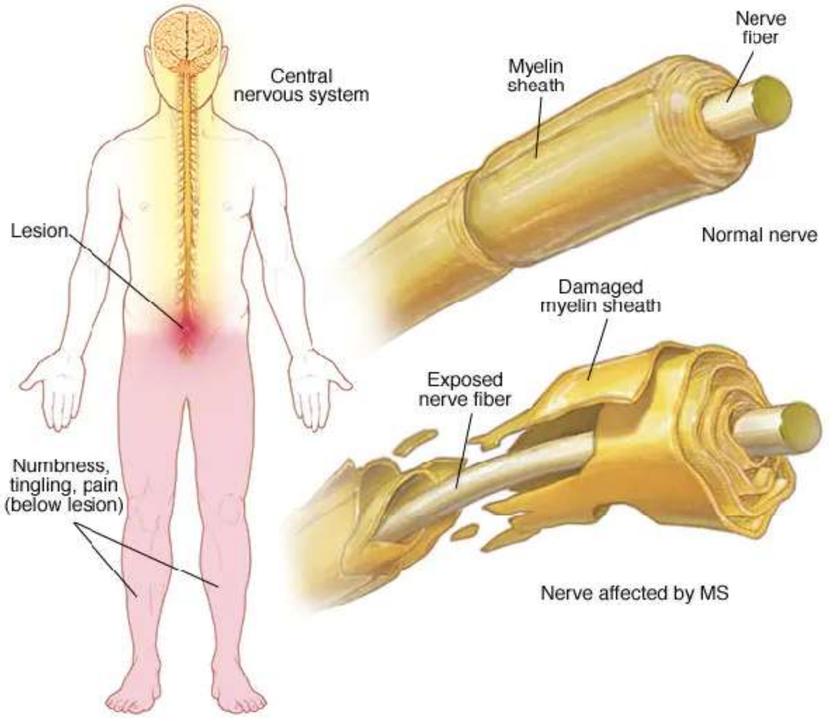


FIGURE 1.1.MULTIPLE SCLEROSIS COURSE OF ACTION [4]

Studies using imaging, serology, pathology and genetics, and patient response to anti-inflammatory treatments indicate that multiple sclerosis (MS) is primarily an inflammatory demyelinating disease of the central nervous system (CNS) with varied clinical presentations and heterogeneous histopathological features. The disease has a peak onset between ages 20 and 40 years [4]; however, it may also develop in children and in addition has been reported in individuals aged above 60 years. MS affects women approximately twice as often as men [5–8]. MS results in a plethora of neurological manifestations and is a leading cause of nontraumatic disability among young adults and has great socioeconomic impact in developed countries [9]. Based on the epidemiological studies, approximately 400,000 people have MS in the United States, with 200 new cases added every week. In Europe is the leading cause of non-traumatic disabilities in young adults, with more than 700,000 EU cases. The pathogenesis of MS remains elusive and there were no definitive cause and no effective cure. Therefore, MS can be classified as an episodic demyelinating disease of the central nervous system. The two main factors of MS are genetic and environmental. Exposure to Epstein-Barr virus [10], low levels of vitamin D [11-12], and smoking [13] have been cited as plausible factors, which may increase the probabilities of developing MS.

The commonly used disease-modifying therapies (DMTs), interferon (IFN) beta and Glatiramer acetate are believed to modulate the immune response, reduce new inflammatory lesions in the CNS and partially protect against progression of disability. However, patients vary considerably in their responsiveness to these therapies, and for any individual patient, the natural history of MS is extremely heterogeneous, varying from a benign condition to a devastating and rapidly incapacitating disease. For these reasons, a better characterization of patients is much needed to ultimately understand the diversity of disease presentation. A number of studies in neurodegenerative disorders and autoimmune diseases [9, 14-16] suggest that gene expression changes in blood mirror pathologic processes in the CNS. Blood transcriptomics have also been used to study therapeutic response to treatment with different drugs, toxins and infections in different diseases [17–19]. Several microarray-based gene expression studies have used whole blood or peripheral blood mononuclear cells (PBMCs) to investigate de-regulated patterns of gene expression in MS patients [20-22]. Unfortunately, owing to small sample sizes and disease heterogeneity, reproducibility across studies has been limited.

1.2 Multiple Sclerosis and Bioinformatics

Multiple Sclerosis occurs in both men and women, in younger as well as older individuals. Although a cure has not yet been found, identifying the genetic causes that rule the disease can play an important role. Bioinformatics is an integrative area combining biological, statistical and computational sciences. Bioinformatics enables researchers not only to manage, analyze and understand the currently accumulated, valuable, high-throughput data, but also to integrate these in their current research programs. The need for bioinformatics will become even more important as new technologies increase the already exponential rate at which data are generated. Computational models could give a considerable advance in the study of diseases characterized by a partially understood etiology of the disease. The main goal of bioinformatics is to enable the discovery of new biological insights as well as to create a global perspective from which unifying principles in biology can be discerned. We have therefore to do with the development and the advancement of databases, algorithms, computational and statistical techniques and theory to solve formal and practical problems arising from the management and analysis of biological data. (www.wikidoc.org/index.php/Bioinformatics)

1.3 Genomic and Network Analysis

In genetics the term *Genomics* refers to the field that combines recombinant DNA, DNA sequencing methods, and bioinformatics to sequence, assemble, and analyze the function and structure of genomes. Functional genomics employs diverse experimental approaches to investigate gene functions. High-throughput techniques, such as loss-of-function screening and transcriptome profiling, allow the identification of specific sets of genes involved in biological processes of interest (so called hit list of genes). [23-24]

Gene expression profiling is being applied in many areas of research in order to identify new targets for treatment, resistance mechanisms and to improve the current tools of prognosis and treatment. Pathways analysis methods, aim at searching for statistical enrichment of genes with annotated biological process or molecular functions. Thought computational scientists and statisticians that participate in the process of data analysis are often not well informed of the sample collection processes or the impact of genetics/transcriptomics. Therefore, a pressing need has occurred for better understanding of the challenges and limitations of high-throughput approaches, both in experimental design and data analysis. [24-25]

The investigation of the roles and functions of single genes is a primary focus of molecular biology or genetics and is a common topic of modern medical and biological research. Understanding complex systems often requires a bottom-up analysis towards a systems biology approach. The need to investigate a system, not only as individual components but as a whole, emerges. This can be done by examining the elementary constituents individually and then how these are connected. The myriad components of a system and their interactions are best characterized as networks and they are mainly represented as graphs where thousands of nodes are connected with thousands of vertices. [25]

In the field of Bioinformatics the main goal of several studies has been revealing the pathways that give rise to diseases, identifying genetic alterations that determine clinical phenotypes as well as identification of both gene and protein networks causing a disease as well as the investigation of biochemical networks of drugs metabolism and mechanisms of action. Network biology involves the study of the complex interactions of biomolecules that contribute to the structures and functions of living cells. Given the functional interdependencies between the molecular components in a human cell, a disease is rarely a consequence of an abnormality in a single gene but reflects the perturbations of the complex intracellular and intercellular network that links tissue and organ systems [25]. Once the model has been chosen, the parameters need to be fit to the data. Even the simplest network models are complex systems involving many parameters, and fitting them is a non-trivial process, known as network inference, network identification, or reverse engineering. Genetic networks are often described statistically using graphical models. The interpretation of the network structure constitutes a serious challenge in microarray analysis due to the fact that the sample size is small compared to the number of considered genes. As a result, many standard algorithms for graphical models are considered inapplicable. In order to better understand genetic networks, we have to look at graph theory and models. [26]

Graph theoretical models (GTMs) are used mainly to describe the topology, or architecture, of a network. These models feature relationships between genes and possibly their nature, but not dynamics: the time component is not modeled at all and simulations cannot be performed. GTMs are particularly useful for knowledge representation, as most of the current knowledge about gene networks is presented and stored in databases in a graph format. In GTMs, gene networks are represented by a graph structure, G (V, E), where V = {1, 2,.., n} represent the gene regulatory elements, e.g. genes, proteins, etc., and E = {(I, j) |I, j ∈ V } the interactions between them, e.g. activation, inhibition, causality, binding specificity, etc. Most often G is a simple graph, and the edges represent relationships between pairs of nodes, although hyper edges, connecting three or more nodes at once, are sometimes appropriate. Edges can be directed, indicating that one (or more) nodes are precursors to other nodes. They can also be weighted, the weights indicating the strengths of the relationships. Either the nodes, or the edges, or both are sometimes labeled with the function, or nature of the relationship, i.e. activator, activation, inhibitor, inhibition, etc. The edges imply relationships which can be interpreted as temporal (e.g. causal relationship) or interactional. (en.wikipedia.org/wiki/Graph_theory)

1.4 Related Work

High-throughput techniques, such as loss-of-function screening and transcriptome profiling, allow to identify lists of genes potentially involved in biological processes of interest (so called hit list). Several computational methods exist to analyze and interpret such lists, the most widespread of which aim either at investigating of significantly enriched biological processes, or at extracting significantly represented subnetworks. Also, in the field of drug discovery, taking into account that discovery and design is a time-consuming process, it often requires a lengthy period. A drug prescribed for a specific disease can be also effective for another disease if the two diseases share a common pathophysiologic mechanism. To identify a new use of existing drugs is called drug repositioning, and this approach is gathering momentum because it can markedly shorten the time to obtain drug approval.[27]

In order to comprehend the mechanisms and improve the methods of prognosis and treatment many studies focus on the analysis of gene expression profiles to identify markers linked to a disease as well as pathways and associations between gene expression and phenotype which can be extended to enable systematic search for candidates for drug repositioning [28]. Protein-protein interaction (PPI) networks, co-expression networks or pathways from databases such as KEGG, has been proposed to overcome variability of prognostic signatures and to increase prognostic performance. Relevant studies have been made that focus on the interaction or association between genes and clinical outcomes and the discovery of disease-related gene signatures and the integration of PPI networks in their methodology [27-29].

Y Liu et al., 2019 [30] proposed a methodology combining gene expression data for the investigation of hub genes in bipolar disorder integrating PPI networks and graph theory. In addition, Machine learning techniques for biological networks are proposed in [31]. Most recent work in Multiple Sclerosis to identify the potential key candidate genes of MS and uncover mechanisms in the disease is [32-33] where data from the microarray expression profile of MS patients were combined and bioinformatics analysis was performed. Defective pathways suggest viral or bacterial infections as plausible mechanisms involved in MS development were examined in [34] providing additional knowledge to identify new therapeutic targets.

This thesis is based on the study of D Nickles et al., 2013 [35] combining different methodological approaches [32-37] to create a new pipeline for disease investigation, gene signature discovery and drug repositioning analysis for Multiple Sclerosis. D Nickles et al., 2013 study proposes a protein network-based approach that identifies markers not as individual genes but as subnetworks from differentially expressed genes in MS extracted from protein interaction databases. Gene expression differences between MS patients and controls as well as MS patients that have received treatment, of a large data set allowed several significant de-regulated genes to be detected. A proportion of transcripts up-regulated in untreated patients were counter-regulated by IFN treatment, suggesting a set of possible effectors for this first-line therapy in MS. We have followed same steps of the methodology and combined it with the works of Diogo M. Camacho et al., [31], Y Liu et al., 2019 [30], AS Nangraj, 2020 [37], in order to explore the potentials of our data set and investigate the ability to perform drug repositioning based on G Fiscon et al., 2021 [38]. Clustering and classification algorithms have been successfully used to elucidate the functional relationship between genes and pathways. In this context, our goal in this thesis is to implement our methodology into our main transcriptomic dataset and locate the structural differences within the network between the two populations MS untreated patients *versus* Healthy controls and MS Interferon treated *versus* MS untreated patients as well as patients in different stages of the disease such as Relapsing Remitting (RRMS) *versus* Healthy controls, Secondary Progressive (SPMS) *versus* Healthy Controls and Primary Progressive (PPMS) *versus* Healthy controls.

The gene expression profile of each gene differentiates along the samples and according to the group that each sample belongs; the value of each gene alters significantly. Therefore, we aim in finding the genes that most differ between the two groups and are more likely to dominate in our networks. The resulting subnetworks will give us the information we need in order to determine how the genes behave and probably going to behave, as well as how they influence each other so as to have a better knowledge in predicting "disease triggering" relations/pathways.

1.5 Thesis Outline and Innovation

The development of this thesis is based on the necessary theoretical background covered In Chapter 2. In the first part of this Chapter the human genome and biological concepts regarding DNA microarrays are included and form the biological background. Gene networks and methodologies concerning the analysis of DNA microarray data as well as the construction of gene networks compose the second part. Machine learning approaches and the mathematical background involving the knowledge in the field of bioinformatics and its applications is also presented. Chapter 3 introduces the proposed methodology concerning this study and we analyze in detail the steps chosen for the elaboration of our methodology for the gene subnetwork construction, the hub genes discovery, and the steps towards drug repositioning examination. We have also performed an evaluation method implemented for the generalization ability of the observed results. The integration of the Multiple sclerosis gene expression datasets and the methodology is presented in section 4, as well as the generation of PPI networks from our data along with their organization in subnetworks. Our results were evaluated after applying the supervised and unsupervised classification methods in the steps accordingly, for statistical prediction and examination of the biological significance of our results.

In this work the innovative concept involves the process of gene expression data from a combinational pipeline, that to our knowledge, has not been performed on Multiple sclerosis data.
Moreover, taking into account the heterogeneity of the disease as well as the limited sample size, we can safely say that investigating Multiple sclerosis at the molecular level has provided valuable insight, but there is a lot of research in this to be done. The current knowledge for the development of strategies for preventing or predicting the progression of the disease is insufficient, therefore a combination of clinical data and different machine learning techniques must be explored.

2 Theoretical Background

In this Chapter we introduce the reader to the necessary biological background followed by the mathematical background (bioinformatics and machine learning), needed for the composition of this thesis. The human genome is presented in the first section and the significance of DNA microarrays as well as their analysis is covered in section 2.1. Following, in section 2.2, which constitutes the beginning of the second theoretical part, we introduce the scientific field of machine learning and pattern recognition followed by the process of feature subset selection (FSS), applied in DNA microarray data, which is distinguished in three fundamental algorithms, also presented, wrappers, filters and embedded methods, is interpreted in Section 2.3. In sections 2.4 and 2.5, the general process of classification and an introduction of classifiers, including linear and nonlinear classifiers, along with the classification methods Support Vector Machines (SVM) and decision trees, implemented in this thesis, are covered respectively. Furthermore, in section 2.6, different evaluation methods are described such as holdout validation, k-fold cross validation, leave one out cross validation, repeated random sub-sampling validation.

Finally, the relationship of network biology and bioinformatics is introduced in section 2.7 where a part of different biological networks that exist are presented.

2.1 The Human Genome

2.1.1 Genome

The human genome is a complete set of nucleic acid sequences for humans, encoded as the molecule of DNA (deoxyribonucleic acid) within the 23 chromosome pairs in cell nuclei and in a small DNA molecule found within individual mitochondria. These are usually treated separately as the nuclear genome and the mitochondrial genome. Human genomes include both protein-coding DNA genes and noncoding DNA. Haploid human genomes, which are contained in germ cells (the egg and sperm gamete cells created in the meiosis phase of sexual reproduction before fertilization creates a zygote) consist of more than three billion DNA base pairs, while diploid genomes (found in somatic cells) have twice the DNA content. The study, analysis and mapping of HUMAN GENOME, has been the subject of the "Human Genome Project" (www.genome.gov). All living organisms are composed of cells, small units of biological activity.

The discovery that DNA contains the code for life, urged a global effort to understand how the genome sequences of many organisms associated with their health. The study of the human genome led to the genomic revolution since the notification of the first draft sequence of the genome had a huge impact on human cancer research. Genes is the basic physical unit of inheritance. Genes are passed from parents to offspring and contain the information needed to specify traits. Genes are arranged, one after another, on structures called chromosomes. A chromosome contains a single, long DNA molecule, only a portion of which corresponds to a single gene. Humans have approximately 20,000-25,000 genes arranged on their chromosomes. (www.medlineplus.gov)

DNA and RNA

Each gene is made of DNA. Deoxyribonucleic acid (DNA) is the central information storage system of most animals and plants, and even some viruses. The name comes from its structure, which is a sugar and phosphate backbone which have bases sticking out from it--so-called bases. So that "deoxyribo" refers to the sugar and the nucleic acid refers to the phosphate and the bases. The bases go by the names of adenine, cytosine, thymine, and guanine, otherwise known as A, C, T, and G. DNA is a remarkably simple structure. It's a polymer of four bases--A, C, T, and G--but it allows enormous complexity to be encoded by the pattern of those bases, one after another. DNA is organized structurally into chromosomes and then wound around nucleosomes as part of those chromosomes. Functionally, it's organized into genes, of which are pieces of DNA, which lead to observable traits. And those traits come not from the DNA itself, but from the RNA that is made from the DNA, or most commonly of proteins that are made from the RNA which is made from the DNA. So, the central dogma, so-called of molecular biology, is that genes, which are made of DNA, are made into messenger RNAs, which are then made into proteins. But for the most part, the observable traits of eye color or height or one thing or another of individuals come from individual proteins. Sometimes, we're learning in the last few years they come from RNAs themselves without being made into proteins--things like micro RNAs. But those still are relatively the exception for accounting for traits. (www.technologynetworks.com)

As mentioned above RNA is a nucleic acid that is similar in structure to DNA but different in subtle ways. The cell uses RNA for several different tasks, one of which is called messenger RNA, or mRNA. And that is the nucleic acid information molecule that transfers information from the genome into proteins by translation. Another form of RNA is tRNA, or transfer RNA, and these are non-protein encoding RNA molecules that physically carry amino acids to the translation site that allows them to be assembled into chains of proteins in the process of translation. (www.technologynetworks.com)

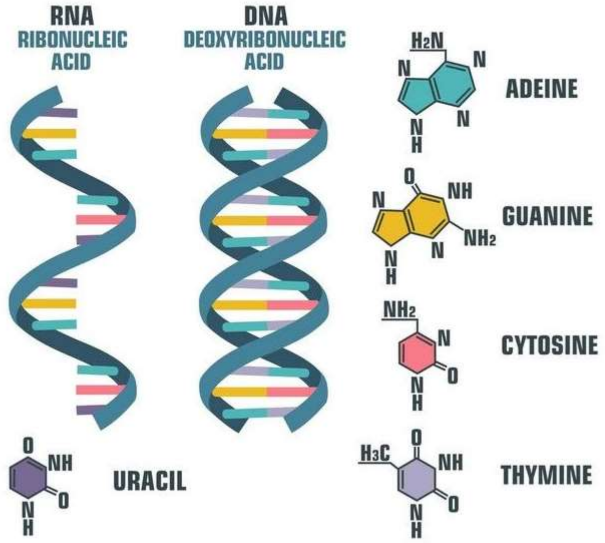


FIGURE 2.1 DNA AND RNA DIFFERENCES

Genes are the blueprint for our bodies. Humans typically have 46 chromosomes in each cell of their body, made up of 22 paired chromosomes and two sex chromosomes. These chromosomes contain between 20,000 and 25,000 genes. New genes are being identified all the time. The paired chromosomes are numbered from 1 to 22 according to size. (Chromosome number 1 is the biggest.) These non-sex chromosomes are called autosomes. People usually have two copies of each chromosome. One copy is inherited from their mother (via the egg) and the other from their father (via the sperm). A sperm and an egg each contain one set of 23 chromosomes. When the sperm fertilises the egg, two copies of each chromosome are present (and therefore two copies of each gene), and so an embryo forms. The chromosomes that determine the sex of the baby (X and Y chromosomes) are called sex chromosomes. Typically, the mother's egg contributes an X chromosome, and the father's sperm provides either an X or a Y chromosome. A person with an XX pairing of sex chromosomes is biologically female, while a person with an XY pairing is biologically male. As well as determining sex, the sex chromosomes carry genes that control other body functions. There are many genes located on the X chromosome, but only a few on the Y chromosome. Genes that are on the X chromosome are said to be X-linked. Genes that are on the Y chromosome are said to be Y-linked.

Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product that enables it to produce end products, protein, or non-coding RNA, and ultimately affect a phenotype, as the final effect. These products are often proteins, but in non-protein-coding genes such as transfer RNA (tRNA) and small nuclear RNA (snRNA), the product is a functional non-coding RNA. Gene expression is summarized in the central dogma of molecular biology first formulated by Francis Crick in 1958, further developed in his 1970 article, [39] and expanded by the subsequent discoveries of reverse transcription and RNA replication. The process of gene expression is used by all known life—eukaryotes (including multicellular organisms), prokaryotes (bacteria and archaea), and utilized by viruses—to generate the macromolecular machinery for life. (en.wikipedia.org/wiki/Gene_expression)

In genetics, gene expression is the most fundamental level at which the genotype gives rise to the phenotype, i.e., observable trait. The genetic information stored in DNA represents the genotype, whereas the phenotype results from the "interpretation" of that information. Such phenotypes are often expressed by the synthesis of proteins that control the organism's structure and development, or that act as enzymes catalyzing specific metabolic pathways. All steps in the gene expression process may be modulated (regulated), including the transcription, RNA splicing, translation, and post-translational modification of a protein. Regulation of gene expression gives control over the timing, location, and amount of a given gene product (protein or ncRNA) present in a cell and can have a profound effect on the cellular structure and function. Regulation of gene expression is the basis for cellular differentiation, development, morphogenesis and the versatility and adaptability of any organism. Gene regulation may therefore serve as a substrate for evolutionary change. (www.basic2tech.com/genetics/)

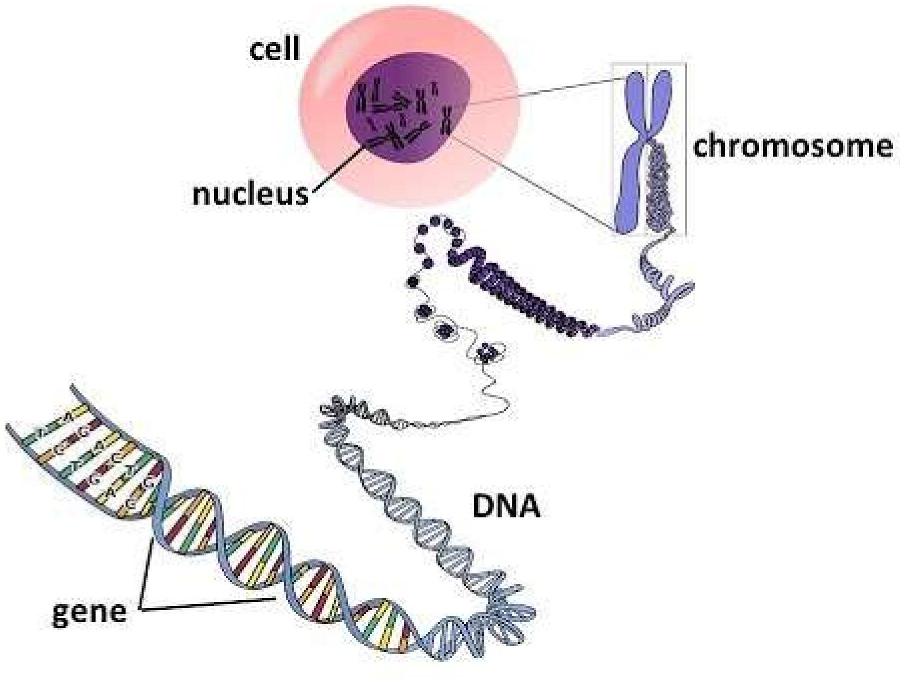


FIGURE 2.2 DNA

2.1.2 Genetics

Genetics is a branch of biology concerned with the study of genes, genetic variation, and heredity in organisms. Though heredity had been observed for millennia, Gregor Mendel, Moravian scientist and Augustinian friar working in the 19th century in Brno, was the first to study genetics scientifically. Mendel studied "trait inheritance", patterns in the way traits are handed down from parents to offspring over time. He observed that organisms (pea plants) inherit traits by way of discrete "units of inheritance". This term, still used today, is a somewhat ambiguous definition of what is referred to as a gene. Trait inheritance and molecular inheritance mechanisms of genes are still primary principles of genetics in the 21st century, but modern genetics has expanded beyond inheritance to studying the function and behavior of genes. Gene structure and function, variation, and distribution are studied within the context of the cell, the organism (e.g., dominance), and within the context of a population. Genetics has given rise to several subfields, including molecular genetics, epigenetics and population genetics. Organisms studied within the broad field span the domains of life (archaea, bacteria, and eukarya). Genetic processes work in combination with an organism's environment and experiences to influence development and behavior, often referred to as nature *versus* nurture. The intracellular or extracellular environment of a living cell or organism may switch gene transcription on or off. A classic example is two seeds of genetically identical corn, one placed in a temperate climate and one in an arid climate (lacking sufficient waterfall or rain). While the average height of the two corn stalks may be genetically determined to be equal, the one in the arid climate only grows to half the height of the one in the temperate climate due to lack of water and nutrients in its environment. (www.basic2tech.com/genetics/)

2.1.3 DNA Microarray and analysis

A microarray is a laboratory tool used to detect the expression of thousands of genes at the same time. DNA microarrays are microscope slides that are printed with thousands of tiny spots in defined positions, with each spot containing a known DNA sequence or gene. Often, these slides are referred to as gene chips or DNA chips. The DNA molecules attached to each slide act as probes to detect gene expression, which is also known as the transcriptome, or the set of messenger RNA (mRNA) transcripts expressed by a group of genes. To perform a microarray analysis, mRNA molecules are typically collected from both an experimental sample and a reference sample. For example, the reference sample could be collected from a healthy individual, and the experimental sample could be collected from an individual with a disease like cancer. The two mRNA samples are then converted into complementary DNA (cDNA), and each sample is labeled with a fluorescent probe of a different color. For instance, the experimental cDNA sample may be labeled with a red fluorescent dye, whereas the reference cDNA may be labeled with a green, fluorescent dye. The two samples are then mixed together and allowed to bind to the microarray slide. The process in which the cDNA molecules bind to the DNA probes on the slide is called hybridization. Following hybridization, the microarray is scanned to measure the expression of each gene printed on the slide. If the expression of a particular gene is higher in the experimental sample than in the reference sample, then the corresponding spot on the microarray appears red. In contrast, if the expression in the experimental sample is lower than in the reference sample, then the spot appears green. Finally, if there is equal expression in the two samples, then the spot appears yellow. The data gathered through microarrays can be used to create gene expression profiles, which show simultaneous changes in the expression of many genes in response to a particular condition or treatment. (www.nature.com/scitable/definition/microarray-202/).

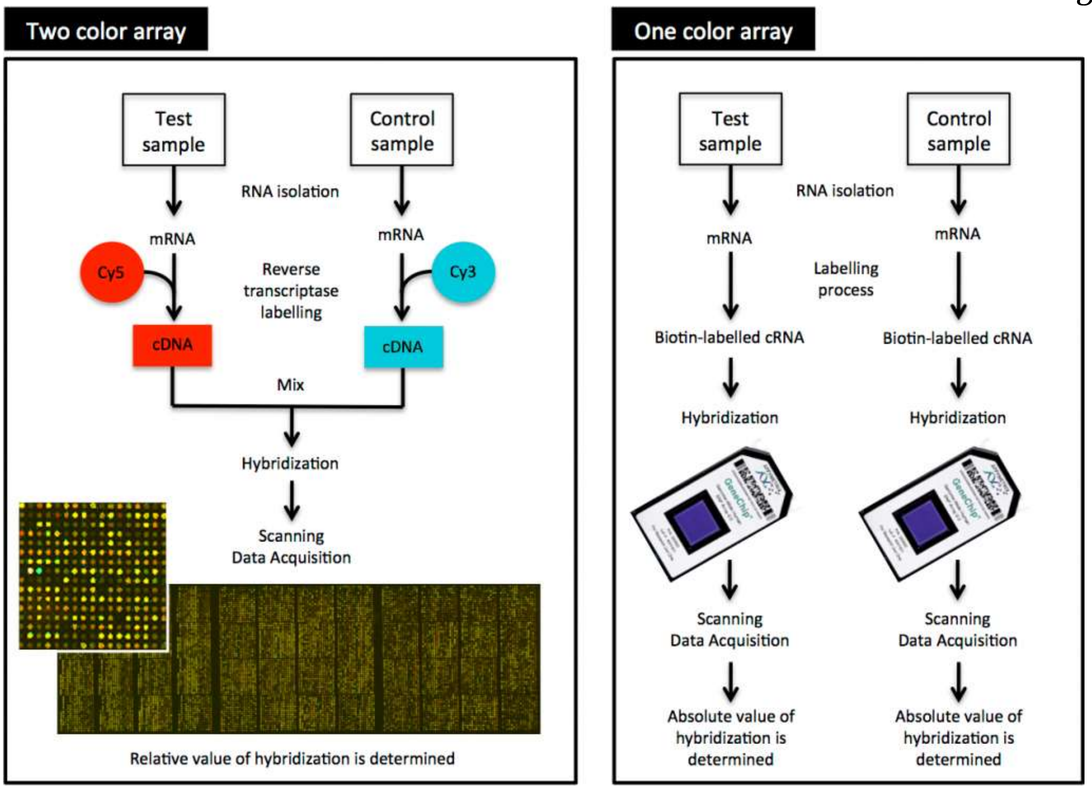


FIGURE 2.3 ONE-COLOR VS TWO-COLOR ARRAYS

Microarray can be a valuable tool in order to define transcriptional signatures bound to a pathological condition, to determine whether the DNA from a particular individual contains a mutation in genes as well as to exclude molecular mechanisms tightly bound to transcription. Microarray analysis frequently does not imply a final answer to a biological problem but allows the discovery of new research paths which let to explore it by a different perspective. (www.genome.gov)

Today, DNA microarrays are used in clinical diagnostic tests for some diseases. Sometimes they are also used to determine which drugs might be best prescribed for certain individuals, because genes determine how our bodies handle the chemistry related to those drugs. With the advent of new DNA sequencing technologies, some of the tests for which microarrays were used in the past now use RNA sequencing instead. But microarray tests still tend to be less expensive than sequencing, so they may be used for very large studies, as well as for some clinical tests. (www.genome.gov)

The principal steps of a microarray analysis are [40]:

Analysis step	**Caveats**
Experimental design and implementation-	• Define the biological question and hypothesis clearly • Design the microarray experimental scheme carefully; include biological replication in experimental design • Avoid experimental errors
Data collection and archival	Compliance with microarray information collection standards (e.g. MIAME)
Image acquisition	• Try to balance the overall intensities between the two dyes • Scan image at appropriate resolution

Analysis step	**Caveats**
Image analysis	• Inspect the gridding result manually; adjust the mask and flag poor-quality spots if necessary • Choose and apply an appropriate segmentation algorithm • Apply quality measures to aid decision of spot quality
Data pre-processing	• Remove poor-quality spots • Remove spots with intensity lower the background plus two standard deviations. • Log-transform the intensity ratios
Data normalization	• Use diagnostic plots to evaluate the data • Consider using LOWESS and its variants for normalization
Identifying differentially expressed genes	• Do not use fixed threshold (i.e. two-fold increase or decrease) to infer significance • Calculate a statistic based on replicate array data for ranking genes • Select a cut-off value for rejecting the null hypothesis that a gene is not differentially expressed; remember to adjust for multiple hypothesis testing
Exploratory data analysis	• Use different analysis tools with different setting to 'explore' the data • Validate the result by follow-up experiments

TABLE2.1 SUMMARY OF MICROARRAY ANALYSIS STEPS

2.2 Machine Learning and Pattern Recognition

2.2.1 Datasets

Data

Data completeness and generalizability are other important considerations when developing and training Machine Learning (ML) algorithms. The familiar concept of "garbage-in/garbage-out" highlights the critical importance of having high-quality data for ML applications, since incomplete and/or erroneous values may inappropriately train an algorithm in the wrong direction. Likewise, highly controlled data may not represent real-world conditions. "Quality data" for AI/ML training applications must include accurate, precise, complete, and generalizable information [43]. Laboratory data are often assumed to be sufficiently accurate and precise by both health-care providers and researchers.

Unfortunately, it is a truism that not all laboratory tests are created equal, and poor analytical bias and imprecision degrade the performance of ML algorithms. Additionally, both providers and researchers are often not aware that test methods may lack standardization. The concept of imprecision reported as coefficient of variation is also poorly understood by most bedside providers with many assuming any change in numerical values reflecting a true biological change without taking into account sources of variability. Despite the convenience of collecting real-world information from electronic health records, the retrieved medical data are often incomplete. This is attributed to the several inconsistencies in test ordering and resulting. Ordered laboratory tests may be cancelled due to patients not showing up for a visit, or samples were found to be not acceptable upon receipt by the laboratory. Incomplete data create significant challenges for ML developers, where the predictive power of algorithms may be severely diminished. The limitation of real-world evidence has thus prompted investigators to gravitate toward more complete and rigorous data derived from clinical trials. However, caution is advised when using data that are “too complete” or “too controlled,” since it may not represent the real-world population and contribute to overfitting [44]. Ultimately, the best and most balanced approach is to pilot ML algorithms using more controlled data during the initial stages and later refining these algorithms using real-world data to confirm generalizability.

Here, our data is presented as a set of N samples. Each sample contains the expression value of K genes also called predictors. In the dataset, each sample N can be expressed as a vector $\boldsymbol{x_i} \in \boldsymbol{R^K}$ where i = 1,...., N. To each of the samples, a class label y is assigned. The data can also be expressed in array form as X ∈ $R^{N,K}$ where each row represents a sample containing the expression values of K genes, while the class labels of all samples are expressed as a vector y ∈ R^N.

Pattern recognition

Pattern recognition [41-42] is classified in the field of machine learning, a scientific area that focuses on the recognition of patterns and regularities in vast amount of data. Today, there is clearly a need to apply rational and systems-based data science principles for handling the ever-growing body of both qualitative and quantitative aspects of medical laboratory information and classification. Faced with the limitations of human processing of rapid, accurate, and precise retrieval of data in real time, the heuristic provided and amplified by Machine Learning offers an attractive approach to substantially improve the delivery of health care. Current health problems that are deemed suitable to ML include, but are not limited to, integrating multiple variables to mimic human clinical decision-making skills (eg, multiparameter disease diagnosis), automation of testing and treatment algorithms (eg, reflex testing) and workflows, pattern recognition using imaging data (eg, radiology, histology slides, and vital sign waveforms), and/or test utilization trends. However, although one could use AI/ML, it may not always be necessary to apply such tools for every situation since simple statistical approaches may sometimes suffice[41].

➢ Supervised learning

Supervised learning entails learning a mapping between a known dataset called the training dataset, a set of input variables X and an output variable Y and applying this mapping to predict the outputs for unseen data. If the desired output consists of continuous variables, then the task is called regression whereas cases, in which the output falls within discrete values the task is called classification. Supervised learning is the most important methodology in machine learning and it also has a central importance in the processing of class prediction in DNA microarray data analysis. (ex. linear regression, logistic regression, naive Bayes, decision tree, k-nearest neighbor (k-NN), support vector machine (SVM), and the ensemble decision tree algorithm random forest (RF)).

➢ Unsupervised learning

Unsupervised learning is the type of machine learning that is trying to find hidden structure in data with unlabeled responses. Due to the fact that the data given are unlabeled, this concludes that there is no error or reward signal to evaluate a potential solution. Various unsupervised classification techniques can be employed with DNA microarray data in microarray data analysis that affect statistical analysis, in the part of class discovery. (ex. k-means algorithm, principal component analysis (PCA), hierarchical clustering).

- **Reinforcement learning**

Reinforcement learning is the type of machine learning where an agent interacts with its environment. The agent senses the environment and based on this sensory input choosing an action to perform in it. This action changes the environment in some manner and this change is communicated to the agent through a scalar reinforcement signal. Reinforcement learning utilizes a positive or negative reward signal sent to the agent after an action is complete (ex. International Business Machine (IBM)'s Deep Blue (Armonk, New York) and Google's Go (Alphabet, Mountain View, California)). Currently reinforcement learning approaches are rarely employed in pathology.

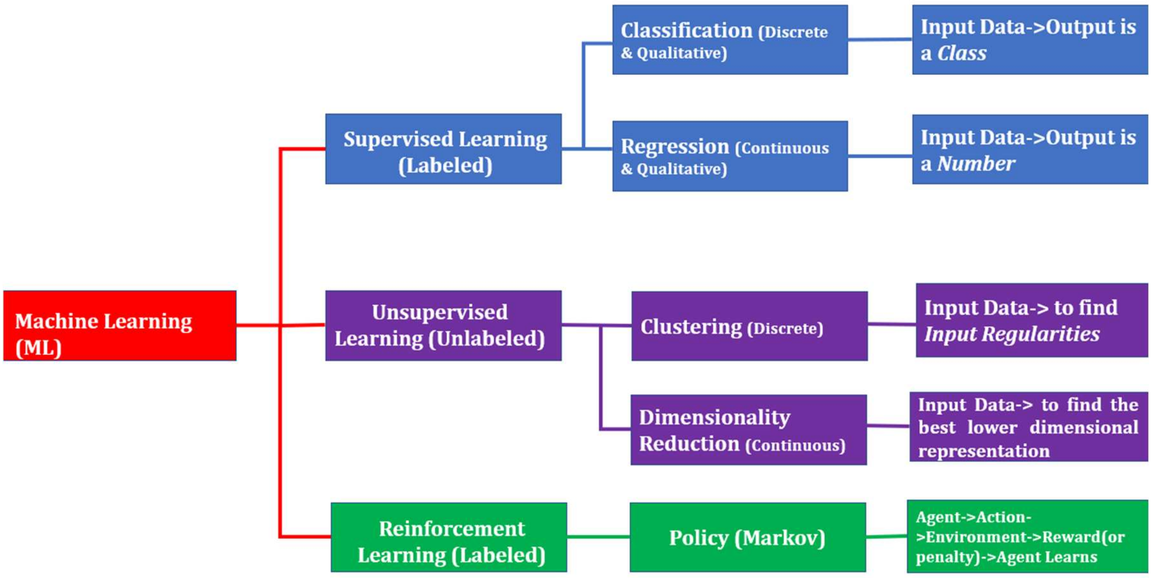

Machine Learning (ML)
Supervised Learning (Labeled)
Classification (Discrete & Qualitative)
Input Data->Output is a *Class*
Regression (Continuous & Qualitative)
Input Data->Output is a *Number*
Unsupervised Learning (Unlabeled)
Clustering (Discrete)
Input Data-> to find *Input Regularities*
Dimensionality Reduction (Continuous)
Input Data-> to find the best lower dimensional representation
Reinforcement Learning (Labeled)
Policy (Markov)
Agent->Action->Environment->Reward(or penalty)->Agent Learns

FIGURE 2.4 OVERVIEW DIAGRAM OF MACHINE LEARNING ALGORITHMS

2.2.2 Patterns –Classes – Features

Machine learning starts with the design of appropriate data representations. In machine learning and pattern recognition the features can be symbolic (e.g. condition) or numerically (e.g. weight). The combination of some features is the *feature vector*. A *pattern* is a composition of characteristics which are divided into specific decision areas called *classes*. The classes are separated by decision boundaries. The n-dimensional space defined by the feature vector space is called feature space. Feature spaces may overlap each other, allowing patterns of different classes to share same characteristics. Moreover, each pattern can be illustrated in the set of features F. Thus, each feature can be a member not only of different patterns but also different classes. The classification model is a pair of variables {x, ω} where x is a collection of features, feature vector, and ω is the concept of observation, the label [45-46].

2.2.3 Applications and implementation of pattern recognition

Pattern recognition as a field of study developed significantly in the 1960s. It is an interdisciplinary subject, covering developments in the areas of medical, engineering, artificial intelligence, computer science, psychology and physiology, among others. Human being has natural intelligence and so can recognize patterns [47-48]. As we mentioned above pattern recognition is the study of how machines can observe the environment, learn to distinguish patterns of interest from their background, and make sound and reasonable decisions about the patterns [49]. But in spite of almost 50 years of research, design of a general-purpose machine pattern recognizer remains an elusive goal. The best pattern recognizers in most instances are humans, yet we do not understand how humans recognize patterns. Given a pattern, its recognition/classification may consist of one of the following two tasks:

- supervised classification in which the input pattern is identified as a member of a predefined class,
- unsupervised classification (e.g., clustering) in which the pattern is assigned to an unknown class.

The steps that take place in a pattern recognition task are:

1. **Data acquisition.** Through data acquisition the data are converted from one form (speech, character, pictures etc.) into another in order to be acceptable to the computing device.
2. **Preprocessing and Feature extraction**. After data acquisition the task of analysis begins. Where the learning about the data takes place and information is collected about the different events and pattern classes available in the data.
3. **Classification.** Its purpose is to decide the category of new data on the basis of knowledge received from data analysis process. Classifier is the algorithm that implements classification and maps input data to class which performs classification. Finally, it is ought to evaluate the decision taken. This involves applying the trained classifier to an independent test set of labeled patterns.

System learns from training set and efficiency of system is checked by presenting testing set to it.

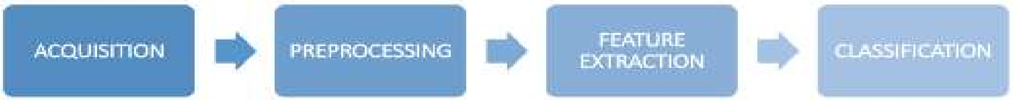


FIGURE 2.5 PATTERN RECOGNITION PROCESS

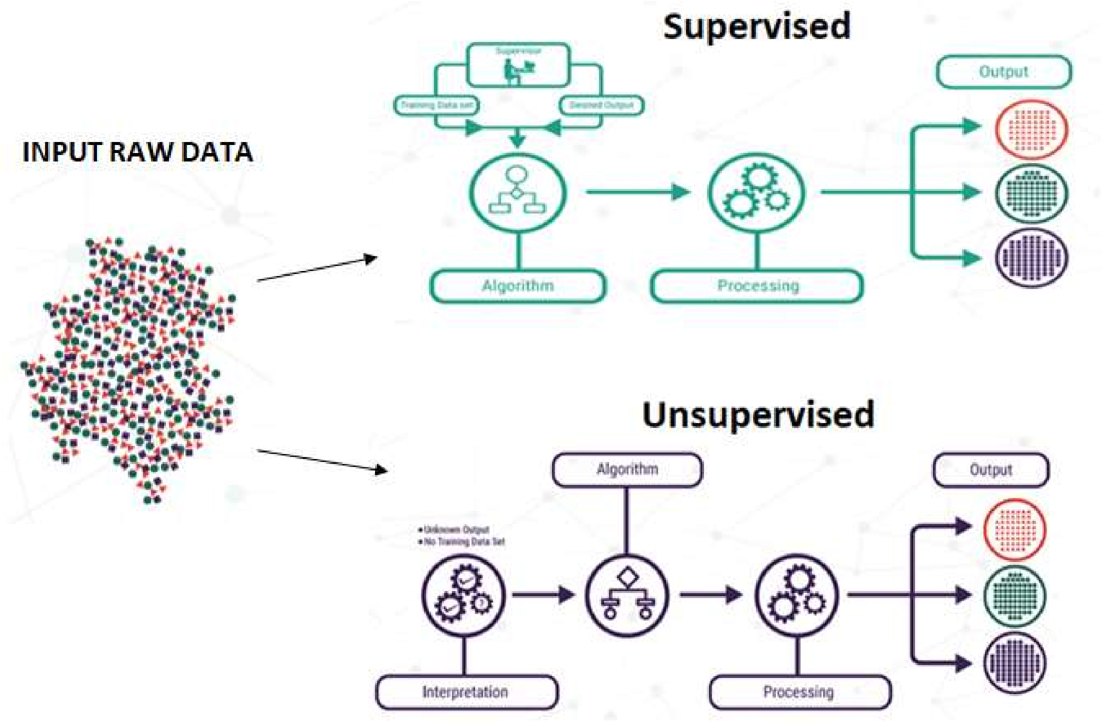


FIGURE 2.6 PATTERN RECOGNITION TASKS

2.3 Processing Features

2.3.1 Feature pre-processing

In a typical high throughput experiment, we assay thousands of features (gene transcripts, proteins, metabolites) in a certain number of biologically diverse samples (from about 6 to hundreds or thousands). In biomedical research, experiments aim at measuring biological variability by comparing two or more biological conditions in a controlled setting. To be able to measure any biologically signal in the data, the biological variability of interest, i.e., the one produced by the treatment, must be larger than the technical variability. However, before analyzing any data, it is necessary to make the samples as comparable as possible by removing the unwanted technical variability that should be shared among all samples without removing biological variability, that will differentiate the samples biologically. [49-50]

The steps that can be followed are:

- **Data transformation**

The first step in preparing a dataset is to visualize the distribution of the values. Very often dew to the skewness of their distribution we see that most of the data are at very low values with some very high values. Such data are difficult to visualize and to analyze, therefore we log-transform the data.

- **Normalization**

In order to remove as much as technical variability as possible while keeping biological variability of the data, it is necessary to further process them through normalization. One of the important requirements of most normalization techniques is that most features aren't expected to change among biological conditions thus normalization expects only a minority of biological features to be differentially expressed in the conditions of interest. Normalization techniques are:

1. Centering

Centering refers to the operation of modifying the mean value of a set of values by subtracting a fixed value from each individual value. A typical value is the mean of all the data to be centered. The reasons for centering are quite subjective and qualitative. It is possible to formulate rational reasons for centering on scientific grounds. Basically, centering should be performed only if there are common offsets in the data or if modeling such offsets provides an approximately reasonable model. Thus centering is performed to make interval-scale data behave as ratio-scale data, which is the type of data assumed in most multivariate models. Said more simply, centering should make a difference. This difference can manifest itself as:

(i) reduced rank of the model

(ii) increased fit to the data

(iii) specific removal of offsets

(iv) avoidance of numerical problems.

2. Scaling

Scaling refers to the operation of rescaling a set of values to scale in the range of 0 and 1 (or -1 and 1). Scaling is a subject often treated in conjunction with centering. Scaling is used for several reasons. Some important ones are:

(i) to adjust scale differences.

(ii) to accommodate for heteroscedasticity.

(iii) to allow for different sizes of subsets of data (block scaling)

However, the purpose of scaling is very different from that of centering. Scaling is a way of introducing a loss function other than the least squares loss function normally used, therefore scaling does not change the interpretation of the model and its parameters. As for centering, scaling must be performed in a specific way in order not to introduce artificial structure that needs to be modeled. This becomes even more apparent when going to three-way models.[50]

3. Quantile normalization

Quantile normalization is a non-parametric normalization method. The goal of the quantile method is to make the distribution of probe intensities for each array in a set of arrays the same. The method is motivated by the idea that a quantile–quantile plot shows that the distribution of two data vectors is the same if the plot is a straight diagonal line and not the same if it is other than a diagonal line. This concept is extended to n dimensions so that if all n data vectors have the same distribution, then plotting the quantiles in n dimensions gives a straight line along the line given by the unit vector ($\frac{1}{\sqrt{n}}$ .... $\frac{1}{\sqrt{n}}$). This suggests we could make a set of data have the same distribution if we project the points of our n dimensional quantile plot onto the diagonal.

Let q_k = $(q_{k1},..., q_{kn})$ for k = 1,..., p be the vector of the kth quantiles for all n arrays q_k = $(q_{k1},..., q_{kn})$ and d= $(\frac{1}{\sqrt{n}} .... \frac{1}{\sqrt{n}})$ be the unit diagonal. To transform from the quantiles so that they all lie along the diagonal, consider the projection of q onto d

$$proj_{dq_k} = \left(\frac{1}{n} \sum_{j=1}^{n} q_{kj}\,, \dots, \frac{1}{n} \sum_{j=1}^{n} q_{kj} \right)$$

This implies that we can give each array the same distribution by taking the mean quantile and substituting it as the value of the data item in the original dataset. This

motivates the following algorithm for normalizing a set ofdata vectors by giving them the same distribution:

1. given n arrays of length p, form X of dimension p × n where each array is a column;

2. sort each column of X to give X_{sort};

3. take the means across rows of X_{sort} and assign this mean to each element in the row to get X'_{sort};

4. get $X_{normalized}$ by rearranging each column of $X'_{sort}$to have the same ordering as original X

The quantile normalization method is a specific case of the transformation $x'_i = F^{-1}(G(x_i))$, where we estimate G by the empirical distribution of each array and F using the empirical distribution of the averaged sample quantiles. Extensions of the method could be implemented where $F^{-1}$and G are more smoothly estimated. One possible problem with this method is that it forces the values of quantiles to be equal. This would be most problematic in the tails where it is possible that a probe could have the same value across all the arrays. However, in practice, since probeset expression measures are typically computed using the value of multiple probes, we have not found this to be a problem [50].

2.3.2 Feature extraction

Feature extraction addresses the problem of finding the most compact and informative set of features, to improve the efficiency or data storage and processing. Defining feature vectors remains the most common and convenient means of data representation for classification and regression problems. Data can then be stored in simple tables (lines representing "entries", "data points, "samples", or "patterns", and columns representing "features"). Each feature results from a quantitative or qualitative measurement, it is an "attribute" or a "variable". Modern feature extraction methodology is driven by the size of the data tables, which is ever increasing as data storage becomes more and more efficient [51].

Dimensionality reduction is an important approach in machine learning. To identify the set of significant features and to reduce the dimension of the dataset, there are three popular dimensionality reduction techniques that are used.

- **Principal Component Analysis (PCA)**

Principal Component Analysis (PCA) is the main linear approach for dimensionality reduction. It performs a linear mapping of the data from a higher-dimensional space to a lower-dimensional space in such a manner that the variance of the data in the low-dimensional representation is maximized.

- **Kernel PCA (KPCA)**

Kernel Principal Component Analysis (KPCA) is an extension of PCA that is applied in non-linear applications by means of the kernel trick. It is capable of constructing nonlinear mappings that maximize the variance in the data.

2.3.3 Feature Subset Selection (FSS)

When building a machine learning model in real-life, it's almost rare that all the variables in the dataset are useful to build a model. Adding redundant variables reduces the generalization capability of the model and may also reduce the overall accuracy of a classifier. Furthermore, adding more and more variables to a model increases the overall complexity of the model. The goal of feature selection in machine learning is to find the best set of features that allows one to build useful models of studied phenomena.

There are three important reasons why we choose *Feature Selection* and not just give all the features to the ML algorithm and let it decide which feature is important. The first reason is the Curse of dimensionality — Overfitting. As the dimensionality of the feature space increases, the number configurations can grow exponentially and thus the number of configurations covered by an observation decreases. The second reason is that we want our models to be simple and explainable. We lose ability to explain our models properties when we have a lot of features. Finally, most of the times, we will have many non-informative features. For example, Name or ID variables. Poor-quality input will produce Poor-Quality output. Also, a large number of features make a model bulky, time-taking, and harder to implement in production.

In supervised learning, feature selection is often viewed as a search problem in a space of feature subsets. To carry out this search we must specify a starting point, a strategy to traverse the space of subsets, an evaluation function and a stopping criterion. Depending on how and when the utility of selected characteristics is evaluated, different methods may be adopted which are divided into the following categories: [52-53]

1. Filter methods
2. Wrapper methods
3. Embedded methods

Filter methods

Filter approaches [52, 53] remove irrelevant features according to general characteristics of the data. Filter algorithms provide fast execution, since they do not include repetitions and they are not based on a specific classifier. They have a simple construction, which typically uses a simple search strategy and characteristics evaluation criterion is planned based on a specific criterion, the feature/feature subset relevance. In this method for every possible characteristics combination, we choose a criterion (e.g. Bhattacharya distance, Divergence, Scatter Matrices) and select the best combination of features vector. We must note that filter algorithms are relatively robust against overfitting and may fail to select the most “useful” features. The primary advantage of filter methods is their speed and ability to scale, to large datasets.

Filter methods are divided into *multivariate* and *univariate* methods. Multivariate methods are able to find relationships among the features, while univariate methods consider each feature separately. Univariate filter techniques can be divided into two categories: *parametric* and *model-free* methods. In parametric methods the data is drawn from a given probability distribution while in model-free methods, or non-parametric, the data may not follow a normal distribution. In microarray studies the most widely used techniques are t-test and ANOVA.

A typical feature selection process involves two phases:

- Selection of characteristics and
- Fitting the model to evaluate performance.

It consists of three steps:

1. The first step is the creation of a candidate set which contains a subset of the original features through certain research strategies. Some of the feature selection techniques are:

- **Chi-square Test:**

The Chi-square test is used for categorical features in a dataset. We calculate Chi-square between each feature and the target and select the desired number of features with the best Chi-square scores. In order to correctly apply the chi-squared in order to test the relation between various features in the dataset and the target variable, the following conditions have to be met: the variables have to be categorical, sampled independently and values should have an expected frequency greater than 5.

- **Control cases: t-test**

The basic idea in the t-test is to check if the mean value of the attribute of each class differs significantly from another. T-test is the most popular option when the data follow a normal distribution.

The aim is to check which of the following two cases applies:
H1: The feature has a different average value in each class
H0: The feature has the same average in each class
If H0 (null hypothesis) is applied, then feature is discarded because it is difficult on this basis to distinguish data into categories. On the contrary if H1 (alternative hypothesis) is applicable, the attribute values differ considerably between categories and can be distinguished easily. This feature is selected.

- **The Receiver Operating Characteristic (ROC) curve**

If when applying the previous method, the respective average values are close, the information may not be sufficient to guarantee good properties classification. The ROC technique gives information on the overlap between categories after quantifying an area defined by two curves.

- **Fisher Discrimination Ratio**

In order to quantify the resolution of a feature Fisher Discrimination Ratio is used. The ratio is independent of the distribution followed by the class and defined as:

$$\sum\nolimits_{w} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \overline{x_i}) \, (x_{ij} - \overline{x_i})^T$$

These criteria do not take into consideration the correlations between features and also do not exploit the cross- correlation coefficient between them. In the scalar selection of characteristics, after choosing a criterion is needed to prioritize features in descending order and calculate the cross-correlation of the first in hierarchy, with all the rest. The cross-correlation process may affect significantly the hierarchy of features.

Additionally, in feature selection a high-dimensional generalization scheme which maximizes the mutual information between the joint distribution and other target variables is found to be useful.

The mutual information (MI) of two discrete random variables X and Y is defined as:

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

, where $p(x,y)$ is the joint probability distribution function of X and Y, and p(x) and p(y) are the marginal probability distribution functions of X and Y respectively. In the case of continuous random variables, the summation is replaced by a definite double integral

$$I(X;Y) = \int_Y \int_X p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) dx\,dy$$

, where $p(x,y)$ is now the joint probability density function of X and Y, and are the marginal probability density functions of X and Y respectively.

Mutual information measures the information that X and Y share. Thus, this can be translated as a measurement of the “knowledge” one of these variables gives us, to reduce uncertainty about the other. In the case that X and Y are independent, then knowing X does not give any information about Y and vice versa, so their mutual information is zero. On the other hand, if X is a deterministic function of Y and Y is a deterministic function of X then all information conveyed by X is shared with Y:
Knowing X determines the value of Y and vice versa. As a result, in this case the mutual information is the same as the uncertainty contained in Y (or X) alone, namely the entropy of Y (or X). Moreover, this mutual information is the same as the entropy of X and as the entropy of Y, with a very special case of this is when X and Y are the same random variable.

Mutual information is a measure of the inherent dependence expressed in the joint distribution of X and Y relative to the joint distribution of X and Y under the assumption of independence. Mutual information therefore measures dependence in the following sense:

(X; Y) = 0 if and only if X and Y are independent random variables. this is easy to see in one direction: if X and Y are independent, then $p(x,y) = p(x)\,p(y)$, and therefore:

$$\log\left(\frac{p(x,y)}{p(x)p(y)}\right) = \log 1 = 0$$

Moreover, mutual information is nonnegative I(X; Y) ≥ 0 and symmetric $I(X;Y) = I(Y;X)$.

- **Correlation Coefficient**

Correlation is a measure of the linear relationship of 2 or more variables. Through correlation, we can predict one variable from the other. The logic behind using correlation for feature selection is that the good variables are highly correlated with the target. Furthermore, variables should be correlated with the target but should be uncorrelated among themselves. If two variables are correlated, we can predict one from the other. Therefore, if two features are correlated, the model only really needs one of them, as the second one does not add additional information.

- **Variance Threshold**

The variance threshold is a simple baseline approach to feature selection. It removes all features which variance doesn't meet some threshold. By default, it removes all zero-variance features, i.e., features that have the same value in all samples. We assume that features with a higher variance may contain more useful information but note that we are not taking the relationship between feature variables or feature and target variables into account, which is one of the drawbacks of filter methods.

Continuing on, with the steps needed to create a candidate set of features the second step is the evaluation of the candidate set and assess the usefulness of characteristics in the set. Based on the assessment, some features in the candidate set may be rejected or added to selected set of features.

Finally, the last step is to determine whether the current set of selected features is quite good after applying certain switching criteria. If the set meets the prerequisites, a selection algorithm characteristics will return all the selected features, otherwise, it will be repeated until the stop criterion is satisfied. [53]

Significance Analysis of Microarrays (SAM)

Significance Analysis of Microarrays (SAM) [54-55] is a filter, univariate, statistical technique for finding significant genes in a set of microarray data. It was proposed by Tusher, Tibshirani and Chu and the software was written by Michael Seo, Balasubramanian Narasimhan and Robert Tibshirani. SAM identifies genes with statistically significant changes in expression by assimilating a set of gene-specific tests. Each gene is assigned a score on the basis of its change in gene expression relative to the standard deviation of repeated measurements for that gene. Genes with scores greater than a threshold are chosen as potentially significant. The percentage of such genes identified by chance is the false discovery rate (FDR). To estimate the FDR, nonsense genes are identified by analyzing permutations of the measurements. The threshold can be adjusted to identify smaller or larger sets of genes, and FDRs are calculated for each set. The cutoff for significance is determined by a tuning parameter delta, chosen by the user based on the false positive rate. One can also choose a fold change parameter, to ensure that called genes change at least a pre-specified.

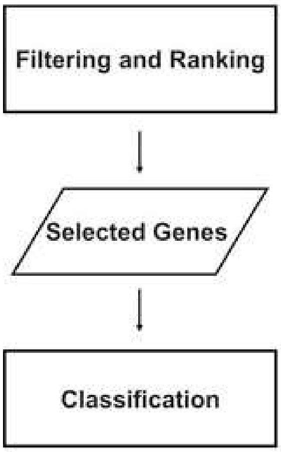


FIGURE 2.7 FILTER PROCESS

Wrapper methods

Wrapper approaches [51] apply machine learning algorithms to feature subsets and use cross-validation to evaluate the score of feature subsets. Wrapper methodology provides a way to resolve the problem of choice characteristics independent of the learning engine that we have chosen. For each combination of feature vectors to estimate the possibility of false classification is estimated and choose based on the lower smallest error. Wrapper feature selection methods create many models with different subsets of input features and select those features that result in the best performing model according to a performance metric. These methods are unconcerned with the variable types, although they can be computationally expensive. In this method the criterion that is used is the feature subset "usefulness" measurement. Finally, we must mention that wrapper methods, in principle, result in the most "useful" features, contrary to filter methods which are prone to overfitting. The main disadvantage of wrapper approaches is that during the feature selection process, the classifier must be repeatedly called to evaluate a subset.

Some of the wrapper selection techniques are:

- **Forward Feature Selection**

This is an iterative method wherein we start with the best performing variable against the target. Next, we select another variable that gives the best performance in combination with the first selected variable. This process continues until the preset criterion is achieved.

- **Backward Feature Elimination**

This method works exactly opposite to the Forward Feature Selection method. Here, we start with all the features available and build a model. Next, the variable from the model which gives the best evaluation measure value is chosen. This process is continued until the preset criterion is achieved.

- **Exhaustive Feature Selection**

This is the most robust feature selection method covered so far. This is a brute-force evaluation of each feature subset. This means that it tries every possible combination of the variables and returns the best performing subset.

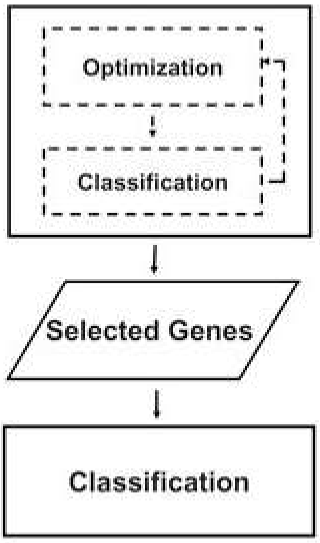


FIGURE 2.8 WRAPPER PROCESS

Embedded methods

The embedded model algorithms [51-53] incorporate the feature selection as part of the training/ load process model, and the utility of the characteristics is obtained by optimizing the function of the learning model. This method does not separate the training data in the training dataset and in a set of validation data. Embedded methods are similar to wrappers, they use the same criterion features subset usefulness. Their advantage is that they are less computationally expensive and less prone to overfitting. Some of the embedded selection techniques are:

- **Recursive Feature Elimination (RFE)**

Recursive feature elimination is an embedded feature selection approach based on the idea to repeatedly construct a model, for example an SVM or a regression model, and choose the best or worst performing feature, for example based on coefficients, setting the feature aside and then repeating the process with the rest of the features. This process is applied until all features in the dataset are exhausted. Features are ranked according to when they were eliminated. As such, it is a greedy optimization for finding the best performing subset of features. The least significant feature is determined through a feature weighting scheme which can be the weight given to each feature by a linear classifier or by non-linear feature weighting methods.

- **LASSO Regularization (L1)**

Regularization consists of adding a penalty to the different parameters of the machine learning model to reduce the freedom of the model, i.e., to avoid over-fitting. In linear model regularization, the penalty is applied over the coefficients that multiply each of the predictors. From the different types of regularization, Lasso or L1 has the property that is able to shrink some of the coefficients to zero. Therefore, that feature can be removed from the model.

- **Random Forest Importance**

Random Forests is a kind of a Bagging Algorithm that aggregates a specified number of decision trees. The tree-based strategies used by random forests naturally rank by how well they improve the purity of the node, or in other words a decrease in the impurity (Gini impurity) over all trees. Nodes with the greatest decrease in impurity happen at the start of the trees, while notes with the least decrease in impurity occur at the end of trees. Thus, by pruning trees below a particular node, we can create a subset of the most important features.

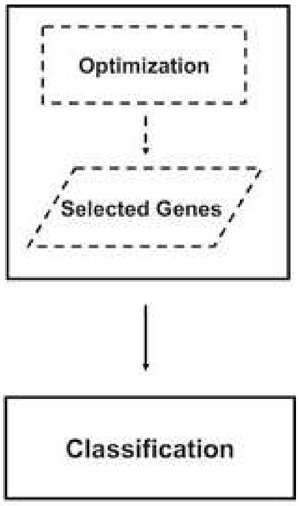


FIGURE 2.9 EMBEDDED METHOD PROCESS

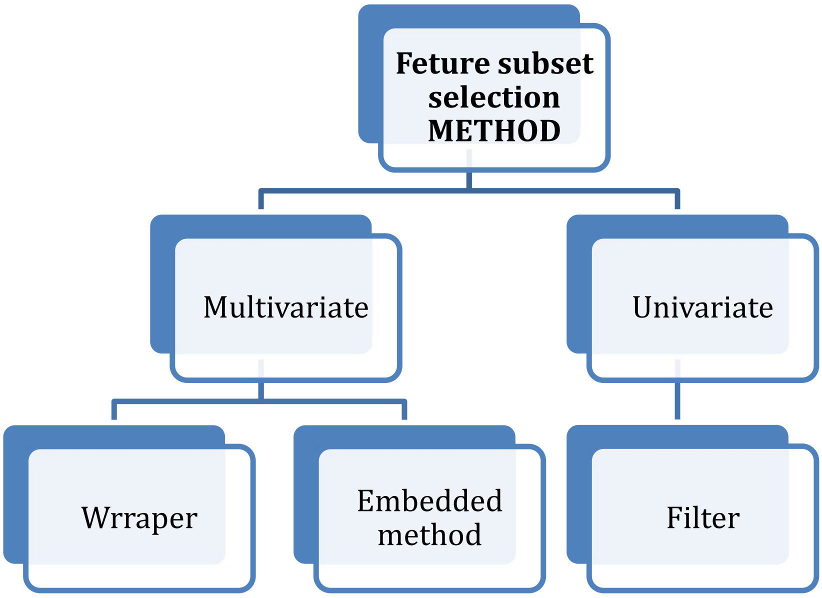


FIGURE 2.10 FEATURE SUBSET SELECTION METHODS

2.4 Clustering

2.4.1 Clustering analysis and methods

Clustering analysis is a type of unsupervised learning which aims to find the most natural way of grouping a dataset. This is achieved by organizing a set of observations based on a similarity criterion, such that observations in the same group are more alike than observations in different groups [56-57]. Many gene clustering methods have been proposed and applied in the literature. Hierarchical clustering [57], K-means [58], partitioning around medoids (PAM; a.k.a. K-memoids) [59], self-organizing maps (SOM) [60] are traditional algorithms and are among the most popular ones in microarray analysis.

Hierarchical clustering

Hierarchical clustering is the first method used to cluster genes and samples in microarray data. It starts by considering the *n* data points as *n* nodes. Instead of partitioning into several clusters, at each iterative stage, a pair of nodes with the shortest distance between them are agglomerated to form a new node (agglomerative method) or the *n* nodes are successively separated into finer groups (divisive method). Thus, a hierarchical tree is constructed after *n-1* steps. In this paper we only consider agglomerative hierarchical clustering. To define distance between two nodes, different linkages including single linkage (shortest pair-wise distance), complete linkage (largest distance), or average linkage (average distance) may be chosen in the method. Hierarchical clustering has been widely used in clustering microarray data and is especially successful in ordering genes to visualize the global patterns. The method, however, suffers from some intrinsic difficulties. At each iterative stage, the merge of two nodes is based on pair-wise distances of all nodes at that stage instead of any global criterion. When *n* is large, accumulation of mistakes is pronounced, and the method lacks robustness. The method by nature forms a hierarchical tree and does not require estimation of the number of clusters. It is, however, possible to generate clusters by cutting the tree at a pre-determined level of branch.[57]

K-means

This is a classical clustering method [58] also widely used in microarray data. The algorithm aims to split the data into K clusters by minimizing the within cluster dispersion $\sum_{j=1}^{K}\sum_{x_i\in C_j}\left|x^i - x^{-(j)}\right|^2$where $x^{-(j)}$is the centre of cluster j and ||.|| denotes Euclidian distance. The optimization is usually implemented by a classification EM-type algorithm that very often falls into a local minimum in a complex data. As a result, the clustering may differ using different initial values in the optimization. One common way to avoid such local minimum problem is to run K-means algorithm multiple times with random initial cluster centers and select the cluster solution with smallest within cluster sum of squares. As an algorithm of global criterion, *K*-means usually produces good clustering results if *K* correctly. chosen. The method is, however, unstable and highly affected by the presence of scattered genes in the complex microarray data. In addition, since *K*-means calculates the cluster centers in the criterion, it requires the data be in the Euclidean space with Euclidean distance as the dissimilarity measure.

SOM

Self-organizing maps (SOM) [60] has been applied in many microarray analysis. It first maps *K* nodes in a low-dimensional (usually two-dimensional) grid space from the *d*-dimensional space that the data set is situated and then the nodes are adjusted iteratively. Each time, a point from the data is randomly chosen. The movement of the nodes in *d*-dimensional space depends on their distance to the chosen point and the two-dimensional geometry of the nodes. The magnitude of movement decreases as iterations goes on. Usually, the process continues more than 20,000 iterations for the nodes to converge and serve as cluster centers to form clustering. Essentially SOM can be viewed as a K-means criterion restricted on the two-dimensional grid geometry. Thus, clusters generated from nodes close to each other in the two-dimensional grid geometry will have similar expression patterns. We not only can visualize expression patterns within each cluster but also can observe relation and connections between clusters on the two-dimensional node space. On the other hand, SOM (compared to *K*-means) is a sub-optimal algorithm because the optimization is restricted on the two-dimensional node space. Similar to *K*-means, SOM is very sensitive to the choice of the number of nodes and the presence of scattered genes.

2.5 Classification

2.5.1 Classification analysis and Classifiers

As we already mentioned the aim of classification is to find a rule, which, based on external observations, assigns a sample to one of several classes, which implements training a classifier to accurately recognize patterns from given training samples and to classify test samples with the trained classifier. Binary classification is the simplest case where the classifier categorizes the samples of given set into two different classes based on that rule.

Classifier is the algorithm that implements classification and maps input data to class which performs classification. Classifiers are divided to linear and nonlinear (Linear and nonlinear classifiers" Online. Available: http://cs.brown.edu/courses/cs1955/fall2009/docs/lecture_10-27.pdf)

Linear and non-Linear Classifier

A linear classifier can separate two classes only, when they are linearly separable, i.e. there exists a hyperplane, in two-dimensional case just a straight line, that separates the data points in both classes. An opposite case is that classes are linearly inseparable. In this case it is still possible that only few data points are in the wrong side of a hyperplane, and thus the error in assuming a linear boundary is small. Depending on the degree of error, linear classifier can still be preferable, because the resulting model is simpler and thus less sensitive for overfitting (poor generalization ability to new data points). However, some classes can be separated only by a non-linear boundary and we need a nonlinear classifier.

More precisely: Let's have numeric attributes $x_{1,........}, x_k$ whose values are denoted by $dom(x_i)$. For example if $x_1$can have values between $0 \leq x \leq 1$, then $dom(x_i) = [0,1]$. These compose attribute space: $dom(x_1) \times dom(x_2) \times \ldots \times dom(x_k)$.

All data points lie somewhere in this space. If the points fall into two classes, there is some boundary which separates them. If the classes are linearly separable, then in two-dimensional case we can describe the boundary by a line, for 3-dimensional data we need a plane and for higher dimensional data a hyperplane. One way to define this hyperplane is a discriminant function$f(x_1, \ldots\ldots, x_k)$, which is 0 on the plane, positive, when $(x_1, \ldots\ldots, x_k)$ belongs to class 1, and negative otherwise. The discriminant function is linear i.e.

$$f = a_1x_1 + a_2x_2 + \ldots + a_kx_k + b$$

The simplest example of non-linear boundary is exclusive-or function of two attributes: $XOR(x_1, x_2) = 1$, if x_1 is true or x_2 is true, but not both.

However, if we map the datapoints to higher dimensional attribute space, it becomes possible to separate the classes by a hyperplane.

In this study, the linear classifier that is implemented is linear Support Vector Machine (SVM). Other examples of linear classifiers are RLS methods like RR and the LASSO, as well as RVM. An example of a nonlinear classifier is K Nearest Neighbor (K-NN) Classifier which classifies new samples depending on a set of samples closest to them, which are called their "nearest neighbors". (www.en.wikipedia.org/wiki/K-nearest_neighbors_algorithm)

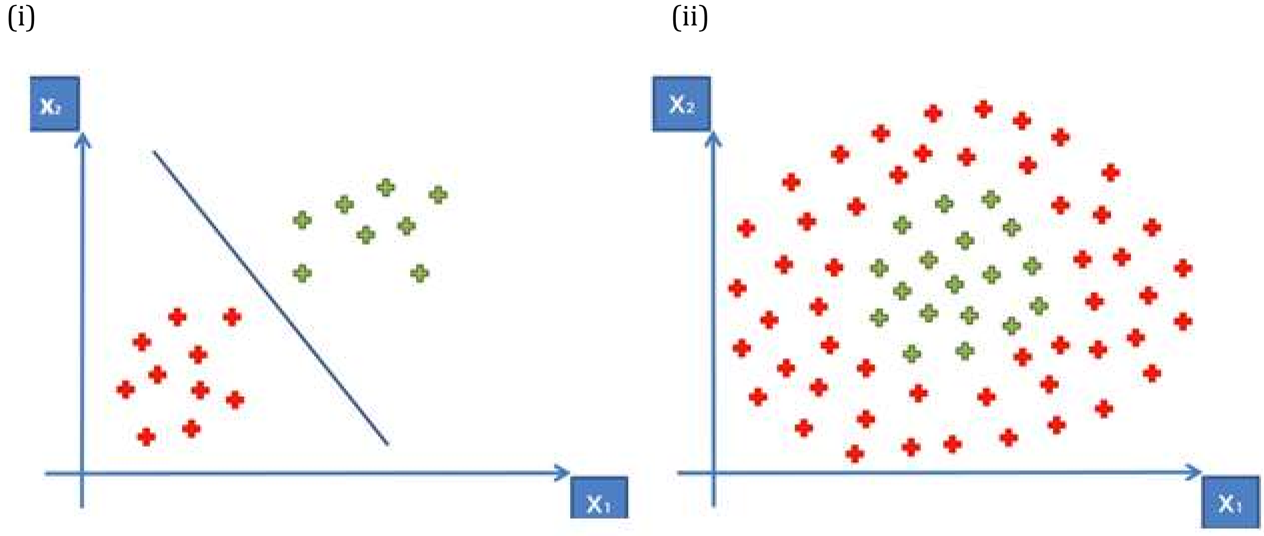


Figure 2.11 Linear (i) and non-linear (ii) problems

Support Vector Machines (SVM)

Support Vector Machines [61] are supervised learning methods used for classification and regression tasks that originated from statistical theory. SVM is a suitable algorithm to deal with interaction among features and redundant features. The advantage of Support Vector Machines is that they can make use of certain kernels to transform the problem, such that we can apply linear classification techniques to non-linear data. Applying the kernel equations arranges the data instances in such a way within the multi-dimensional space, that there is a hyper-plane that separates data instances of one kind from those of another. The kernel equations may be any function that transforms the linearly non-separable data in one domain into another domain where the instances become linearly separable. Kernel equations may be linear, quadratic, Gaussian, or anything else that achieves this particular purpose. Once the data is divided into two distinct categories, our aim is to get the best hyper-plane to separate the two types of instances. This hyper-plane is important because it decides the target variable value for future predictions.

The learnt hyperplane is optimal in the sense that it maximizes the margin while minimizing some measure of loss on the training data. Support vectors are those instances that are either on the separating planes on each side, or a little on the wrong side. SVMs have been shown to work well for high dimensional microarray datasets. One important thing to note is that the data to be separated needs to be binary. Even if the data is not binary, Support Vector Machines handles it as though it is, and completes the analysis through a series of binary assessments on the data.

Linear SVM

In this part of section 2.5.1, we further explain the case of the simple linear SVM algorithm [61],[62] in order to be more clearly the concept of support vectors. Linear SVMs are particular linear discriminant classifiers.

Given a training set X of N samples of the form:

$$X = \{(x_i, y_i) | \, x_i \ \in \ R^m \, , y_i \ \in \ \{-1, +1\}\} \, , i = 1, \dots , N$$

where x_i the samples and y_i the class labels, the support vector method approach aims at constructing the maximum - margin hyperplane of dimension $R^{(m-1)}$ that separate the samples having $y_i \ = \ +1$ from those having $y_i = \ -1$. Any hyperplane can be expressed as the set of samples x satisfying:

$$H : w \cdot x - b = 0$$

, where b a real constant and w the normal vector to the hyperplane. The offset of the hyperplane from the origin along the normal vector w can be expressed by the parameter $\frac{b}{\|w\|}$. If the data are linearly separable, there are two hyperlplanes which can be described by the equations :

$$H_1 : w \cdot x - b = 1$$
$$H_2 : w \cdot x - b = -1$$

that fully separate the two classeses without any samples between of them. The region bounded by these hyperplanes is called "the margin" and is equal to $\frac{2}{\|w\|}$. The aim is to maximize the margin, so $\|w\|$ need to be minimized. Given the fact that $\|w\|$ is minimized, samples of either class may fall into the margin, so in order to avoid it, extra constraints need to be applied:

$w \cdot x_i - b \geq 1$, for samples of class $y_i = +1$
$w \cdot x_i - b \leq -1$, for samples of class $y_i = -1$

The above equations can be expressed in one as:

$y_i(w \cdot x_i - b) \geq 1$, for $i = 1, \dots, N$

Moreover, the previous constrained equation can be expressed as an optimization problem:

Minimize in w, b

$$\|w\|$$

Subject to

$y_i(w \cdot x_i - b) \geq 1$, for $i = 1, \dots, N$

This optimization problem is difficult to solve because it is necessary to calculate the norm of w, which involve a square root. Without changing the solution, it is possible to substitute $\|w\|$ with $\frac{1}{2}\|w\|^2$. So the optimization problem can be also expressed as:

Minimize in w, b

$$\frac{1}{2}\|w\|^2$$

Subject to

$y_i(w \cdot x_i - b) \geq 1$, for $i = 1, \dots, N$

By using the Lagrange multipliers $\boldsymbol{\alpha}$, the previous problem can be expressed as a problem of quadratic programming:

$$arg \min_{w,b} \max_{a \geq 0} \left\{ \frac{1}{2} ||w||^2 - \sum_{i=1}^{n} a_i [y_i (w \cdot x_i - b) - 1] \right\}$$

Then, conforming to the stationary Katush – Kuhn – Turkey condition, the solution can be expressed as a linear combination of the training input vectors:

$$w = \sum_{i=1}^{N} a_i \, y_i x_i$$

Only a few of the Lagrange multipliers $\boldsymbol{\alpha}$ will be greater than zero. These corresponding x_i are the support vectors and lie on the margin, satisfying:

$$y_i (w \cdot x_i - b) = 1$$

Solving the above equation for b can derive that the support vectors also satisfy:

$$w \cdot x_i - b = \frac{1}{y_i} \implies b = w \cdot x_i - y_i$$

The b depends on x_i , y_i, so it will vary among the samples. In that manner, a more stable approach for b is to average over all support vectors:

$$b = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} (w \cdot x_i - y_i)$$

The optimization problem can also be expressed in its dual form, using the fact that $||w||^2 = w \cdot w$ and $w = \sum_{i=1}^{N} a_i \, y_i x_i$. In dual form the classification task takes into account only a function of the support vectors, which are a small subset of the set of the training samples that lie on the margin. Thus, the problem expressed in dual form is computationally efficient.
Maximize in a_i

$$\check{L} \, (a) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j \, x_i^T x_j =$$

$$\sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j \, k(x_i, x_j)$$

, subject to $a_i \geq 0$, $\sum_{i=0}^{N} a_i y_i = 0$ and the kernel function is defined by $K(x_i, x_j) = x_i \cdot x_j$

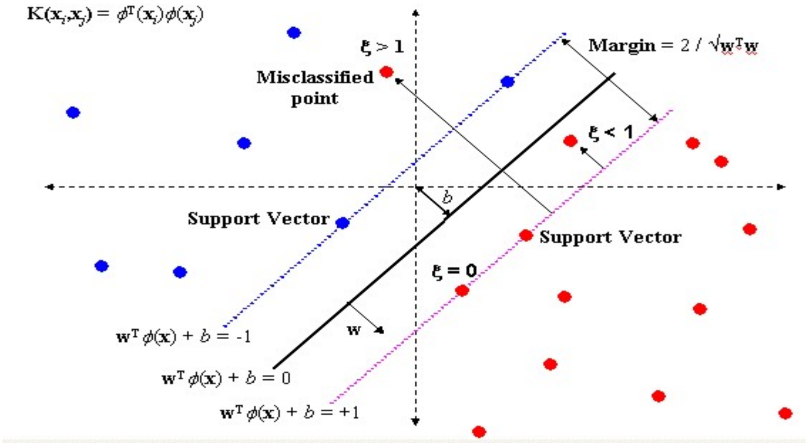


FIGURE 2.12 THE SVM LEARNS A HYPERPLANE WHICH BEST SEPARATES TWO CLASSES. RED DOTS HAVE A LABEL YI = +1 WHILE BLUE DOTS HAVE A LABEL YI = -1

Decision Trees

Decision tree method is a technique in statistical learning that can be applied to both regression and classification problems, where the target variable is categorical, and the tree is used to identify the "class" within which a target variable would likely fall into. They are used to predict a qualitative response. The science and technology behind the review of large and complex datasets to discover valuable patterns is very important for modeling and knowledge extraction from the data which are available [63]. Researchers in this field have continually made great progress and are still making progress in acquiring methods to make the process more efficient, cost effective and accurate. The algorithms were originally implemented in decision theory and statistics and are used to extract knowledge by making decision rules from the large amount of available information. The benefits of decision trees are in its ability to handle a variety of input data such as nominal, numeric, and textual, its processing of dataset that containing errors and missing values, and its availability in various packages of data mining and number of platforms. A decision tree classifier [64] has a simple form which can be compactly stored and that efficiently classifies new data.

When choosing a decision tree, we start with **N** labeled “training records” of the form (**X, Y**) where **X** is a k-dimensional vector of features describing the data we have, and Y is a label we give this record.

Each component of **X** is called as “input variable”, **Y** is called “dependent variable” or “target variable”, and each row in such a table is called a “training example”. Let us consider two input variables, such that $\boldsymbol{X}$= ($X1$, $X2$). We assume there is a value of $X1$ that we can split the dataset around and few values of $X2$. Then, an example partitioning of our space of ($X1$, $X2$) values is depicted in the left side of Figure 2.13, and a decision tree corresponding to such a partitioning is shown in the right side of Figure 2.13. Given an unlabeled vector $\boldsymbol{X}$ = ($X1$, $X2$), we first test whether $X1>a$. Then, if that turns out to be true, we test whether $X2>d$. This allows us to classify $\boldsymbol{X}$ into the region $R4$ or the region $R5$. If we initially had that $X1\leq a$, then we will test $X2$ against c and then against b, which allows us to further classify $\boldsymbol{X}$ into one of the regions $R1$,2, or $R3$.

Next, let Y take on a single constant value for each of the regions $R1$...,5. Let Yi be the value chosen for the region Ri, and let ($\boldsymbol{X}$) be an indicator function that equals 1 when $\boldsymbol{X}\in Ri$. This allows us to obtain a model that can predict Y based on $\boldsymbol{X}$:

$$\hat{Y}(\mathrm{X})=\sum_{i=1}^{5} Y_i \times I_i(X)$$

Obtaining such a model is the ultimate goal of training a decision tree. Same as the model represented in Fig. 2.11 as a partition of 2D space and as a decision tree.

The most basic process of training a decision tree on a dataset involves the following elements as,

1. The selection of attribute
2. Splits in the tree
3. Stop splitting a node and mark it terminal
4. The assignment of a label to each terminal node

Some algorithms add an element called pruning. There are many ways of implementing splitting criteria, stopping criteria, and pruning methods. Splitting criteria are algorithmic rules that decide which input variable to split the dataset around next. Stopping criteria are rules that determine when to stop splitting the dataset and instead output a classification. Stopping criteria are actually optional, but in their absence, a trained tree would have a separate region for each training record. As this is undesirable, stopping criteria are used as a method of deciding when to stop growing the tree. Lastly, pruning methods are ways to reduce the size and complexity of an already trained tree by combining or removing rules that do not significantly increase classification accuracy. All three of these things directly affect the complexity of a tree, which can be measured according to various metrics such as tree height, tree width, and a number of nodes. It is desirable to train trees that are not overly complex because of the fact that simpler trees require less storage.[64]

- **SPLITTING CRITERIA**

An option of making splits is the classification error rate and this is simply the fraction of the training observations in that region that do not belong to the most common class. The classification error is given by;

$$E = 1 - \max \hat{p}_{mk} \qquad (1)$$

where $\hat{p}mk$ represents the proportion of training observations in the region m that are from class k.
Other measures for making splits are Cross entropy and Gini index which are preferred since the classification error is insufficiently sensitive for tree growth. The Gini Index, is given by

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \qquad (2)$$

which is a measure of the total variance across k classes, where $\hat{p}mk$ represents the proportion of training observations in the region m that are from class k. Gini Index is also called a measure of node purity because if all of the values of $\hat{p}mk$, the proportion of training observations in the region m that are from class k are close to 0 or 1 then the Gini index has a small value which can be verified from (1). This implies that a node contains mostly training observations from a single class k.

Cross entropy, is an alternative to the Gini Index and its given by

$$-\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk})$$

Since;

$$0 \leq \hat{p}_{mk} \leq 1 \text{ we have } 0 \leq -\hat{p}_{mk} \log(\hat{p}_{mk})$$

The cross entropy will take a value near 0 if the $\hat{p}_{mk}\,'s$ are all near 0 or 1.

In building a classification tree, we use either Cross entropy or Gini index to evaluate the quality of a particular split, because these two approaches are more sensitive to node purity than the classification error rate. However, when pruning the tree any of the three approaches can be used but the classification error rate is preferable if the prediction accuracy of the final pruned tree is the goal. In the case of classification trees, the deviance is given by the summary function and it can be calculated by

$$-2 \sum_{m} \sum_{k} \hat{n}_{mk} \log (\hat{p}_{mk})$$

$\hat{n}_{mk}$is the number of observations in the m^{th} terminal node that belongs to class k. A tree gives a good fit to the training data if the deviance is small. The residual mean deviance is simply the deviance divided by $n - |T0\ |$.

In order to improve the accuracy of machine learning algorithms for statistical classification and regression, bagging, random forest and boosting are machine learning ensembles that can be used.[66] They are most commonly applied to decision tree methods as building blocks in the creation of very powerful predictive models.

- **STOPPING CRITERIA**

Stopping criteria are usually not as complicated as splitting criteria. Common stopping criteria include:

1. Tree depth exceeds a predetermined threshold
2. Goodness-of-split is below a predetermined threshold
3. Each terminal node has less than some predetermined number of records

Generally stopping criteria are used as a heuristic to prevent overfitting, when a decision tree begins to learn noise in the dataset rather than structural relationships present in the data. An over-fit model still performs very well in classifying the dataset it was trained on, but would not generalize well to new data, just like the example with credit card numbers or other unique identifiers. If we did not use stopping criteria, the algorithm would continue growing the tree until each terminal node would correspond to exactly one record.[64]

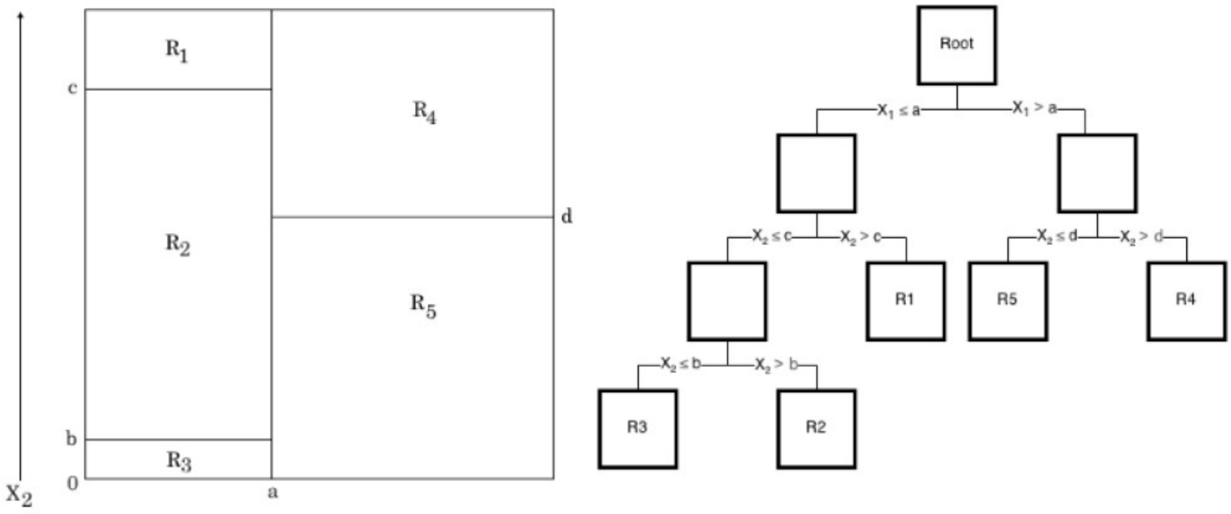


Figure 2.13 Decision Tree

2.6 Validation

2.6.1 Validation methods

Holdout Validation

Holdout Validation is the simplest cross validation method. The dataset is partitioned in two sets, the training set and the testing set. Using the training set only, which consists of the majority of available samples the model, is trained. Then the function is asked to predict the output values for the data in the testing set where the values are unknown. The errors it makes are accumulated to give the mean absolute test set error, which is used to evaluate the model. The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute. However, the drawback of the method is that its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made. These limitations of this holdout method can be overcome with other validation methods at the expense of higher computational cost. (www.towardsdatascience.com)

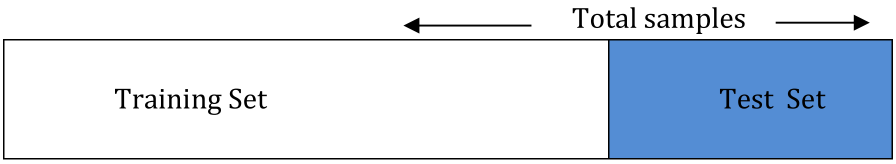


FIGURE 2.14 HOLDOUT VALIDATION METHOD

K-Fold Cross Validation (K-Fold CV)

As we mention before we can use other cross validation methods to improve over the holdout method. K-fold cross validation is one of them. Here, the data set is divided into *k* subsets, and the holdout method is repeated *k* times. Each time, one of the *k* subsets is used as the test set and the other *k-1* subsets are put together to form a training set. Then the average error across all *k* trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once and gets to be in a training set *k-1* times. The variance of the resulting estimate is reduced as *k* is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch *k* times, which means it takes *k* times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set *k* different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over. (www.towardsdatascience.com)

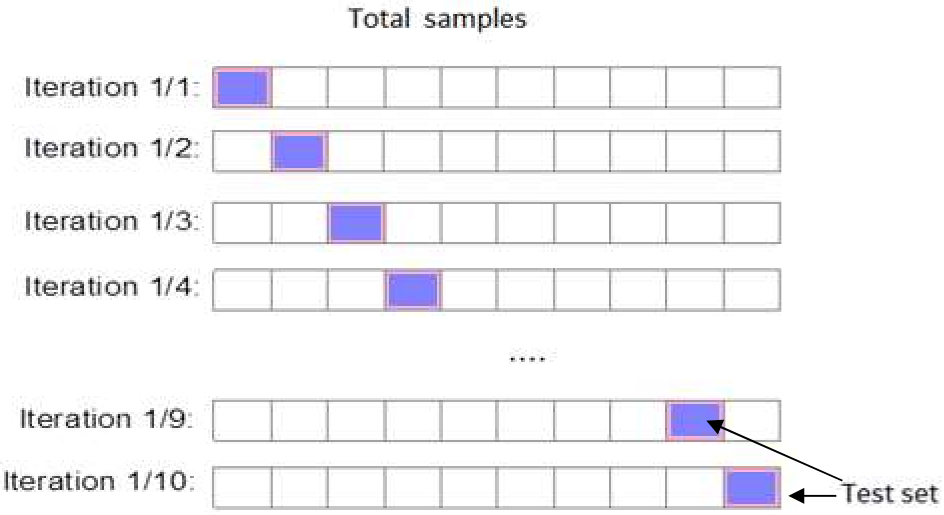


FIGURE 2.15 K-FOLD CROSS VALIDATION METHOD

Leave One Out Cross Validation (LOOC)

Leave-one-out cross validation is K-fold cross validation taken to its logical extreme, with K equal to N, the number of data points in the set. This means that for each fold use N-1 samples for training and the remaining sample for testing. As before the average error is computed and used to evaluate the model. The evaluation given by leave-one-out cross validation error is good, but at first it seems very expensive to compute. Fortunately, locally weighted learners can make LOO predictions just as easily as they make regular predictions. That means computing the LOO validation error takes no more time than computing the residual error and it is a much better way to evaluate models.

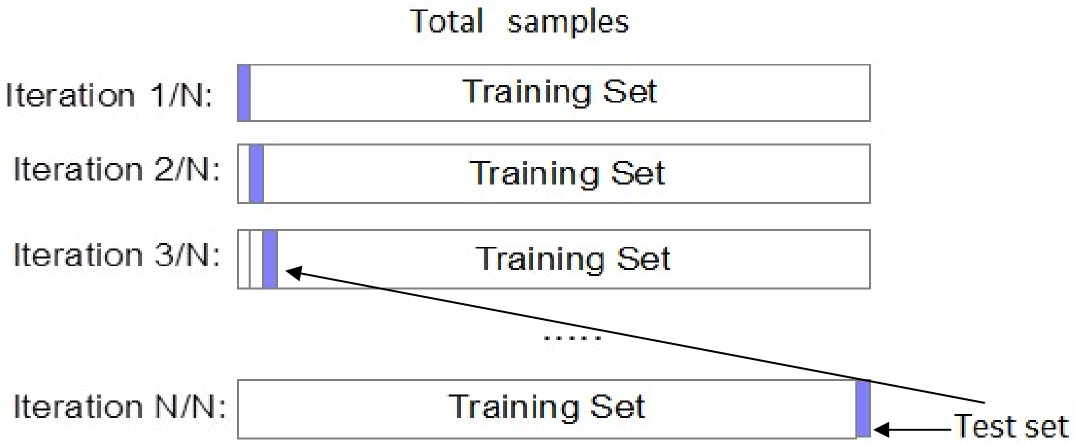


FIGURE 2.16 LEAVE ONE OUT VALIDATION METHOD

Repeated Random Sub-Sampling Validation

In Repeated random sub-sampling validation [56] the dataset splits K times. Each data split randomly selects a fixed number of samples without replacement. For each such iteration, the model is fit to the training data, and predictive accuracy is assessed using the validation data. The results are then averaged over all iterations. In this method unlike *k*-fold cross validation, the proportion of the training split is not dependent on the number of folds. But the disadvantage using repeated random sub-sampling is that some observations may never be selected in the validation subsample, whereas others may be selected more than once.

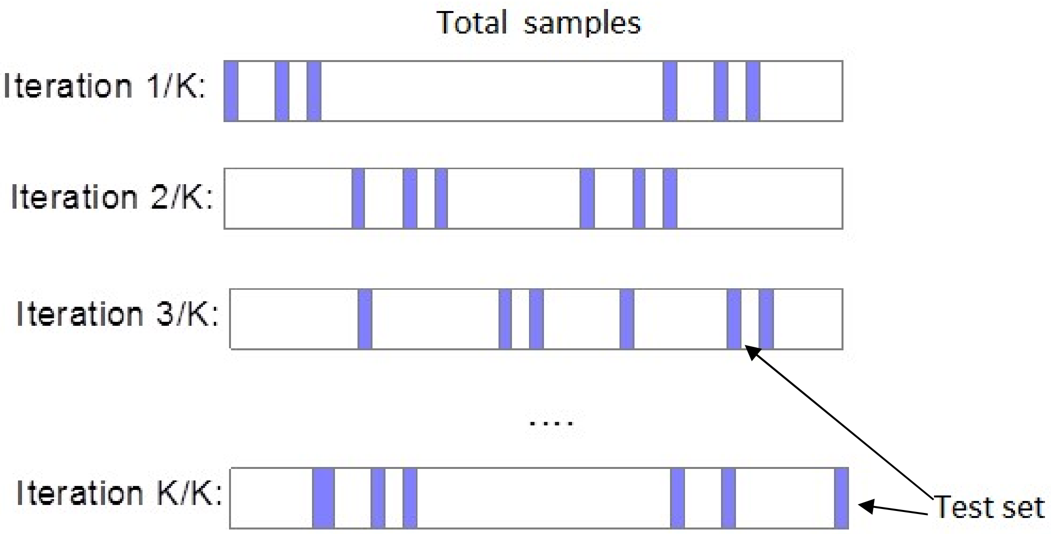


FIGURE 2.17 REPEATED RANDOM SUB-SAMPLING VALIDATION METHOD

2.7 Biological Networks

The term Biological Networks is assigned on biological systems which are represented as networks. Biological networks are the interpretation of the interaction between molecules such as DNA, RNA, proteins and metabolites. There are different types of biological networks such as Gene co-expression network (GCN), Protein-protein interaction networks (PPI), Metabolic networks, Transcriptional regulation networks, Boolean Networks, Bayesian Networks. Those examined in our study are:

Gene co-expression network (GCN)

A gene co-expression network (GCN) is an undirected graph, where each node corresponds to a gene, and a pair of nodes is connected with an edge if there is a significant co-expression relationship between them [66]. Having gene expression profiles of a number of genes for several samples or experimental conditions, a gene co-expression network can be constructed by looking for pairs of genes which show a similar expression pattern across samples, since the transcript levels of two co-expressed genes rise and fall together across samples. Gene co-expression networks are of biological interest since co-expressed genes are controlled by the same transcriptional regulatory program, functionally related, or members of the same pathway or protein complex.

The direction and type of co-expression relationships are not determined in gene co-expression networks like in a gene regulatory network (GRN). Compared to a GRN, a GCN does not attempt to infer the causality relationships between genes and in a GCN the edges represent only a correlation or dependency relationship among genes. Modules or the highly connected sub graphs in gene co-expression networks correspond to clusters of genes that have a similar function or involve in a common biological process which causes many interactions among themselves.

Gene co-expression networks are usually constructed using datasets generated by high-throughput gene expression profiling technologies such as Microarray or RNA-Sequencing. (www.illumina.com)

Protein-protein interaction networks (PPI)

Protein-protein interaction networks (PPIs) can be associations of proteins such as functional interactions and their role is highly important for the structure and the function of a cell. These interactions participate in signal transduction and play an important role in many diseases (e.g., cancer). We can encounter stable interactions that form a protein complex (a form of a quaternary protein structure, set of proteins which bind to do a particular function (e.g., ribosome), or transient interactions, which form the dynamic part of PPI networks, are brief interactions that modify a protein that can further change PPIs –(e.g., protein kineases, add a phosphate group to a target protein). It is estimated that about 70% of interactions are stable and 30% are dynamic in a PPI network thus they are essential to almost every process in a cell. Understanding PPIs is crucial for understanding life, disease, as well as the development of new drugs.[27]

Boolean Networks

Boolean Networks [66] are a class of graphical deterministic models represented as a graph $G(V, E)$, annotated with a set of states $X = \{x_i \mid i = 1, \dots, n\}$, together with a set of Boolean functions $B = \{b_i \mid i = 1, \dots, k\}$,$b_i$:$\{0,1\}^k \rightarrow \{0,1\}$. Each node v_i has associated to it a function, with inputs the states of the nodes connected to v_i.The state of node v_i at time t is denoted as $x_i(t)$ the state of that node at time t+1 is given by : $x_i(t+1) = b_i(x_{i1}, x_{i2}, \dots, x_{ik})$ where x_{ij} are the states of the nodes connected to v_i. This set of functions determines topology connectivity on the set of variables, which then become nodes in a network.

In biological Boolean networks each node represents a gene which takes on two possible values, as mentioned, 0 and 1 and the way these nodes interact with each other is formulated by standard logic functions and genetic interactions and regulations are inextricably linked with the assumption of biological determinism. Though, a gene regulatory network is not a closed system and has interactions with its environment and other genetic networks, and it is also likely that genetic regulations are inherently stochastic; therefore, Boolean networks will have limitations in their modeling power. Probabilistic Boolean networks [67] were introduced to address this issue, such that they are composed of a family of Boolean networks, each of which is considered a context. At any given time, gene regulations are governed by one component Boolean network and network switching is possible such that at a later time instant, genes can interact under a different context. In this sense, probabilistic Boolean networks are more flexible in modeling and interpreting biological data. Interaction networks have proven to be a useful source of information for analyzing genomic data. Using gene expression data we attempt to estimate the network structure using gene and protein information. Boolean Network models belong to the group of qualitative network models, because they do not yield any quantitative predictions of gene expression in the system.

Bayesian Networks

Bayesian Networks [68] are a class of graphical probabilistic models that provide a well-ordered representation for the expression of joint probability distributions (JPDs) and inference. Their application is found in many domains such as the of inference of cellular networks, modeling protein signaling pathways, systems biology, data integration, classification and genetic data analysis. They combine two very well developed mathematical areas: probability and graph theory. A Bayesian network consists of an annotated directed acyclic graph $G(X, E)$, where the nodes $x_i \in X$, are random variables representing gene expressions and the edges indicate the dependencies between the nodes. The random variables are drawn from conditional probability distributions $P(x_i|Pa(x_i))$, where $Pa(x_i)$ is the set of parents for each node. A Bayesian network implicitly encodes the Markov Assumption that given its parents; each variable is independent of its non-descendants.

Besides the set of dependencies (children nodes depend on parent nodes) a Bayesian network implies a set of independencies too. This probabilistic framework is very appealing for modeling causal relationships because one can query the joint probability distribution for the probabilities of events (represented by the nodes) given other events. From the joint distribution one can do inferences and choose likely causalities.

The complexity of such a distribution is exponential in the general case, but it is polynomial if the number of parents is bounded by a constant for all nodes.

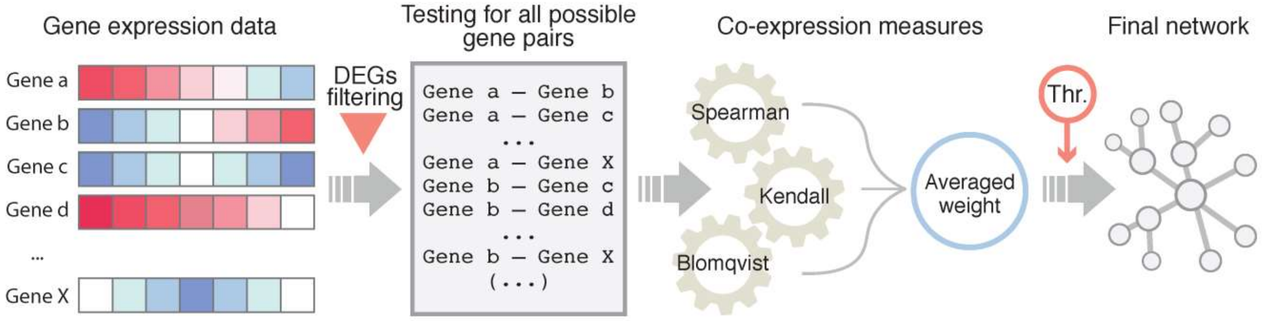


FIGURE 2.18 GENE NETWORK INFERENCE

3 METHODOLOGY

In our study, we aim to provide reliable biomarkers that could be predictive of responder status. By using gene expression profiles from untreated and interferon treated patients as well as healthy controls, we followed a feature selection strategy by combining differential expression analysis, Pigengene methodology, network analysis, and clustering approaches in order to identify key modules, and also hub genes as potential biomarkers for early identification of IFNβ responders as well as Multiple sclerosis affected patients. Moreover, based on related studies as in [69], we sought to identify hub genes, i.e. a limited number of genes - a varying number of 10 to 77 has been recognized in different disease contexts - that interact with many other genes in the clustering modules; thus conferring them high importance in the biological system under study. Publicly available databases were also used for the exploration of drug repurposing relating Multiple sclerosis. The proposed methodology follows.

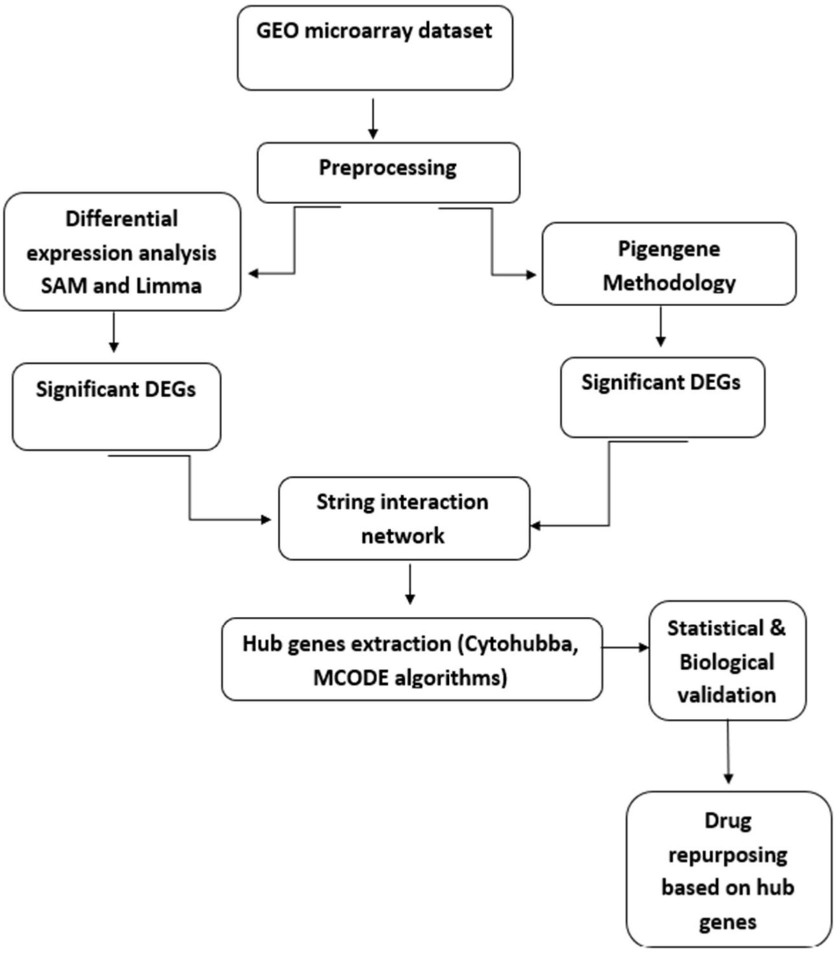

GEO microarray dataset
Preprocessing
Differential expression analysis SAM and Limma
Pigengene Methodology
Significant DEGs
Significant DEGs
String interaction network
Hub genes extraction (Cytohubba, MCODE algorithms)
Statistical & Biological validation
Drug repurposing based on hub genes

FIGURE 3.1 PROPOSED METHODOLOGY

3.1 Microarray Dataset Preprocessing

Quantile Normalization has been performed on all data and log2 transformation has been performed on their expression values. Variance filtering was applied to the dataset as a feature selection method. Scaling as a normalization method was also applied on the validation datasets.

3.2 Differential expression

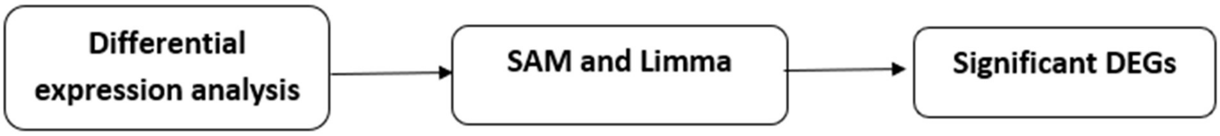


FIGURE 3.2 DIFFERENTIAL EXPRESSION ANALYSIS STEP

Differential expression analysis was performed on the significant genes using the package 'limma' [70], as well as "Significant analysis of Microarray" (SAM)". In general, when the list of Differentially Expressed Genes (DEGs) is only obtained with the use of one high-level analysis, it should not be regarded as reliable and definitive. A possible approach is to use a few methods and acknowledge DEGs as only those genes that are within an intersection of sets of DEGs obtained by different methods [71].

3.2.1 Limma

LIMMA stands for "linear models for microarray data" and contains functionality for fitting a broad class of statistical models called "linear models". Examples of such models include linear regression and analysis of variance. While most of the functionality of limma has been developed for microarray data, the model fitting routines of limma are useful for many types of data and is not limited to microarrays. The objective of Differential expression analysis is to discover which features (genes) are different between groups or stated differently: to discover which genes are differentially expressed between cases and controls.

How samples are distributed between groups determines the design of the study. In addition to the design, there is one or more question(s) of interest(s) such as the difference between two groups. Such questions are usually formalized as contrasts; an example of a contrast is indeed the difference between two groups.
This can be formalized a

$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

In this equation of a linear model, Y is the response variable. It must be a continuous variable. In the context of DEA, it is a relative measure of mRNA expression level for one gene. X_1 is an explanatory variable, which can be continuous or discrete, for example, control group *versus* treatment, or mutant *versus* wild type. β_1 quantifies the effect of the explanatory variable on the response variable. Furthermore, we can add additional explanatory variables to the equation for more complicated experimental designs. Lastly, models the random noise in the measurements.

3.2.2 Significant analysis of Microarrays (SAM)

SAM is a statistical method used to determine statistical significance in gene expressions between groups. In terms of mode of action, SAM uses a modified t-statistic and permutations of the repeated measurements of the data in order to decide if the gene expression is strongly related to the response. However, SAM uses non-parametric statistics since microarray data are not normally distributed. The input to SAM is gene expression measurements from a set of microarray experiments, as well as a response variable from each experiment. The response variable may be a grouping like untreated,

treated (either unpaired or paired), a multiclass grouping (like breast cancer, lymphoma, colon cancer), a quantitative variable (like blood pressure) or a possibly censored survival time. SAM computes a statistic di for each gene i, measuring the strength of the relationship between gene expression and the response variable and order the genes according to their d- values. It uses repeated permutations of the data to determine if the expression of any genes is significantly related to the response by randomly shuffle the values of the genes between groups, such that the reshuffled groups have the same number of elements as the original groups and computes the d-value for each randomized gene. These two steps are repeated many times.

Thus, each gene has many randomized d-values corresponding to its rank from the observed (unpermitted) d-value (100 or 200 permutations are descent for initial exploratory analysis). Then, take the average of the randomized d-values for each gene which is the expected d-value of that gene.
The observed d-values *versus* the expected d-values are then plotted and for each permutation of the data, the number of positive and negative significant genes for a delta parameter, which is the cutoff for significance, chosen by the user based on the false positive rate, is computed. The median number of significant genes from these permutations is the median False Discovery Rate (FDR). Thus, any genes designated as significant from the randomized data are being picked up purely by chance. Therefore, the median number picked up over many randomizations is a descent estimate of FDR. One can also choose a fold change parameter, to ensure that called genes change at least a pre-specified amount.

For accessing the Differential expression of the “Untreated MS patients in different disease stages vs Healthy Controls” cases, we chose to proceed by performing only SAM. When the sample size is small usually leads to unstable test results. In addition, by chance some genes have very small variance, which will result in large t-statistics and small p-values even when the difference is small. Finally, Sometimes data are not normally distributed that can lead to incorrect p-values. For these reasons we proceed with the non-parametric approach to obtain p-values. All results are shown in Chapter 4.

3.3 Pigengene Methodology

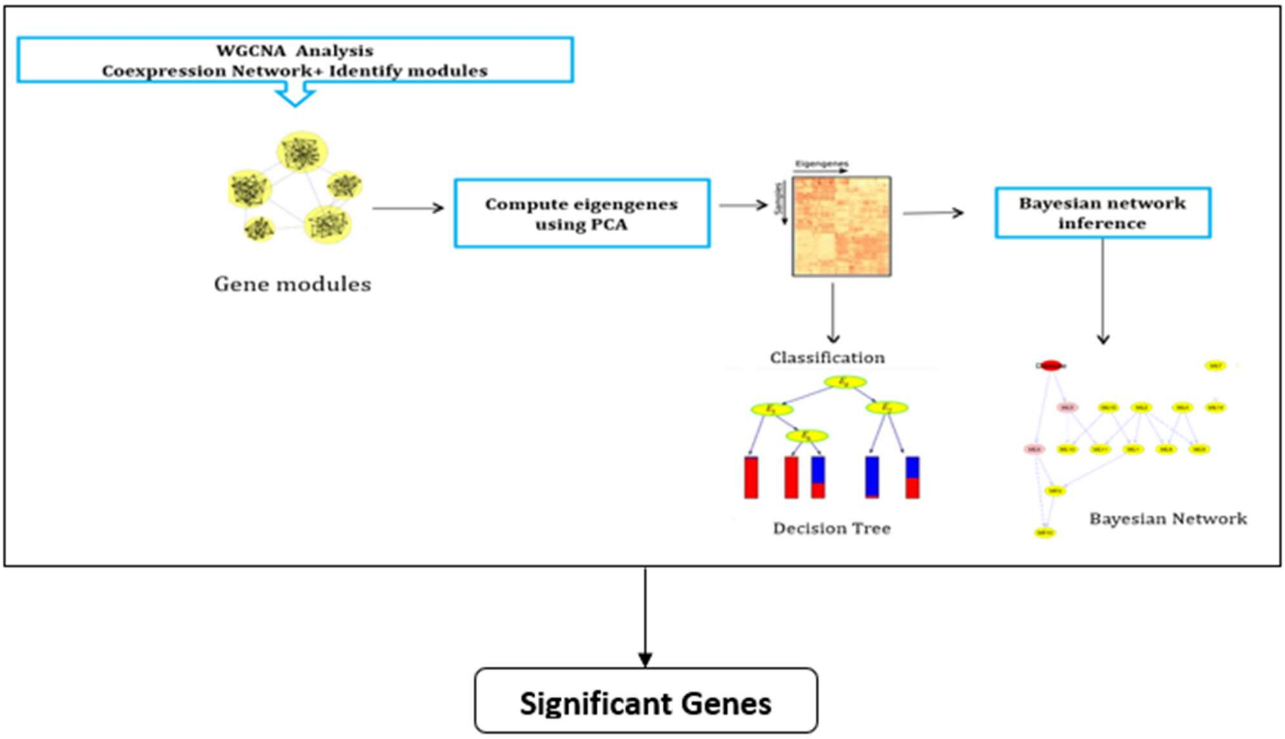


FIGURE 3.3 PIGENGENE STEPS

Pigengene methodology [72] provides an efficient way to infer biological signatures from gene expression profiles. The signatures are independent from the underlying platform, e.g., the input can be microarray or RNA Seq data. It can even infer the signatures using data from one platform and evaluate them on the other. Pigengene identifies the modules (clusters) of highly co expressed genes using co expression network analysis, summarizes the biological information of each module in an eigengene, learns a Bayesian network that models the probabilistic dependencies between modules, and builds a decision tree based on the expression of eigengenes. The crucial steps of the methodology include the identification of gene modules using coexpression network analysis [73] and the summarization of the biological information of each module in one eigengene using principal component analysis (PCA) [74]. The approach is different from applying PCA directly to the entire expression profile, which can lead to a significant loss of information. The eigengenes are used as features of our biological signature to identify mechanisms underlying the disease. They are also used to train a Bayesian network that models the probabilistic dependencies between all modules. In addition, we infer a decision tree to predict the state based on eigengenes.

The Pigengene methodology is presented in detail, in the following sections.

3.3.1 Weighted correlation network analysis (WGCNA)

Weighted correlation network analysis (WGCNA) [75] can be used for finding clusters (modules) of highly correlated genes, for summarizing such clusters using the module eigengene or an intramodular hub gene, for relating modules to one another and to external sample traits (using eigengene network methodology), and for calculating module membership measures. Correlation networks as mentioned in Chapter 2, facilitate network-based gene screening methods that can be used to identify candidate biomarkers or therapeutic targets. These methods have been successfully applied in various biological contexts, e.g., cancer, mouse genetics, yeast genetics, analysis of brain imaging data and in our study, for the first time, Pigengene methodology is applied in the context of Multiple Sclerosis.

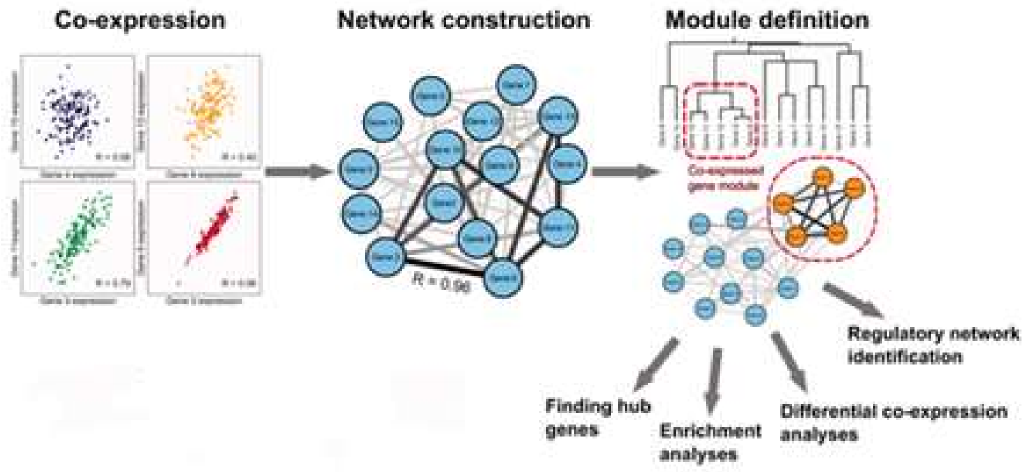


FIGURE 3.4 WEIGHTED CORRELATION NETWORK ANALYSIS (WGCNA) OUTLINE

1. Network construction

The first step of the WGCNA analysis is the creation of a similarity matrix, which is done by Pearson correlation of all gene pairs. The similarity matrix is then transformed into an adjacency Matrix.

From the input n × m matrix X = $[x_{ij}]$ where the row indices (i = 1, . . ., n) correspond to network nodes (such as genes) and the column indices (l = 1, . . ., m) correspond to sample measurements, similarities in expression profiles are calculated by Pearson correlation,

$$\text{Cor}\ (x_i x_j)$$

creating a correlation matrix. The adjacency matrix A = $[a_{ij}]$, is then calculated from the correlation matrix $s_{ij} = |\text{Cor}\ (x_i x_j)|$, by raising the correlation to a soft threshold power β:

$$[a_{ij}] = {s_{ij}}^{\beta},\ \text{for}\ s_{ij} \in [0,1]$$

x_i and x_j are vectors of expression value for gene i and j, s_{ij} represented the Pearson's correlation coefficient of gene i and gene j, a_{ij} encoded the network connection strength between gene i and gene j.

An adjacency function transforms the correlation matrix containing co-expression similarities into the adjacency matrix containing connection strengths. The choice of adjacency function is determined by the weight properties of the network. The term weight properties references whether a network is weighted or unweighted. Unweighted networks apply hard thresholding using the signum function

$$s_{ij} = signum(s_{ij}, \tau) = \begin{cases} 1\ if\ s_{ij} \geq \tau \\ 0\ if\ s_{ij} < \tau \end{cases}$$

which presents intuitive networks (i.e. the number of direct neighbors equals the node connectivity). However, this can present a problem. For example, if the threshold τ is 0.75 and the similarity is 0.74, the connection does not occur and consequently information is lost. Additionally, node connectivity using hard thresholding is sensitive to the choice of the threshold.

The basis of choice of the power β is the assumption of scale free topology of the gene expression network. The node degree distribution p(k) follows a power law in a scale free network. To calculate the scale free model fit for each soft threshold power β, log(p(k)) is plotted against log(k). The R2 value (model fitting index) is close to 1 if the network is scale free. The scale free topology criterion [76] proposes only to consider β that leads to a network that satisfies scale free topology approximately, with R2 > 0.8.

From the adjacency matrix, topological overlap matrix (TOM) Ω = $[\omega_{ij}]$ is constructed, which describes how well connected the genes are, in respect of how many neighbors they share. All entries in the TOM have a connection value to each other between 0 and 1, where a value of 1 meaning that all connections between two nodes and other nodes are the shared and 0 meaning that no connections to other nodes are shared. (TOM) Ω = $[\omega_{ij}]$ provides a similarity measure, which has been found useful in biological networks (Ravasz et al., 2002; Ye and Godzik,2004). For unweighted networks (i.e., a_{ij}= 1 or a_{ij}= 0), Ravasz and colleagues report the following topological overlap matrix in the methods supplement of their paper

$$\omega_{ij} = l_{ij} + a_{ij}/\min\{k_i, k_j\} + 1 - a_{ij}$$

Where l_{ij}=$\sum_u a_{iu}a_{uj}$, and k_i=$\sum_u a_{iu}$ is the node connectivity. Then

$$\omega_i = \sum_{j=1}^{n} \omega_{ij}$$

a TOM-based measure of connectivity ω_i is superior to the standard k_i measure. The topological overlap matrix Ω = $[\omega_i]$ is transformed into a *dissimilarity* matrix defined by $d_{ij} = 1 - \omega_{ij}$, which is subsequently used for clustering gene expression profiles.

2. Gene Module Identification

The following step is the identification of gene modules through unsupervised hierarchical clustering using a TOM-based dissimilarity. Specifically, average linkage hierarchical clustering is performed, and modules are depicted as dendrogram branches. Cutting is performed using the dynamic hybrid tree cut algorithm.

A TOM plot is a color-coded matrix representation of a summary of the co-expression network, which depicts the values of the dissimilarity matrix. Rows and columns are sorted by the hierarchical clustering dendrogram. Red and yellow indicate low and high dissimilarity respectively . Modules are described as red squares along the diagonal. Note that TOM plots are symmetric along the diagonal because they are graphical representations of the topological overlap matrix which is also symmetric. Modules, i.e., groups of genes that are highly co-expressed in most samples are then created from the clusters given from the topological overlap matrix.

3. Module eigengenes

After the construction of the modules, for each module, an eigengene is computed as a weighted average of the expression of all genes in that module. This is a representative gene, defined as the 1st principal component for the co-expression module. The biological information of each module is summarized in one eigengene. By clustering the eigengenes, modules that are very similar are joined together. These steps produced a final set of modules, grouped together based to similarity in gene expression pattern and connectivity.

The modules can then be compared to an external trait or another group, to find the most significant modules to work with. We investigate gene significance (correlation between gene and sample trait) for the trait for each gene in the chosen module, as well as a quantitative measure of module membership (based on the correlation between each gene to the module eigengene). The module membership should be closely correlated to intramodular connectivity and can therefore be used as a measure for this. By investigating the module membership for the genes in the module, it is possible to detect hub genes, which are likely to be biologically important for the pathways or processes represented by that module.[75]

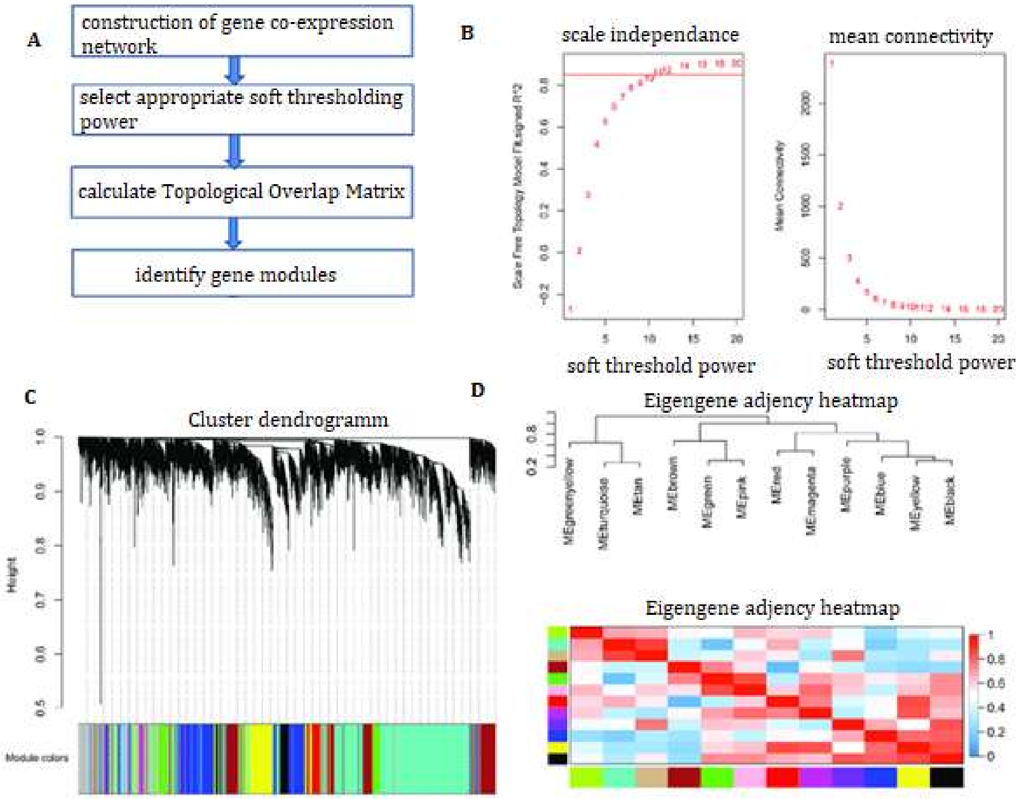


Figure 3.5 Weighted gene coexpression network analysis (WGCNA). (A) WGCNA steps. (B) Determination of the soft threshold. Analysis of the scale-free topology fitting index R 2 (left) and mean connectivity (right) for various soft threshold powers. (C) Clustering diagram showing modules represented by different colors. (D) Clustering tree of module eigengenes and the heatmap of the correlation between any two module eigengenes.

3.3.2 Bayesian network

A Bayesian network is a statistical model that represents a set of random variables using a directed acyclic graph. Nodes of the network correspond to random variables and the edges (arcs) model their conditional dependencies. An important property of Bayesian networks is that each node conditioned on its parent variables is independent of its non-descendants. In particular, if two nodes are not connected by a directed path, they are conditionally independent. We trained a Bayesian network to model the probabilistic dependencies between the modules. Each module eigengene was represented by a node (observed random variable). To model the state of the disease we added

"Disease" as an observed random variable to the network taking values 0 and 1 accordingly. No eigengene was allowed to be a parent of Disease node.

We used bnlearn package to infer the edges and fit the above Bayesian network to the eigengenes. Specifically, we discretized the values of eigengenes into three levels using Hartemink's method. We used the bn.boot() function from the bnlean package to fit 1000 networks to the discretized data. This function used hill climbing strategy to optimize Bayesian Dirichlet equivalent (BDe) score. Consistent with the approach taken by other scholars, we averaged one-third of the networks with the highest scores to obtain the consensus network.

3.3.3 Inferring the decision tree

Module eigengenes are used as features to infer a decision tree as described in Chapter 2. To achieve optimal performance and select the best set of features, when too many features are provided, the Bayesian network is used to determine the relationships of the modules with each other and with the type of sample state in each case of our study. In addition, the parameters of the algorithm, were adjusted, enforcing the number of samples in each node to be at least 10%. We fitted a decision tree to the children of the Disease node in our Bayesian network. We used our data to infer the topology of the tree and the corresponding parameters. Module eigengenes are used to build a classifier that distinguishes two or more classes. Each eigengene is a weighted average of the expression of all genes in the module, where the weight of each gene corresponds to its membership in the module. Each module might contain dozens to hundreds of genes, and hence the final classifier might depend on the expression of many genes. In practice, it is desirable to reduce the number of necessary genes by a decision tree.

3.4 Comparison of resulted significant genes

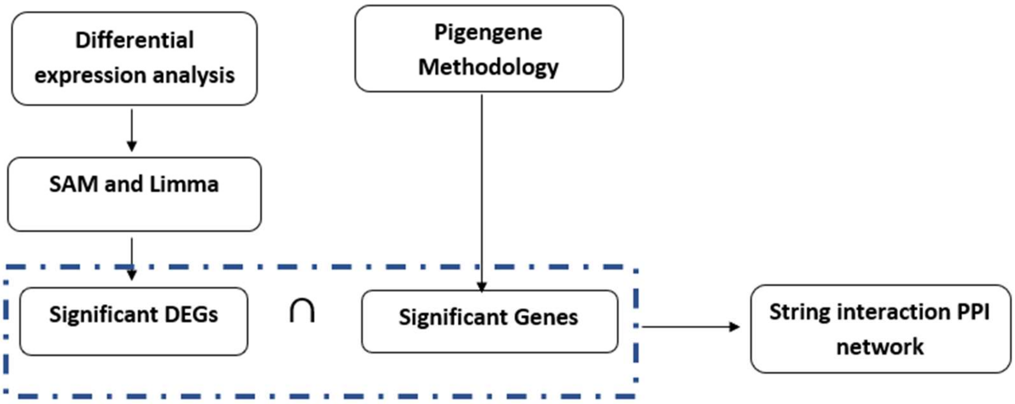


FIGURE 3.6 RESULTED SIGNIFICANT GENES & PPI INPUT STEPS

We have decided to combine the Differentially expressed genes and the significant genes resulted from each Pigengene module. The intersection of the resulted signature in most cases, is then used to explore the protein-protein interaction network and investigate the potential to identify biomarkers related to the study.

3.5 Protein-Protein Interaction Network

The STRING App [77] in the Cytoscape software [78] was used to analyze the significant genes resulted from the two methods mentioned above. The STRING database is one of several online resources dedicated to organism-wide protein association networks. STRING aims to place its focus on coverage (applying to thousands of genome-sequenced organisms), on completeness of evidence sources (e.g. including automated text mining) and on usability features (such as customization, enrichment detection and programmatic access). It allows users to log on and make their searches persistent, and it offers online-viewers to facilitate the inspection of the underlying evidence supporting each protein–protein association. The criteria for constructing the network are based on text-mining, co-expression and databases as well as minimum required interaction score with highest confidence ≥ 0.8.

3.6 Critical Subnetworks and Hub Genes

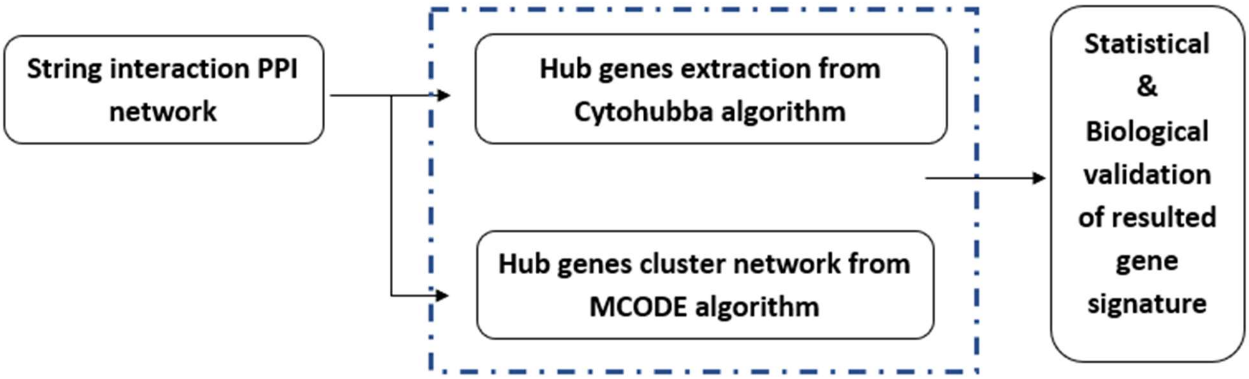

String interaction PPI network
Hub genes extraction from Cytohubba algorithm
Hub genes cluster network from MCODE algorithm
Statistical & Biological validation of resulted gene signature

FIGURE 3.7 FINDING HUB GENES

We used the STRING database to analyze the up-regulated and down-regulated DEGs and analyzed the protein protein interaction (PPI). Cytoscape is a software for visualizing interaction networks and biological pathways. The MCODE plugin was used to find clusters in PPI networks with the degree cutoff, node score cutoff, k-core and max depth as 2, 0.2, 0.2, and 100 as threshold. Moreover, the cytoHubba plugin was used to identify hub genes of the network we imported by calculating the node scores. To get a more reliable result, we analyzed the top 20 nodes with highest degree with all the 11 methods. Then we ordered the number of occurrences of these genes, and the genes with the highest occurrence were the most significant hub genes.

3.6.1 CYTOHUBBA

Based on the PPI network, hub genes were screened according to network topology. Cytoscape software (version 3.9.1) and the cytoHubba plugin [79] was used for ranking nodes in a network by their network features. CytoHubba provides 11 topological analysis methods including Degree, Edge Percolated Component, Maximum Neighborhood Component, Density of Maximum Neighborhood Component, Maximal Clique Centrality and six centralities (Bottleneck, EcCentricity, Closeness, Radiality, Betweenness, and Stress) based on shortest paths.

CytoHubba provides a simple interface to analyze a network with eleven scoring methods. First, scores from all eleven methods are granted to each node of the loaded PPI network by executing "compute hubba result" function in the cytoHubba options in cytoscape menu bar [plugins]. Next, top-ranked nodes of a particular scoring method are retrieved from the cytoHubba tab in Cytoscape control panel, listed in the result panel, and the sub-graph of these selected nodes are shown in the main window with a color scheme from highly essential (red) to essential (yellow). The sub-graph of essential nodes is extendable to include nodes that directly interact with these top-ranked nodes by the option of "check first stage node" in the control panel. Network topological features of nodes are retrievable in the data panel as options of node attributes.

The algorithms

A. Local-based Methods

We assume that a biological network G = (V, E) is an undirected network, where V is the collection of nodes within the network and E is the edge set. We can use another notation G = (V(G), E(G)) to represent a network, where V(G) is the collection of nodes in a network G, and E(G) is the collection of edges in a network G. For a set S, we use |S| to denote its cardinality (i.e. the number of elements in the set).

Local based method only considers the direct neighborhood of a vertex. Given a node v, N(v) denotes the collections of its neighbors. There are four local based methods shown as follows:

i. Degree method (Deg)

$$Deg(v)=|N(v)|.$$

ii. Maximum Neighborhood Component (MNC)

$$MNC(v) = |V(MC(v)) |,$$

where MC(v) is a maximum connected component of the G[N(v)] and G[N(v)] is the induced subgraph of G by N(v).

iii. Density of Maximum Neighborhood Component (DMNC)

Based on MNC, Lin et. al. proposed $DMNC(v) = |E(MC(v))| / |V(MC(v))^{\varepsilon}|$, where ε = 1.7 [80].

iv. Maximal Clique Centrality (MCC)

To increase the sensitivity and specificity, MCC is proposed, to discover featured nodes. Given a node v, the MCC of v is defined as $MCC(v) = \sum_{C \in S(v)} (|C| - 1)!$, where S(v) is the collection of maximal cliques which contain v, and (|C|-1)! is the product of all positive integers less than |C|. If there is no edge between the neighbors of the node v, then MCC(v) is equal to its degree.

B. Global-based methods

In CytoHubba six node ranking methods are implemented, based on shortest paths and one method based percolated connectivity. The length of a shortest path between nodes u and v is denoted as dist(u, v). Let C(v) be the component which contains node v. The dist (u, v) is equal to infinite if C(v) ≠ C(w), and it makes methods of this category cannot be applied to networks with disconnected components. To overcome this problem the score of a node in a connected network computed by enhanced method is the same as that computed by original one.

1. **Closeness (Clo)**

$$\mathrm{Clo(v)}=\sum_{w\in V}\frac{1}{dist(v,w)}$$

2. **EcCentricity (EC)**

$$\mathrm{EC(v)}=\frac{|\mathrm{V(C(v))}|}{|\mathrm{V}|}\ \mathrm{x}\ \frac{1}{\max\{\mathrm{dist(v,w)}:\mathrm{w}\in\mathrm{C(v)}\}}$$

3. **Radiality (Rad)**

$$\mathrm{Rad(v)}=\frac{|\mathrm{V(C(v))}|}{|\mathrm{V}|}\ \mathrm{x}\ \frac{\sum_{\mathrm{w}\in\mathrm{C(v)}}(\Delta_{C(v)}+1-\mathrm{dist(v,w)})}{\max\{\mathrm{dist(v,w)}:\mathrm{w}\in\mathrm{C(v)}\},}$$

where $\Delta_{C(v)}$is the maximum distance between any two vertices of the component C(v).

4. **BottleNeck (BN)**

Let T_S be a shortest path tree rooted at node s. BN(v) = $\sum_{\mathrm{s}\in\mathrm{V}} p_{s(v)}$ where $p_{s(v)}$ = 1 if more than |V(Ts)|/4 paths from node s to other nodes in T_S meet at the vertex v; otherwise $p_{s(v)}$ = 0.

5. **Stress (Str)**

$$\mathrm{Str\ (v)} = \sum_{s\neq t\neq v\in C(v)}\sigma_{st}(v)$$

where $\sigma_{st}(v)$ is the number of shortest paths from node s to node t which use the node v.

6. **Betweenness (BC)**

$$\mathrm{BC\ (\upsilon)} = \sum_{s\neq t\neq v\in C(v)}\frac{\sigma_{st}(\upsilon)}{\sigma_{st}}$$

where σ_{st} is the number of shortest paths from node s to node t.

7. **Edge Percolated Component (EPC)**

Given a threshold (0 ≤ the threshold≤ 1), we create 1000 reduced networks by assigning a random number between 0 and 1 to every edge and remove edges if their associated random numbers are less than the threshold.

Let the G_k be the reduced network generated at the kth time reduced process. If nodes u and v are connected in G_k, set $\delta_{ut}^{k}$to be 1; otherwise δ_{ut}^{k}=0. For a node v in G, EPC(v) is defined as $\mathrm{EPC(\upsilon)}=\frac{1}{|\mathrm{V}|}\sum_{k=1}^{1000}\sum_{t\in\mathrm{V}}\delta_{ut}^{k}$

3.6.2 MCODE

"Molecular Complex Detection" (MCODE) [81], is an algorithm that detects densely connected regions in large protein-protein interaction networks that may represent molecular complexes. It is a graph theoretic clustering algorithm, and it is based on vertex weighting by local neighborhood density and outward traversal from a locally dense seed protein to isolate the dense regions according to given parameters. The algorithm has the advantage of having a directed mode that allows fine-tuning of clusters of interest without considering the rest of the network and allows examination of cluster interconnectivity, which is relevant for protein networks.

The algorithm

The MCODE algorithm may be run in an undirected or a directed mode. Typically, when analyzing complexes in a given network, one would find all complexes present (undirected mode) and then switch to the directed mode for the complexes of interest. The algorithm operates in three stages:

1. **vertex weighting**
2. **complex prediction**
3. **optionally post-processing to filter or add proteins in the resulting complexes by certain connectivity criteria.**

A network of interacting molecules can be intuitively modeled as a graph, where vertices are molecules and edges are molecular interactions. If temporal pathway or cell signalling information is known, it is possible to create a directed graph with arcs representing direction of chemical action or direction of information flow, otherwise an undirected graph is used. Using this graph representation of a biological system allows graph theoretic methods to be applied to aid in analysis and solve biological problems. This graph theory approach has been used by other biomolecular interaction database projects such as DIP [82], CSNDB [83] and is discussed by Wagner and Fell [84].

Algorithms for finding clusters, or locally dense regions, of a graph are an ongoing research topic in computer science and are often based on network flow/minimum cut theory [85] and spectral clustering [86]. To find locally dense regions of a graph, MCODE instead uses a vertex-weighting scheme based on the clustering coefficient, C_i, which measures 'cliquishness' of the neighborhood of a vertex.

$$C_i = 2\text{n}/k_i\ (k_i\text{-}1)$$

where k_i is the vertex size of the neighborhood of vertex i and n is the number of edges in the immediate neighborhood density of v not including v. A clique is defined as a maximally connected graph. We can define the density of a graph, G = (V, E), with number of vertices, |V|, and number of edges, |E|, as |E|; divided by the theoretical maximum number of edges possible for the graph, $|E|_{max}$. For a graph with loops, $|E|_{max}$= |V| (|V|+1)/2 and for a graph with no loops, |E|max = |V| (|V|-1)/2.

So, density of G, DG = $|E|/|E|_{max}$ and is thus a real number ranging from 0.0 to 1.0.

Undirected Mode

1. **Vertex weighting**

Vertex weighting, weights all vertices based on their local network density using the highest k-core of the vertex neighborhood. A k-core is a graph of minimal degree k (graph G, for all v in G, deg(v) >= k). The highest k-core of a graph is the central most densely connected subgraph. We define here the term core-clustering coefficient of a vertex, v, to be the density of the highest k-core of the immediate neighborhood of v (vertices connected directly to v) including v (note that Ci does not include v). The core-clustering coefficient amplifies the weighting of heavily interconnected graph regions while removing the many less connected vertices that are usually part of a biomolecular interaction network, known to be scale-free [76]. A given highly connected vertex, v, in a dense region of a graph may be connected to many vertices of degree one (singly linked vertex). These low degree vertices do not interconnect within the neighborhood of v and thus would reduce the clustering coefficient, but not the core-clustering coefficient. The final weight given to a vertex is the product of the vertex core-clustering coefficient and the highest k-core level, k max, of the immediate neighborhood of the vertex. This weighting scheme further boosts the weight of densely connected vertices. This specific weighting function is based on local network density.

2. Complex prediction

Molecular complex prediction, takes as input the vertex weighted graph, seeds a complex with the highest weighted vertex and recursively moves outward from the seed vertex, including vertices in the complex whose weight is above a given threshold, which is a given percentage away from the weight of the seed vertex. This is the vertex weight percentage (VWP) parameter. If a vertex is included, its neighbors are recursively checked in the same manner to see if they are part of the complex and the process stops once no more vertices can be added to the complex based on the given threshold which defines the density of the resulting complex. If the threshold is closer to the weight of the seed vertex a smaller, denser network region around the seed vertex is identified. A vertex is not checked more than once since complexes cannot overlap in this stage of the algorithm. The process is repeated for the next highest unseen weighted vertex in the network in order to identify the densest regions of the network.

3. Post-processing

Complexes are filtered if they do not contain at least a graph of minimum degree=2.
Post-processing can be achieved with two options:

i. 'fluff' option, which increases the size of the complex according to a given parameter between 0.0 and 1.0. For every vertex in the complex, v, its neighbors are added to the complex if they have not yet been seen and if the neighborhood density (including v) is higher than the given parameter. Vertices that are added by the fluff parameter are not marked as seen, so there can be overlap among predicted complexes with the fluff parameter set.

ii. The 'haircut' option where the resulting complexes are 2-cored, thereby removing the vertices that are singly connected to the core complex.

If both options are specified, fluff is run first, then haircut.

Resulting complexes from the algorithm are scored and ranked. The complex score is defined as the product of the complex subgraph, C = (V,E), density and the number of vertices in the complex subgraph (DC × |V|). Thus larger and more dense complexes are ranked higher in the results.

Directed mode

A seed vertex is specified as a parameter. When directed mode is chosen, MCODE only runs once to predict the single complex that the specified seed is a part of. The directed mode allows one to experiment with MCODE parameters to fine tune the size of the resulting complex according to existing biological knowledge of the system. In directed mode, MCODE will first pre-process the input network to ignore all vertices with higher vertex weight than the seed vertex. If this were not done, MCODE would preferentially branch out to denser regions of the graph, if they exist, which could belong to separate, but denser complexes. Thus, a seed vertex for directed mode should always be the highest density vertex among the suspected complex.

The time complexity of the entire algorithm is polynomial O(nmh3) where n is the number of vertices, m is the number of edges and h is the vertex size of the average vertex neighborhood in the input graph, G.

Finally, we have to mention the advantages of MCODE:

- weighting is done once and comprises most of the time complexity, many algorithm parameters can be tried, in O(n), once weighting is complete which is useful when evaluating many different parameters.
- relatively easy to implement
- since it is local density based, has the advantage of a directed mode and a complex connectivity mode. These two modes are generally not useful in typical clustering applications but are useful for examining molecular interaction networks. Additionally, only those proteins above a given local density threshold are assigned to complexes. This is in contrast to many clustering applications that force all data points to be part of clusters, whether they truly should be part of a cluster or not.[81]

3.7 Statistical Evaluation-Generalization

In microarray and data analysis evaluation methods are used to estimate the generalization ability of genome signature, that is to discover predictive relationships of the results in independent data. Evaluation methods can be performed in a portion of the existing dataset as well as in an independent/new dataset, called the training set while a test set is used for evaluating whether the discovered relationships are accurate. A test set is a set of data used to assess the strength and utility of a predictive relationship. Cross-validation, explained in section 2.7 is a well-known and used strategy because of its simplicity and its universality. The k – fold cross validation approach, implemented in this study, can also be used to assess how the results of a statistical analysis will generalize to an independent data set. In this context, a new independent dataset is used and the procedure of 10 – fold cross validation is repeated.

3.8 Biological Evaluation

Apart from the important step of the statistical evaluation of our results and their prediction ability, a fundamental role in the process of evaluation is the biological significance of the resulted genes. In combination, these two methods can help us uncover known as well as new relationships between genes/proteins which if applying either one or the other separately, our conclusion would be incomplete and would lack in terms of statistical as well as biological significance.

A commonly used step to understand biological data is to evaluate the functional properties of gene sets of interest. For this purpose, functional enrichment tests are widely applied in biomedicine field to uncover trends in large scale biological datasets, and to identify disease and drug mechanisms. Here, we performed an over-representation analysis to explore the functional information (biological processes, pathways) of our gene sets, the differentially expressed gene (DEG) signatures and the hub-gene signatures, in order to identify clear trends for each case study. The over-representation analysis of the gene signatures was performed in WebGestalt (2019) (http://www.webgestalt.org/) using Gene Ontology-Biological Process categories and Pathway categories and the entire genome as a reference set. Enrichment p values were adjusted using Benjamini-Hochberg correction and a false discovery rate (FDR) threshold of 5% was used as significance cut-off. In the case that no significant results were found under threshold FDR 0.05, the top 10 enrichment terms were selected to present the general trends. In section 4 the results from the biological evaluation of our resulted gene signature are presented.

3.9 Drugs and Gene signature interaction

The final step of our methodology is the exploration of the ability to repurpose drugs based on oyr resulted gene signature. The resulted hub genes are screened and used for searching drugs-genes associations through the DGIdb database [87] towards drug repurposing in Multiple Sclerosis. This database has drug–gene interaction data from 30 disparate sources such as ChEMBL, DrugBank, Ensembl, NCBI Entrez, PharmGKB, and literature in NCBI PubMed. Drugs supported by no less than 2 databases or PubMed references are validated as the candidate drugs. The final list only contains the drugs that have been approved by the Food and Drug Administration. Additionally, the identified target gene network is constructed through the STITCH database, a software that also incorporates drug–gene relationships [88].

4 Results and Discussion

In this Chapter, we present the results deriving from the application of our proposed methodology after examining two cases: A) MS untreated patients vs MS interferon treated patients, B) MS untreated patients vs Healthy controls and C) MS untreated patients in different stages of the disease vs Healthy controls. In section 4.1 we introduce the datasets that we have used, followed by section 4.2, 4.3 where each case is presented separately. Furthermore, in section 4.4 the statistical significance as well as the resulted genome signature significance of our approach and our implementation results is assessed. In section 4.5 we examine the drug repurpose ability based on the genomic signature deriving from each examined case and finally in section 4.6 we present our conclusions, remarks and future work goals.

4.1 Datasets

In this study the datasets that were examined were acquired from Gene Expression Omnibus [89]. We have selected the raw data in order to process them, as mentioned in the following sections.

For the overall design, the following cases were examined:

A. Untreated MS vs Interferon treated MS (discovery and replication), dataset GSE41850

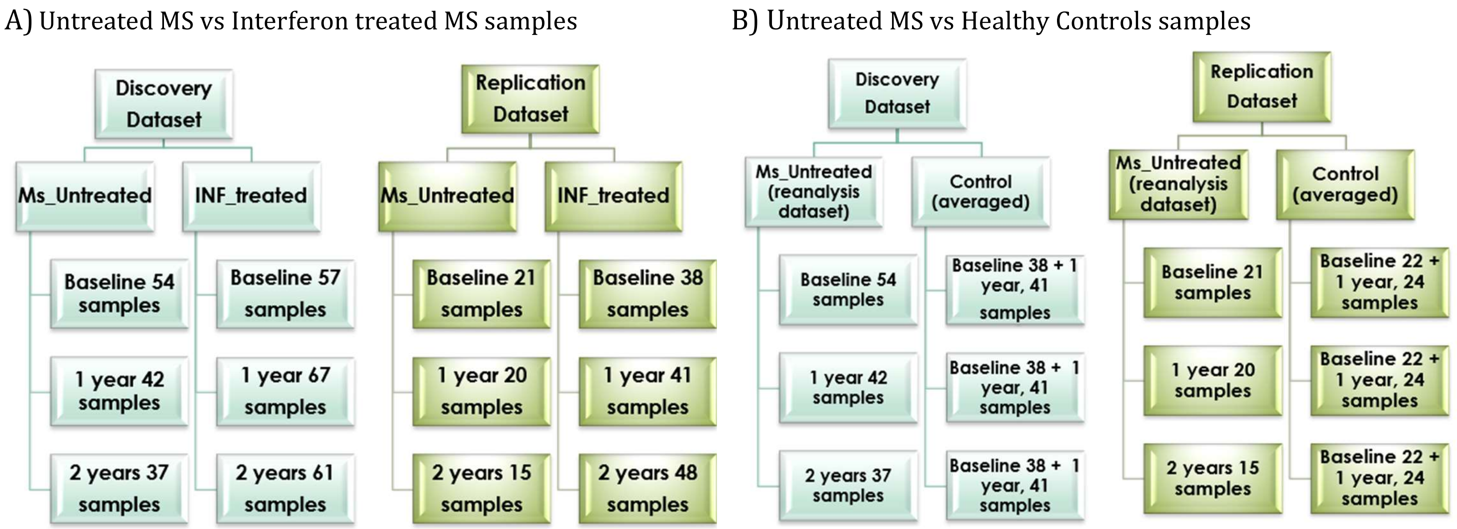
B. Untreated MS vs Controls (discovery and replication), dataset GSE41850

C. Untreated MS in different disease stages vs Controls, dataset GSE136411

4.1.1 Dataset GSE41850

Our first raw dataset was acquired from Gene Expression Omnibus, accession number GSE41850 [35]. Gene expression values derived from whole blood RNA from a cohort of 195 MS patients treated with interferon β and untreated and 66 healthy controls. We examined samples from 120 MS patients (at three consecutive years) and 41 healthy controls (at two time points) as discovery data set and another set of 75 MS patients (at three consecutive years) and 25 healthy controls (at two time points) that were selected at random as the replication data set. In total, 626 Affymetrix exon arrays were analyzed arrays split into discovery and replication data sets. (Figure 4.1A, B)

For each comparison, respective arrays were processed, background corrected and normalized together but separate from other comparisons. The two time points for controls were averaged (baseline + 1 year). Our data were processed in R.


A) Untreated MS vs Interferon treated MS samples

B) Untreated MS vs Healthy Controls samples

Figure 4.1 A) untreated MS *vs* Interferon treated MS (discovery and replication), B) untreated MS *vs* Interferon treated MS (discovery and replication)

4.1.2 Dataset GSE136411

The second dataset that was examined was also from Gene Expression Omnibus, accession number GSE136411 [90]. The dataset includes a total of 313 individuals (172 females and 141 males, with a mean age of 41.7 y.), comprising of 60 healthy controls (HC), 57 subjects with Clinically isolated syndrome (CIS), 169 clinically defined MS cases. The MS cohort contained 108 relapsing-remitting MS (RRMS), 26 secondary progressive MS (SPMS) and 35 primary progressive MS (PPMS) cases. 176 subjects (39 HC, 46 CIS, 23 PPMS, 47 RRMS, 21 SPMS) out of 313 were included in a previously published study, 115 subjects were sampled twice to evaluate biological variability. Additional 137 subjects (21 HC, 11 CIS, 12 PPMS, 61 RRMS, 5 SPMS, 27 OND) were recruited for this study. The datasets raw intensities were background subtracted and filtered according to detection p values (p<0.05 in at least 20% of samples) and then normalized using quantile normalization. Pre-processed data were log2 transformed.

4.1.3 Dataset GSE73608

Dataset GSE73608 [91] was the validation dataset in the "untreated MS vs Interferon treated MS patients" Case study. The dataset had two group of samples**,** first group (N = 35, RRMS-untreated n = 25, RRMS_IFN responders n=10) and second group (N = 50, SPMS_untreated n=30, SPMS_IFN treated n=20). Peripheral blood mononuclear cells (PBMC) were collected from RRMS and SPMS patients. All patients were diagnosed according to McDonald's 2010 diagnostic criteria. The raw dataset was processed based on the pre-processing steps of GSE41850 dataset.

4.1.4 Dataset E-MTAB-5151

We have acquired Dataset E-MTAB-5151 [92] from the ArrayExpress database. This dataset was used as validation in the "untreated MS vs Healthy Controls" case. It was established on the platform of A-AFFY-44-Affymetrix Gene Chip Human Genome U133 Plus 2.0 [HG-U133_Plus_2]. Dataset E-MTAB-5151 contains 76 peripheral blood mononuclear cell samples, including 15 PPMS, 21 RRMS, 13 SPMS, in total 49 MS diagnosed patients and 27 healthy control samples. The patients with MS were diagnosed according to McDonald criteria6 and were not suffering from any other acute or chronic inflammatory diseases or other autoimmune disorders. Furthermore, they had not started any immunomodulatory therapy for MS yet. The raw gene expression data of the three MS stages were considered one group of MS untreated patients and were processed following the pre-processing steps of the GSE41850 dataset.

4.1.5 Dataset E-MTAB-4890

Dataset E-MTAB-4890[93] was downloaded from the ArrayExpress database and was used for the validation and the examination of the generalization ability of the results from case "Untreated MS patients in different disease stages vs Healthy Controls". E-MTAB-4890 includes a total of 182 individuals and global mRNA expression from peripheral blood mononuclear cells (PBMC) was measured, comprising of 142 multiple sclerosis (MS) patients affected with distinct MS clinical forms (PPMS=23, RRMS=52, SPMS=21) and 40 healthy controls. The raw gene expression data of the three MS stages were processed following the pre-processing steps of the GSE136411 dataset.

4.2 Case untreated MS *vs* Interferon treated MS patients

The commonly used disease-modifying treatment (DMT) interferon (IFN) beta is believed to modulate the immune response, reduce new inflammatory lesions in the CNS and partially protect against progression of disability. However, patients vary considerably in their responsiveness to these therapies, and for any individual patient, the natural history of MS is extremely heterogeneous, varying from a benign condition to a devastating and rapidly incapacitating disease. For these reasons, a better characterization of patients is much needed to ultimately understand the diversity of disease presentation.

4.2.1 Datasets preprocessing and Differential expression

The first step was Filtering our dataset. We have created a new file with all discovery samples with 18.726 genes x 318 samples. In the discovery data set, a variance filter, difference between the 10% and 90% quantiles > 0.7, yielding 6.924 genes (329 > than original paper) was applied to normalized gene expression values in order to decrease the number of tested genes. Then group 1 (untreated patients) was compared to group 2 (IFN treated patients) at any of the three measured time points. The union of genes at all three time points passing the FDR cutoff of 0.0001 were considered to be differentially expressed and assessed for differential expression in the replication data set.

In the replication dataset the procedure was repeated: group 1 (untreated patients) was compared to group 2 (IFN treated patients) at all three measured time points, and the union of genes reaching a nominal p-value of 0.05 or smaller at any of these time points was considered to be replicated.

We report differentially expressed genes at the FDR cutoff was 0.0001. The respective genes were validated in the replication data set when they passed a nominal p-value cutoff of 0.05 at any of the three tested time points. In the discovery data set, differentially expressed genes were identified by applying stringent FDR-corrected P-value filters; these genes were then tested for validation in the replication data set. (R limma : Linear Models for Microarray Data)

After applying gene filtering and differential analysis in each time point we have concluded in **6.924** genes from discovery dataset (FDR < 10-4) and their union yielded 313 significant genes. The replication dataset was tested based on these 6.924 genes and 531 genes were selected with p-value<

0.05. Based on the discovery and replication set, **274** genes were common and were considered significant from our “Ms_Untreated- INF_treated” case.

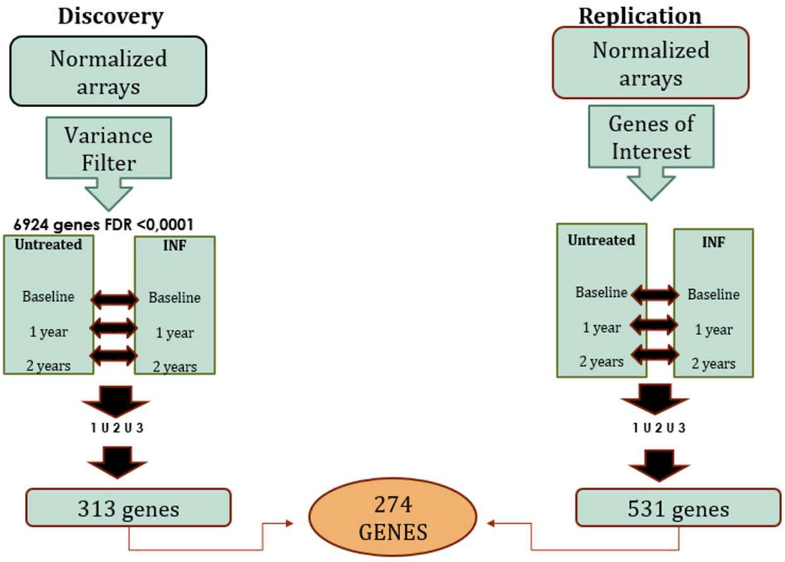


Figure 4.2 Dataset preprocessing and Differential expression: untreated MS *vs* Interferon treated MS (discovery and replication)

4.2.2 Significance Analysis of Microarrays (SAM)

To further evaluate the results, we conducted a Significance Analysis of Microarrays (SAM) on our filtered datasets for both cases, in order to find differentially expressed genes based on T-statists. The cutoff for significance is determined by a tuning parameter delta, chosen by the user based on the false positive rate. One can also choose a fold change parameter, to ensure that called genes change at least a pre-specified amount.

LIMMA and SAM	**MS Untreated *vs* INF treated at three-time points**			**FINAL COMMON GENES**
	Discovery dataset (DEGs)	***Replication dataset (DEGs)***	***Common genes***	
LIMMA SAM	313 777	531 936	274 614	213

Table4.1 Differentially Expressed Genes MS Untreated *vs* INF treated

After the analysis with SAM, we compared the SAM results to our Limma analysis in R and we concluded in 213 genes considering the case MS Untreated- INF treated.

4.2.3 Clustering

Hierarchical clustering using maximum distance and ward clustering was performed on the discriminant 213 signature genes from untreated subjects and Interferon treated (Fig. 2). Two distinct clusters are observed. A subset of patients in both data sets shows a strong IFN response (high IFN gene expression). 182 genes out of 213 were found also in our reference paper [35].

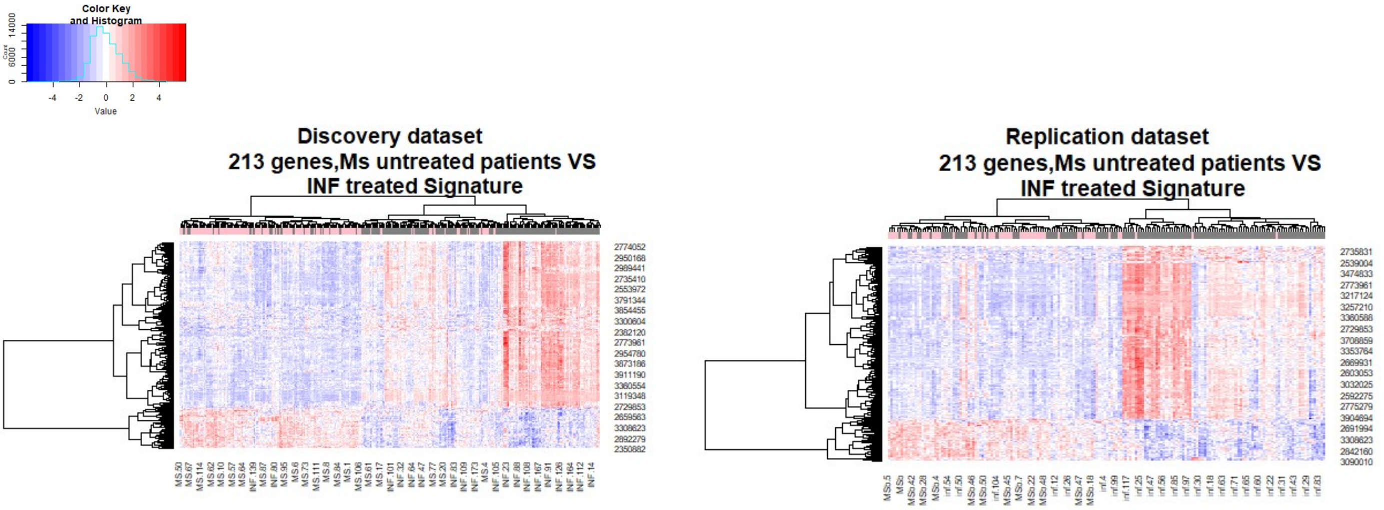


Figure 4.3 Unsupervised hierarchical clustering of MS untreated and IFN treated patients according to the expression of IFN signature genes in the discovery (A) and the replication (B) data sets. The rows are different genes; the columns reflect different samples. The colored bar above the heatmap visualizes whether the patient was untreated (pink) or IFN treated (grey). Dark blue depicts low, red high expression.

We observe the discriminatory power of the resulted 213 gene signature in unsupervised hierarchical clustering (Figure 4.3). The heatmap shows a uniform cluster of MS INF treated patients (grey) with several smaller uniform clusters, an observation that stands in the replication set.MS untreated cases also clustere in a distinct cluster (pink), indicating that gene expression changes evoked by the INF treatement are noticable.

4.2.4 Pigengene Methodology

The proposed methodology aims to find a minimum set of significant genes that will be able to predict the state of a new sample as well as provide meaningful biological information through the correlation and combination of genes in pathways and smaller groups/networks. We apply the Pigengene methodology streps on the 6.924 genes that derived from the preprocessing step.

i. Weighted correlation network: Weighted Coexpression network analysis (WGCNA) was applied to group related genes into gene modules (clusters) based on their coexpression patterns in MS. WGCNA uses the average linkage hierarchical algorithm to cluster the genes. (Figure4.4 A). For each gene module, WGCNA computes one eigengene, which summarizes the biological information in that module into one value per sample. We used these eigengenes to train a Bayesian network (BN) in which nodes (random variables) represent gene modules, and the directed edges (arcs) represent the conditional dependencies between the eigengenes.

ii. Eigengenes: We computed an eigengene for each module as a weighted average of the expression of all genes in that module. Eigengenes are important biological signatures that can predict disease types solely based on gene expression (Figure4.4 B). Module 5 is negatively associated with the Interferon treatment, whereas Module 6 is positively associated with Interferon treatment. To validate this, we modeled the probabilistic dependencies between the eigengenes using a BN (Figure 4.5). We used Bayesian networks as probabilistic predictive models to determine the state.

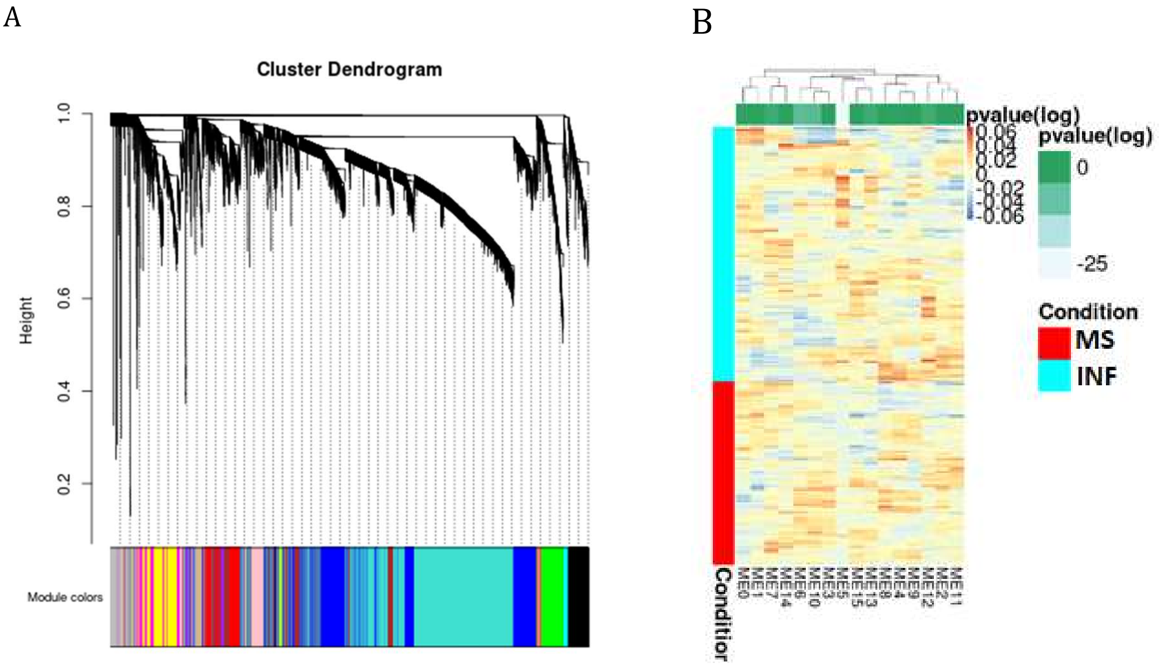


Figure 4.4 A) Modules dendrogram B) The two (2) eigengenes that are differentially expressed ME5, ME6. The intensity of the colors in each heatmap corresponds to the normalized average expression. Each column corresponds to an eigengene. Each row shows the expression of a case from the MS vs MS_INF dataset.

iii. Bayesian network: We fitted a Bayesian network to the eigengenes to determine the relationships of the modules with each other and with the state of the samples. Descendants of the “Disease” node, the variable that models the state, show high dependency between these eigengenes and the state type and suggest that they have useful biological information that can explain the differences between the two states. We trained a Bayesian network to model the probabilistic dependencies between the modules. Several individual networks from random staring networks were built (no.1000) by optimizing their score. Then, we inferred a consensus network from the ones with relatively "higher" scores. The default hyper-parameters and arguments are then selected. Each module eigengene is represented by a node (observed random variable). To model the condition, we added “Disease” as an observed random variable to the network.

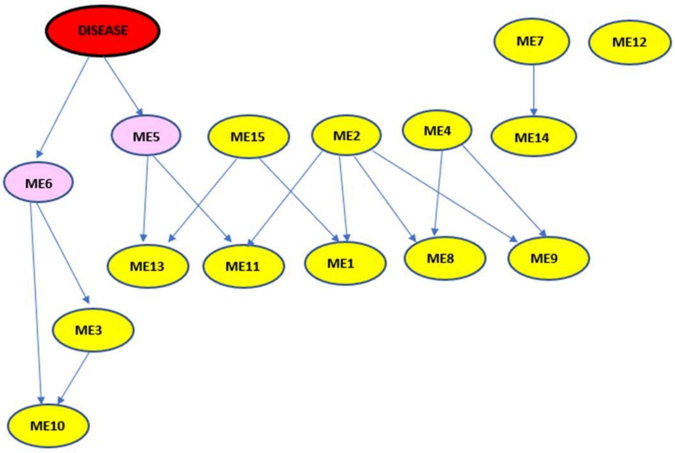


Figure 4.5 The Bayesian network fitted to the eigengenes. Each node represents an eigengene of a module. The arcs model the probabilistic dependencies between the modules. The “Disease” node is set to 1 for MS and 0 for MS_INF, and its children are highlighted in pink.

iv. Decision tree: A decision tree is fitted to the two children of the Disease node in our Bayesian network (R package C50 version 0.1.0-24). We used the data to infer the topology of the tree and the corresponding parameters. The algorithm automatically selected the ME5 and ME6 eigengenes (modules 5 and 6). Module eigengenes are used to build a classifier that distinguishes two or more classes. Each eigengene is a weighted average of the expression of all genes in the module, where the weight of each gene corresponds to its membership in the module. Each module might contain dozens to hundreds of genes, and hence the final classifier might depend on the expression of a large number of genes. In practice, it is desirable to reduce the number of necessary genes by a decision tree. The inferred decision tree had a relatively high predictive accuracy (Figure 4.6).

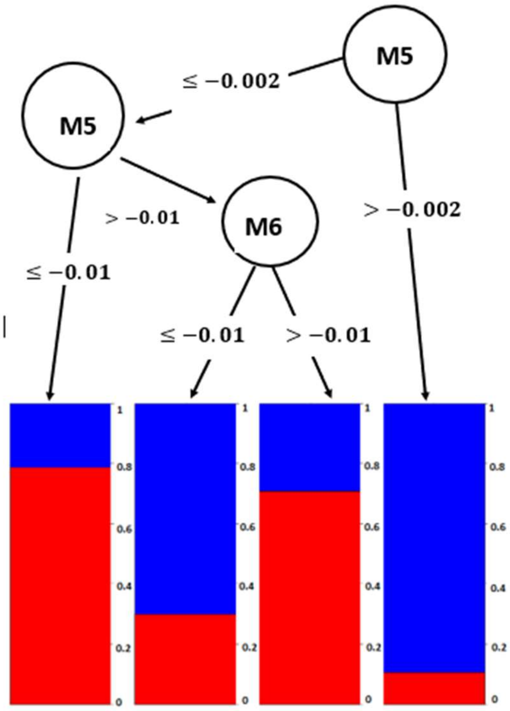


Figure 4.6 The decision tree for distinguishing MS from MS_INF cases. If the normalized eigengene of a case is greater than -0.002, it is classified as MS_INF. If it is less than -0.002 and less than -0.01, it is classified as MS. Otherwise, the ME6 eigengene determines whether the case is MS (>0.01) or MS_INF (≤−0.01). At the fixed thresholds shown above, this tree correctly classified 267 cases (84%) in the dataset.

4.2.5 Construction of PPI Network of Common DEGs for MS and MS_INF Treated Patients from two Approaches

We compared the resulted genes from Modules 5 and Module 6 to our 213 differentially expressed genes from SAM and Limma. 190 genes were common between the two methodological approaches, so we choose to keep the 213 DEGs and proceed by taking into account the extra 23 genes to construct the PPI network and examine for hub genes. The STRING App in the Cytoscape software was used to analyze 213 DEGs that had been entered into the STRING database. A total of 208 genes/nodes with 312 edges were enriched in the construction of the PPI network. (Figure 4.7)

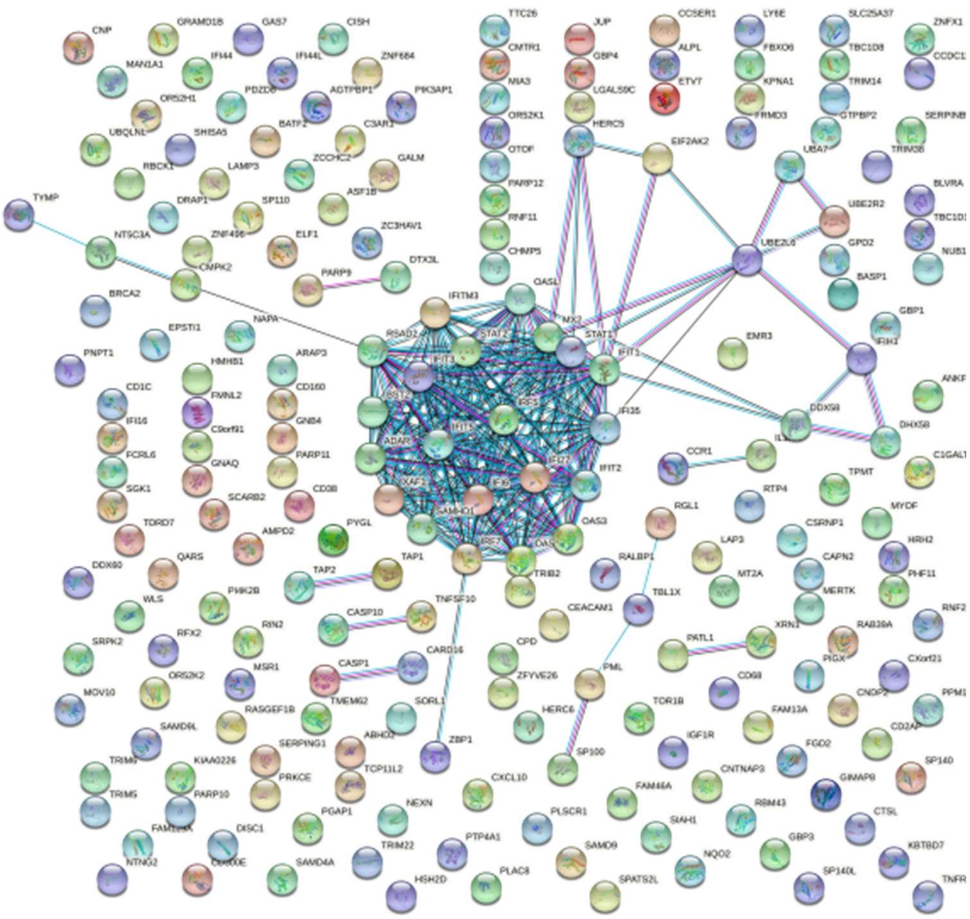


FIGURE 4.7 PPI NETWORK FROM 213 DIFFERENTIALLY EXPRESSED GENES

4.2.6 Critical Subnetworks and Identification of Hub Genes for MS and MS_INF Treated Patients

Hub genes were identified by 11 topological analysis methods from the CytoHubba, a Cytoscape plugin, where the top 20 genes were selected for each method. The 32 resulted genes (Table4.2) were found in the intersection of all methods and were selected as MS_INF related hub genes. We also obtained the clustering module with the highest score from PPI network of all DEGs (Figure 4.8 A) by MCODE algorithm. It was found that 21 genes from 32 hub genes were contained in this module (Figure 4.8 B) providing a minimal gene set toward potential clinical testing.

CytoHubba & MCODE: Hub genes by 11 topological analysis methods or Hub genes by CytoHubba and MCODE algorithm*
OAS3|RSAD2|IFIT3| IRF5|IFIT1|IFI6|IFIT5|OAS2|MX2|IFIT2|STAT2| ***IRF7|BST2|IFITM3|STAT1|ADAR|SAMHD1XAF1|IFI35|IFI27|OASL*** *|IFIH1|UBE2L6|IFI44|CCR1|NT5C3A| HERC5|CASP1| CMPK2|CXCL10| PARP9|DDX58*

TABLE4.2 IDENTIFICATION OF HUB GENES *21 HUB GENES CYTOHUBBA & MCODE (IN BOLD);32 GENES CYTOHUBBA

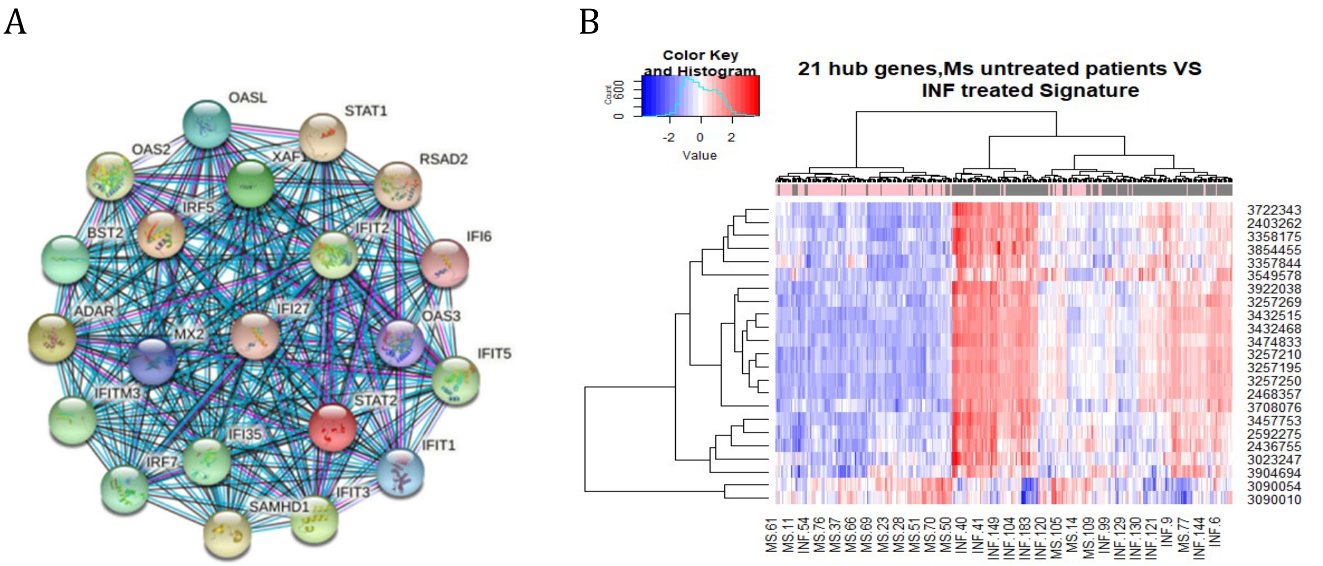


Figure 4.8 A) The highest score clustering module was generated by MCODE, with 21 genes.B) Unsupervised hierarchical clustering of MS untreated and IFN treated patients according to the expression of 21 FINAL IFN signature genes. The rows are different genes; the columns reflect different samples. The colored bar above the heatmap visualizes whether the patient was untreated (pink) or IFN treated (grey). Dark blue depicts low, red high expression.

4.2.7 Statistical Evaluation-Generalization

The resulting genomic signature of 21 hub genes is used to assess the classification and generalization ability of the model. The final gene signature arrived from GSE41850 dataset which was used as a training dataset (N = 224, MS_Utreated= 130, MS_INF Treated = 94) and testing dataset (N = 94, MS_Utreated= 55, MS_INF Treated = 39). The validation dataset GSE73608 as we mentioned in section 4.1 had two group of samples. Both groups were examined as independent validation sets. Twenty-one (21) hub genes served as features in training data set, and their corresponding gene expression profiles were obtained. Then, the classification model was established by support vector machine (SVM).

By applying 10fold cross-validation in the model, 76 out of the 94 samples were correctly classified, with a classification accuracy of 80%, model sensitivity to INF of 77%, specificity of 85%, and area under the ROC curve (AUC) was 0.86 (Figure 4.9 a). Furthermore, the established model was used to predict the samples in the validation data sets to test the prediction ability of the model.

In the first validation group, (N = 35, RRMS-untreated n = 25, RRMS_IFN responders n=10) the samples were classified, with a classification accuracy of 80%, moreover, the sensitivity was 100 % and specificity of the model was 64%, and the area under the receiver operating characteristic (ROC) curve was 0.92 (Figure 4.9 b). In the second validation group, (N = 50, SPMS_untreated n=30, SPMS_IFN treated n=20) the samples were classified, with a classification accuracy of 70%, the sensitivity was 90 % and specificity of the model was 57%, the area under the receiver operating characteristic (ROC) curve was 0.88 (Figure 4.9 c).

We merged the two groups and applied the model to the merged dataset with a classification accuracy of 75%, moreover, the sensitivity was 94 % and specificity of the model was 64%, the area under the receiver operating characteristic (ROC) curve was 0.90 (Figure 4.9 d). Compared to other studies and published results [18] the methodology performs very well and these results indicated that the diagnostic prediction model constructed in this study can effectively distinguish patients with MS from Interferon treated patients, and that the twenty one hub genes can be used as reliable biomarkers for MS diagnosis.

a

GSE41850	Real INF	Real MS	
Predict INF	42	6	
Predict MS	13	33	Totals
Totals	55	39	94
Correct	42	33	75
Sensitivity (%)	77		
Specificity (%)		85	
AUC	0.86		

b

GSE73608	Real INF	Real RRMS	
Predict INF	10	8	
Predict RRMS	0	17	Totals
Totals	10	25	35
Correct	10	17	27
Sensitivity (%)	100		
Specificity (%)		64	
AUC	0.92		

c

GSE73608	Real INF	Real SPMS	
Predict INF	18	13	
Predict SPMS	2	17	Totals
Totals	20	30	50
Correct	18	17	35
Sensitivity (%)	90		
Specificity (%)		57	
AUC	0.88		

d

GSE73608 (Merged)	Real INF	Real SPMS+RRMS	
Predict INF	28	20	
Predict SPMS+ RRMS	2	35	Totals
Totals	30	55	85
Correct	28	35	63
Sensitivity (%)	94		
Specificity (%)		64	
AUC	0.90		

FIGURE 4.9 CONSTRUCTION OF DIAGNOSTIC MODEL AND VALIDATION OF MODEL. A) CLASSIFICATION RESULTS AND ROC CURVES OF SAMPLES BY DIAGNOSTIC MODEL IN TRAINING DATA SET. B) CLASSIFICATION RESULTS AND ROC CURVES OF SAMPLES BY DIAGNOSTIC MODEL IN GSE73608 1ST GROUP. C) CLASSIFICATION RESULTS AND ROC CURVES OF SAMPLES BY DIAGNOSTIC MODEL IN GSE73608 2ND GROUP. D) CLASSIFICATION RESULTS AND ROC CURVES OF SAMPLES BY DIAGNOSTIC MODEL IN GSE73608 (1ST GROUP + 2ND GROUP.)

4.2.8 Biological interpretation

Each selected Affymetrix probe set was mapped to an annotation of Entrez Gene ID and Gene Symbol using the online tool WebGestalt (2013) (http://www.webgestalt.org/2013/). Considering the case untreated MS and INF treated, an over-representation analysis of the resulted 213-DEG-gene signature was performed in WebGestalt (2019). The enriched biological process categories are presented in Table 4.3A, whereas the enriched pathway categories are presented in Figure 4.10A.

Gene Set		P Value	FDR
GO:0098542	defense response to other organism	<2.2e-16	<2.2e-16
GO:0009615	response to virus	<2.2e-16	<2.2e-16
GO:0034340	response to type I interferon	<2.2e-16	<2.2e-16

GO:0043900	regulation of multi-organism process	6.6613E-16	1.42E-13
GO:0035456	response to interferon-beta	9.08E-14	1.54E-11
GO:0019058	viral life cycle	5.61E-13	7.95E-11
GO:0034341	response to interferon-gamma	1.81E-12	2.20E-10
GO:0035455	response to interferon-alpha	4.34E-11	4.61E-09
GO:0002831	regulation of response to biotic stimulus	5.81E-09	5.49E-07
GO:0032606	type I interferon production	1.15E-08	9.78E-07
GO:0060759	regulation of response to cytokine stimulus	8.85E-08	6.8406E-06
GO:0045088	regulation of innate immune response	1.10E-07	7.8222E-06
GO:0007249	I-kappaB kinase/NF-kappaB signaling	5.89E-07	0.000038516
GO:0001818	negative regulation of cytokine production	2.8E-06	0.00016929
GO:0002697	regulation of immune effector process	3.8E-06	0.0002148
GO:0000209	protein polyubiquitination	1.4E-05	0.00072239
GO:0001819	positive regulation of cytokine production	4.8E-05	0.0024223
GO:0061025	membrane fusion	7.8E-05	0.0036921
GO:0016050	vesicle organization	0.00011	0.0050468
GO:0002237	response to molecule of bacterial origin	0.00019	0.0080888
GO:0051701	interaction with host	0.0002	0.0086105
GO:0044764	multi-organism cellular process	0.0002	0.0086492
GO:0002764	immune response-regulating signaling pathway	0.0003	0.0099609
GO:0031349	positive regulation of defense response	0.0007	0.0246
Note: Common GO-biological processes between DEG-signatures and hub-gene-signatures are highlighted in blue.			

TABLE 4.3A GO-BIOLOGICAL PROCESSES ANALYSIS OF 213 DEGS IN THE CASE OF UNTREATED MS *VS* INF TREATED

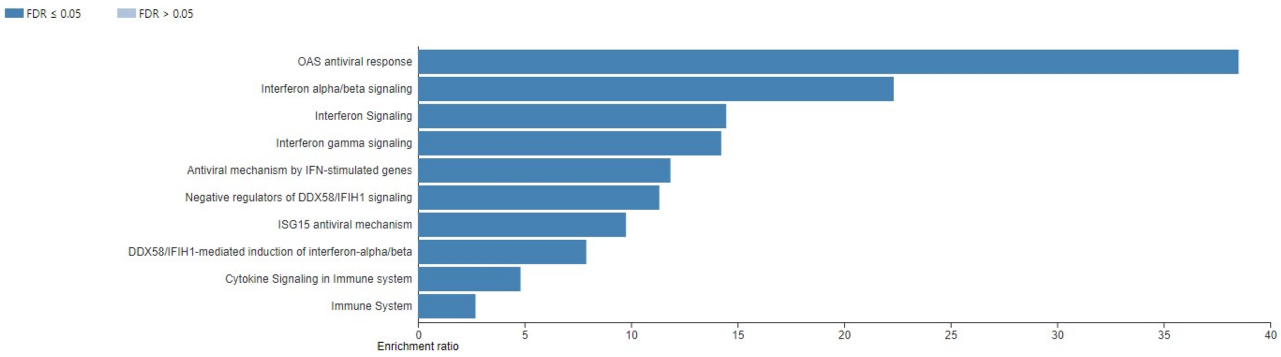


FIGURE 4.10A REACTOME PATHWAY ANALYSIS OF 213 DEGS IN THE CASE OF UNTREATED MS *VS* INF TREATED.

As demonstrated in Table 4.3A and Figure 4.10A, the enrichment analysis of the 213-DEG-gene signature revealed an overwhelming representation of immune processes and pathways, which are known to play a role in MS [35].As interferons are known to take part in regulating innate and adaptive immune responses, the excessive presentation of interferon signaling is rather expected [94].

Of note, in accordance with the original study, four out of 213 genes (*OAS3*, *RSAD2, EPSTI1, IFI44L*) were found among the *most significantly and strongly differentially expressed genes* between untreated MS and INF-treated patients [35].

Moreover, we also performed an over-representation analysis to further explore the functional information (biological processes, pathways) of the 21-hub-gene signature, which may be linked with MS interferon treatment. The analysis of the gene signature was performed using the online tool WebGestalt (2019).

Gene Set	Description	P Value	FDR
GO:0098542	defense response to other organism	<2.2e-16	<2.2e-16
GO:0009615	response to virus	<2.2e-16	<2.2e-16
GO:0034340	response to type I interferon	<2.2e-16	<2.2e-16
GO:0035455	response to interferon-alpha	1.42E-13	3.02E-11
GO:0035456	response to interferon-beta	4.01E-12	6.82E-10
GO:0043900	regulation of multi-organism process	2.14E-11	3.03E-09
GO:0019058	viral life cycle	7.67E-11	9.31E-09
GO:0034341	response to interferon-gamma	1.22E-10	1.29E-08
GO:0032069	regulation of nuclease activity	3.72E-06	0.0003513
GO:0032606	type I interferon production	0.0005205	0.044243
Note: Common GO-biological processes between DEG-signatures and hub-gene-signatures are highlighted in blue.			

TABLE 4.3B GO-BIOLOGICAL PROCESSES ANALYSIS OF THE 21-HUB-GENE SIGNATURE IN THE CASE OF UNTREATED MS *VS* INF TREATED

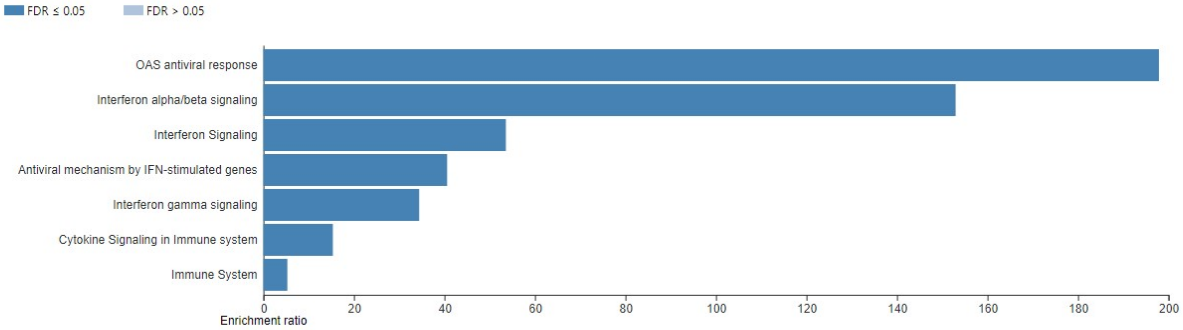
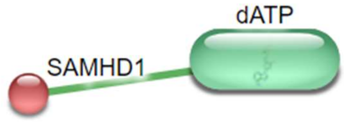
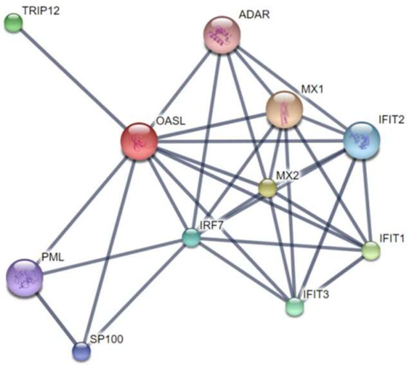


FIGURE 4.10B REACTOME PATHWAY ANALYSIS OF THE 21-HUB-GENE SIGNATURE IN THE CASE OF UNTREATED MS *VS* INF TREATED.

As presented in Figure 4.8A, our approach resulted in a clustering module (subnetwork) with 21 highly interconnected hub genes. After performing enrichment analysis, significant (P<0.01, FDR<0.001) GO biological processes (Table 4.3B), KEGG, Wikipathways, and Reactome (Figure 4.10B) pathways were obtained. The GO functional enrichment analysis indicated that the 21 MS_INF-associated genes were enriched in biological processes, such as response to type I interferon, defense response to other organism, response to virus, and response to interferon-alpha/beta/gamma. Furthermore, the pathway enrichment analysis (KEGG/Wikipathways/Reactome) showed that these genes were significantly enriched in signaling pathways, such as NOD-like receptor signaling, Toll-like receptor signaling, Cytokine Signaling in Immune system, Interferon Signaling, Interferon alpha/beta signaling, Type II/Type III interferon signaling, IL-10 Anti-inflammatory Signaling, but also Non-genomic actions of 1,25 dihydroxyvitamin D3 [95]. These findings highlight the affected processes and pathways in MS that are linked with a response to INF treatment and are in accordance with current knowledge [35, 96, 97, 98, 99, 100]. Moreover, eight out of 21 hub genes (*OAS2*, *IRF5*, *MX2*, *OASL*, *IFIT1*, *IRF7*, *IFI35* implicated in Interferon inducible and interferon pathway; *STAT1* implicated in Cell signaling) were found to be up-regulated in MS patients following IFNβ therapy [100] whereas two out of 21 genes (*OAS3*, *RSAD2*) were found, as 77forementioned, among the most significant differentially expressed genes between untreated MS and INF-treated patients, reported in the original study [35]. As depicted in Figure 4.8B, there is a variance in the expression pattern of some cases illustrating both the heterogeneity of the clinical course of MS and the partial response to IFNβ therapy. Thus, we suggest that the derived hub genes provide a reliable 21-hub-gene signature that could predict the response of IFNβ therapy in patients with MS [95].

4.2.9 Candidate drugs targeting hub genes

Using the DGIdb database we explore drug-gene interactions of the 21 hub genes that derived from the MCODE analysis. 21 genes were found common between CYTOHUBBA and MCODE. The drugs for possibly addressing patients in MS when they do not respond to INFb, are shown in Table4.4. We used the STITCH database, in order to construct downstream networks of the genes that have a drug relationship, to investigate the additional effects caused by inhibitors of these genes. All networks are also included in Table4.4. The network setting were "Experiments"," Databases","Coexpression" and confidence was set to high=0,9.

GENE SYMBOL	**DRUGS**	**NETWORK** Protein-protein interactions: Grey, Chemical-protein interactions: Green Interactions between chemicals: red.
SAMHD1		
OASL	- RIBAVIRIN	 TRIP12 ADAR MX1 IFIT2 OASL MX2 IRF7 IFIT1 PML IFIT3 SP100

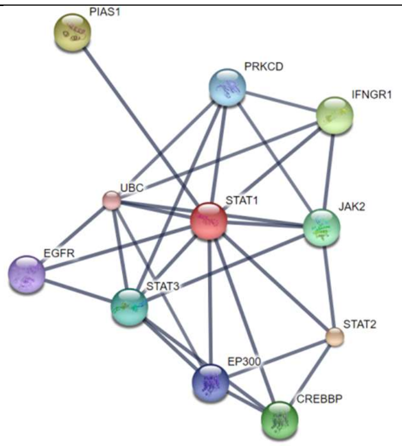
STAT1	- GARCINOL - GUTTIFERONE K - PICOPLATIN - CISPLATIN - CHEMBL85826 - IPRIFLAVONE	 PIAS1 PRKCD IFNGR1 UBC STAT1 JAK2 EGFR STAT3 STAT2 EP300 CREBBP

TABLE 4.4 GENES THAT HAVE DRUG INTERACTIONS AND INHIBITOR NETWORKS OF THE GENES THAT HAVE A DRUG RELATIONSHIP

4.3 Case untreated MS *vs* Healthy Controls

4.3.1 Dataset preprocessing and Differential expression

We have created a new file with all discovery samples, and we have 18.726 genes x 174 samples. Gene expression patterns were relatively stable across the three time points, so we adopted a cross-sectional analysis strategy.

In the discovery data set, a variance filter, difference between the 10% and 90% quantiles > 0.6, yielding 8.979 genes (104 > than original paper) was applied to normalized gene expression values in order to decrease the number of tested genes. Then group 1 (untreated patients) was compared to group 2 (averaged controls) at any of the three measured time points. The union of genes at all three time points passing the FDR cutoff of 0.01 were considered to be differentially expressed and assessed for differential expression in the replication data set. In the replication dataset the procedure was repeated: group 1 (untreated patients) was compared to group 2 (averaged controls) at all three measured time points, and the union of genes reaching a nominal p-value of 0.05 or smaller at any of these time points was considered to be replicated. Differentially expressed genes were identified in the discovery data set and then validated in the replication data set. In the discovery data set, differentially expressed genes were identified by applying stringent FDR-corrected P-value filters; these genes were then tested for validation in the replication data set.(R limma)
After applying gene filtering and differential analysis in each time point we have concluded in **8.979** genes from discovery dataset (FDR < 10-4) and their union yielded 76 significant genes. For controls, the two available time points were considered biological replicates and were averaged; these averaged expression profiles were then used for comparison with untreated MS patients for all three time points.

The replication dataset was tested based on these 8.979 genes and 2.270 genes were selected with p-value< 0.05. Based on the discovery and replication set, **76 genes** were common and were considered significant from our “MS Untreated- Control” case.

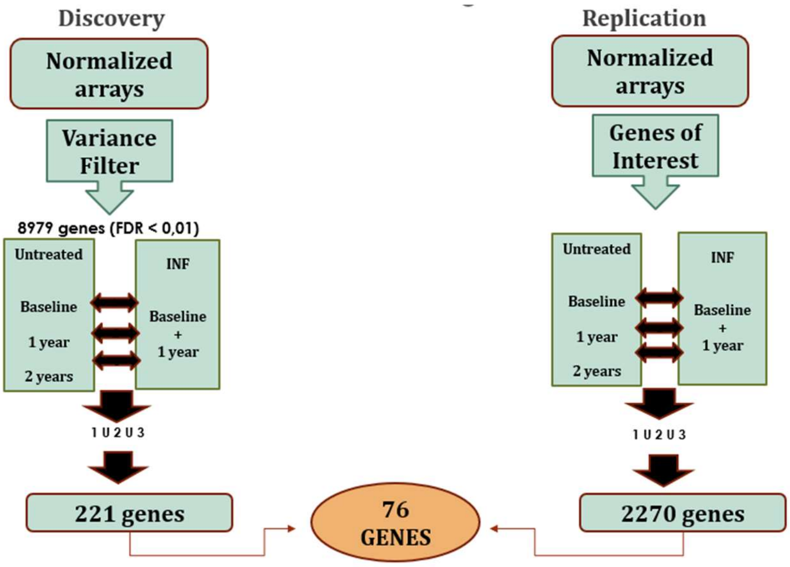


Figure 4.11 Dataset preprocessing and Differential expression: B) Untreated MS *vs* Healthy Controls (discovery and replication)

4.3.2 Significance Analysis of Microarrays (SAM)

As previously stated, in order to further evaluate the results, we conducted a Significance Analysis of Microarrays (SAM) on our filtered dataset so as to find differentially expressed genes based on T-statistats. The cutoff for significance is determined by a tuning parameter delta, chosen by the user based on the false positive rate. One can also choose a fold change parameter, to ensure that called genes change at least a pre-specified amount.

LIMMA and SAM	**MS Untreated vs Control**			**FINAL COMMON GENES**
	Discovery dataset (DEGs)	***Replication dataset (DEGs)***	***Common genes***	
LIMMA SAM	221 284	2270 1440	76 238	31

Table4.5 Differentially Expressed Genes MS Untreated vs Control

After the analysis with SAM, we compared the SAM results to our Limma analysis in R and we concluded in 31 genes considering the case Ms Untreated- Control.

We proceed with hierarchical clustering and k means clustering of our datasets to visualize and evaluate our results.

4.3.3 Clustering

In contrast with the transcriptional responses observed for IFN treatment, gene expression differences between untreated cases and controls were much more subtle. Despite modest differences in expression levels, the identified MS signature is discriminatory in unsupervised hierarchical clustering (Figure 4.11). The heatmap shows a uniform cluster of MS patients (group A) as well as several smaller uniform clusters of controls, an observation that stands in the replication set. MS cases who do not belong to group A, rather clustered with the controls (group B), indicating that gene expression changes evoked by the disease are much more heterogeneous and complex than those induced by IFN. Hierarchical clustering was performed using Euclidean distance and average clustering.

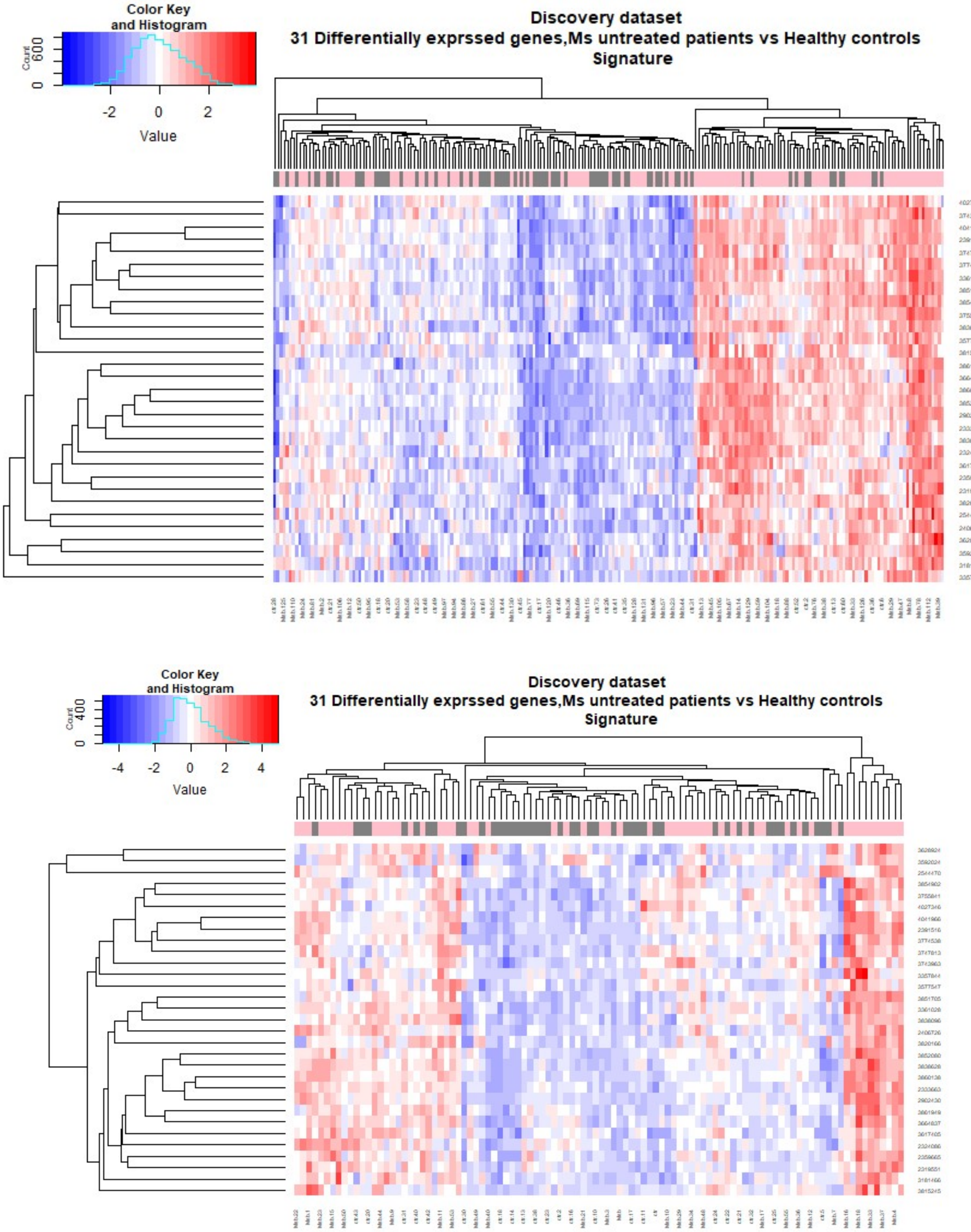


FIGURE 4.12 UNSUPERVISED HIERARCHICAL CLUSTERING OF MS PATIENTS AND HEALTHY CONTROLS ACCORDING TO THE EXPRESSION OF MS SIGNATURE GENES IN THE DISCOVERY (A) AND THE REPLICATION (B) DATA SETS. THE ROWS ARE DIFFERENT GENES, THE COLUMNS REFLECT DIFFERENT EXPERIMENTS. THE COLORED BAR ABOVE THE HEATMAP IDENTIFIES PATIENTS (PINK) AND CONTROLS (GREY). TWO SUBGROUPS OF MS PATIENTS, GROUP A WITH A STRONGER SIGNATURE AND GROUP B, EMERGE. BLUE DEPICTS LOW EXPRESSION AND RED HIGH EXPRESSION.

4.3.4 Statistical Evaluation

The resulting genomic signature of 31 hub genes derived from GSE41850 dataset and we proceed to examine the classification and generalization ability of it in an independent dataset. We have acquired Dataset E-MTAB-5151 [92] downloaded from the ArrayExpress database. This dataset was established on the platform of A-AFFY-44-Affymetrix Gene Chip Human Genome U133 Plus 2.0 [HG-U133_Plus_2]. The final gene signature from GSE41850 dataset was used as a training dataset (N = 150, MS_Utreated= 94, Healthy controls = 56) and testing dataset (N = 62, MS_Utreated= 39, MS_INF Treated = 23)**.** Thirty-one (31) genes served as features in training data set, and their corresponding gene expression profiles were obtained. Then, the classification model was established by support vector machine (SVM).

By applying 10fold cross-validation in the model, 38 out of the 62 samples were correctly classified, with a classification accuracy of 63%, model sensitivity to CTR of 65%, specificity of 80%, and area under the ROC curve (AUC) was 0.78 (Figure 4.13 a). Furthermore, the established model was used to predict the samples in the validation data sets to test the prediction ability of the model.

In the validation group, (N = 76, MS-untreated n = 49, Healthy controls n=27) the samples were classified, with a classification accuracy of 79%, moreover, the sensitivity was 55 % and specificity of the model was 98%, and the area under the receiver operating characteristic (ROC) curve was 0.69 (Figure 4.13 b).

a

GSE41850	Real CTR	Real MS	
Predict CTR	8	8	
Predict MS	15	31	Totals
Totals	23	39	62
Correct	8	31	39
Sensitivity (%)	65		
Specificity (%)		80	
AUC	0.78		

b

E-MTAB-5151	Real CTR	Real MS	
Predict CTR	12	1	
Predict MS	15	48	Totals
Totals	27	49	76
Correct	12	48	60
Sensitivity (%)	55		
Specificity (%)		98	
AUC	0.69		

Figure 4.13 Construction of diagnostic model and validation of model. a) Classification results and ROC curves of samples by diagnostic model in training data set. b) Classification results and ROC curves of samples by diagnostic model in E-MTAB-5151.

As we can see from the results, the algorithm performs well when classifying MS patients based on the expression values of the datasets**.** As we have already mentioned gene expression changes evoked by the disease are much more heterogeneous and complex and thus, we choose to examine each MS stage *versus* Healthy control samples in order to identify candidate genes that could be indicative of the disease progression as well as the prediction of new samples. In section 4.4 we present the application of our methodology in a new data set with MS cases in different stages and Healthy control samples.

4.3.5 Biological Interpretation

Each selected Affymetrix probe set was mapped to an annotation of Entrez Gene ID and Gene Symbol using the online tool WebGestalt (2013) (http://www.webgestalt.org/2013/). Considering the case untreated MS and controls, an over-representation analysis of the resulted 31-DEG-gene signature was performed in WebGestalt (2019). The results of the enrichment analysis of the 31-DEG-gene signature are presented in Table 4.6 and Figure 4.14.

Gene Set	Description	P Value	FDR
GO:0002446	Neutrophil mediated immunity	1.29E-05	0.0058
GO:0036230	granulocyte activation	1.36E-05	0.0058

TABLE 4.6 GO-BIOLOGICAL PROCESSES ANALYSIS OF 31 DEGS IN THE CASE OF UNTREATED MS *VS* CONTROLS

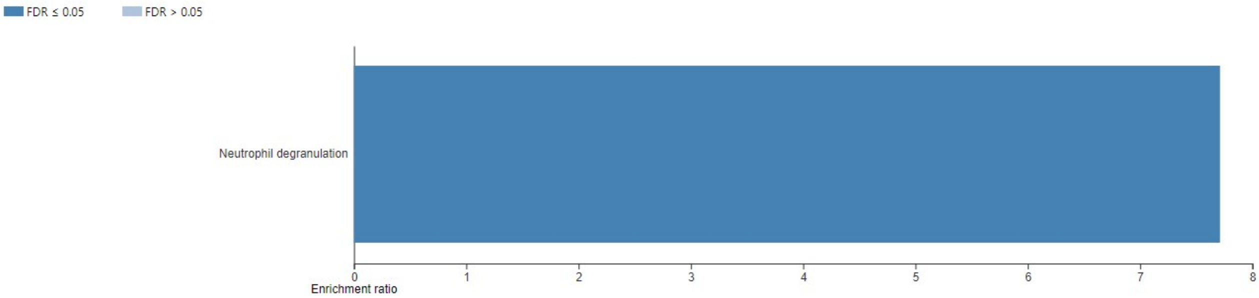


FIGURE 4.14 REACTOME PATHWAY ANALYSIS OF 31 DEGS IN THE CASE OF UNTREATED MS *VS* CONTROLS.

The depicted pathway and biological processes become more important as our knowledge about neutrophils, first-responding innate myeloid cells, and their effector functions as contributing components in the pathogenesis of MS is increasing. Naegele et al [??] showed that neutrophils in MS patients are more numerous and exhibit a primed state that is based, among others, on enhanced degranulation and oxidative burst. [paper in preparation]

In order to reveal more information about MS, we sought to identify the DEGs among the distinct stages of MS *versus* healthy controls, as aforementioned.

4.4 Cases Untreated MS patients in different disease stages *vs* Healthy Controls

In this section we will present the application of our pipeline on a new dataset, GSE136411. There are three clinical courses of MS. The most frequent is the relapsing-remitting form (RRMS), which accounts for approximately 85% of MS cases. RRMS is characterized by relapse followed by remission, where symptoms may vary from mild to severe, and relapses and remissions may last for days or months. After a variable time, most individuals with RRMS advance to a secondary progressive phase (SPMS), where neurologic worsening occurs without periods of remission. In contrast, 15% of individuals with MS experience a progressive course, called primary progressive MS (PPMS), which is characterized by a steady worsening of neurologic functioning, without any distinct relapses or periods of remission. For PPMS, the rate of progression may vary over time, with occasional plateaus or temporary improvements, but the progression is continuous. [90]

The cases that we have examined are:

i. Relapsing-Remitting MS (RRMS) vs Healthy Controls
ii. Primary Progressive MS (PPMS) vs Healthy Controls
iii. Secondary progressive MS (SPMS) vs Healthy Controls

4.4.1 Case Relapsing-Remitting MS (RRMS) *vs* Healthy Controls

After normalization, our dataset consists of 188 samples (RRMS N=121 and HC N=67) and 10.160 gene with their expression values.

4.4.1.1 Significance Analysis of Microarrays (SAM)

We conducted a Significance Analysis of Microarrays (SAM) on our filtered dataset, so as to find differentially expressed genes based on T-statistists. The cutoff for significance is determined by a tuning parameter delta, chosen by the user based on the false positive rate. One can also choose a fold change parameter, to ensure that called genes change at least a pre-specified amount.

SAM	**RRMS Untreated vs Control**		
	upregulated	***downregulated***	***FINAL GENES***
	35	95	130

Table4.7 Differentially Expressed Genes RRMS Untreated *vs* Control

After the analysis with SAM, we concluded in 130 Differentially expressed genes considering the case RRMS Untreated-Control. We proceed with the Pigengene methodology

4.4.1.2 Pigengene Methodology

Our goal is to find a minimum set of significant genes that will be able to predict the state of a new sample as well as provide meaningful biological information through the correlation and combination of genes in pathways and smaller groups/networks. We apply the Pigengene methodology streps on the 10.160 genes that derived from the preprocessed dataset.

i. Weighted correlation network: Weighted Coexpression network analysis (WGCNA) was applied to group related genes into gene modules (clusters) based on their coexpression patterns in MS.

ii. Eigengenes: We computed an eigengene for each module as a weighted average of the expression of all genes in that module. (Figure 4.15 B). To validate the association of the modules to each state, we modeled the probabilistic dependencies between the eigengenes using a BN (Figure 4.16). We used Bayesian networks as probabilistic predictive models to determine the state.

A B

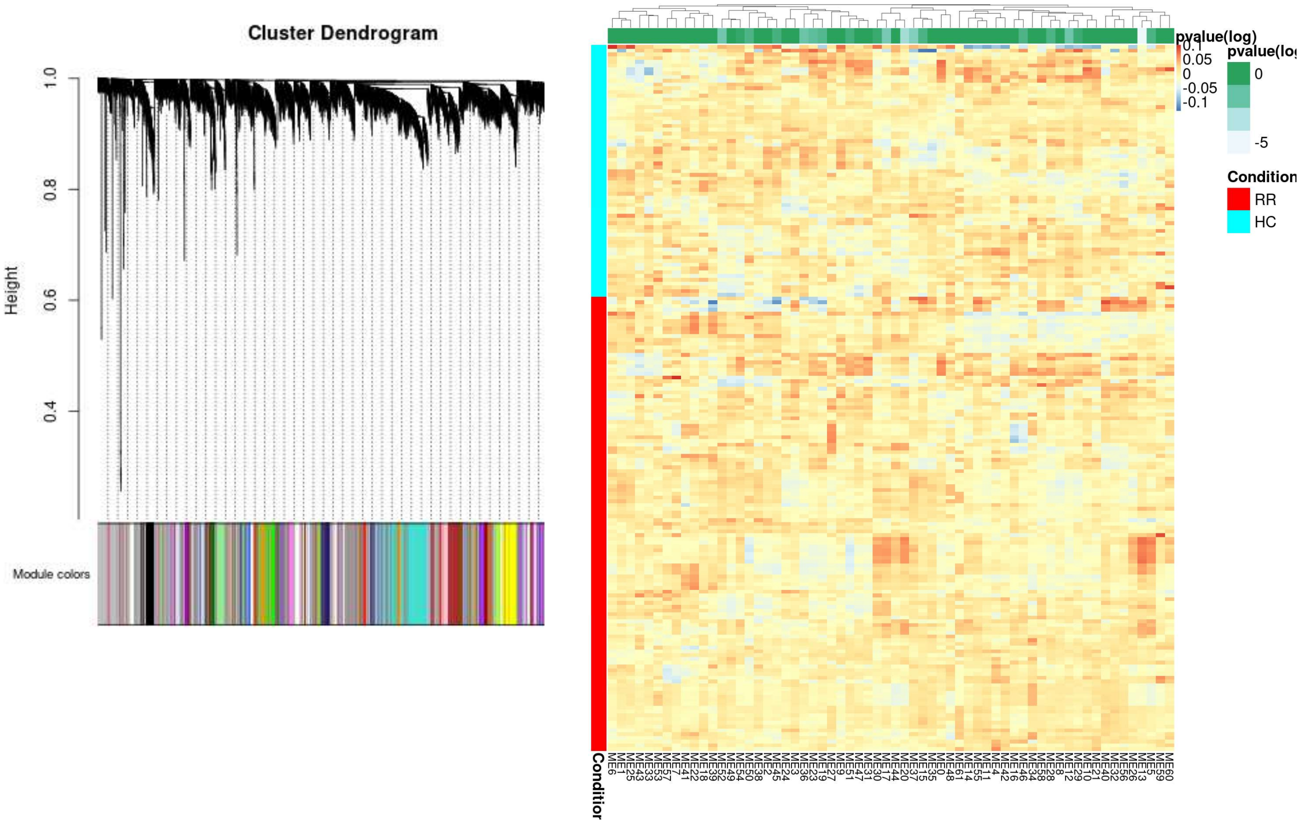


FIGURE 4.15 A) MODULES DENDROGRAM B) THE EIGENGENES HEATMAP. WE COMPUTED AN EIGENGENE FOR EACH MODULE AS A WEIGHTED AVERAGE OF THE EXPRESSION OF ALL GENES IN THAT MODULE. THE INTENSITY OF THE COLORS IN EACH HEATMAP CORRESPONDS TO THE NORMALIZED AVERAGE EXPRESSION. EACH COLUMN CORRESPONDS TO AN EIGENGENE. EACH ROW SHOWS THE EXPRESSION OF A CASE FROM THE MICROARRAY DATASET. THE EIGENGENE THAT IS DIFFERENTIALLY EXPRESSED IS ME13.

iii. Bayesian network: We fitted a Bayesian network to the eigengenes to determine the relationships of the modules with each other and with the state of the samples. Descendants of the “Disease” node, the variable that models the state, show high dependency between these eigengenes and the state type and suggest that they have useful biological information that can explain the differences between the two states. We trained a Bayesian network to model the probabilistic dependencies between the modules. Several individual networks from random staring networks were built (no.1000) by optimizing their score. Then, we inferred a consensus network from the ones with relatively “higher” scores. The default hyper-parameters and arguments are then selected. Each module eigengene is represented by a node (observed random variable). To model the condition, we added “Disease” as an observed random variable to the network.

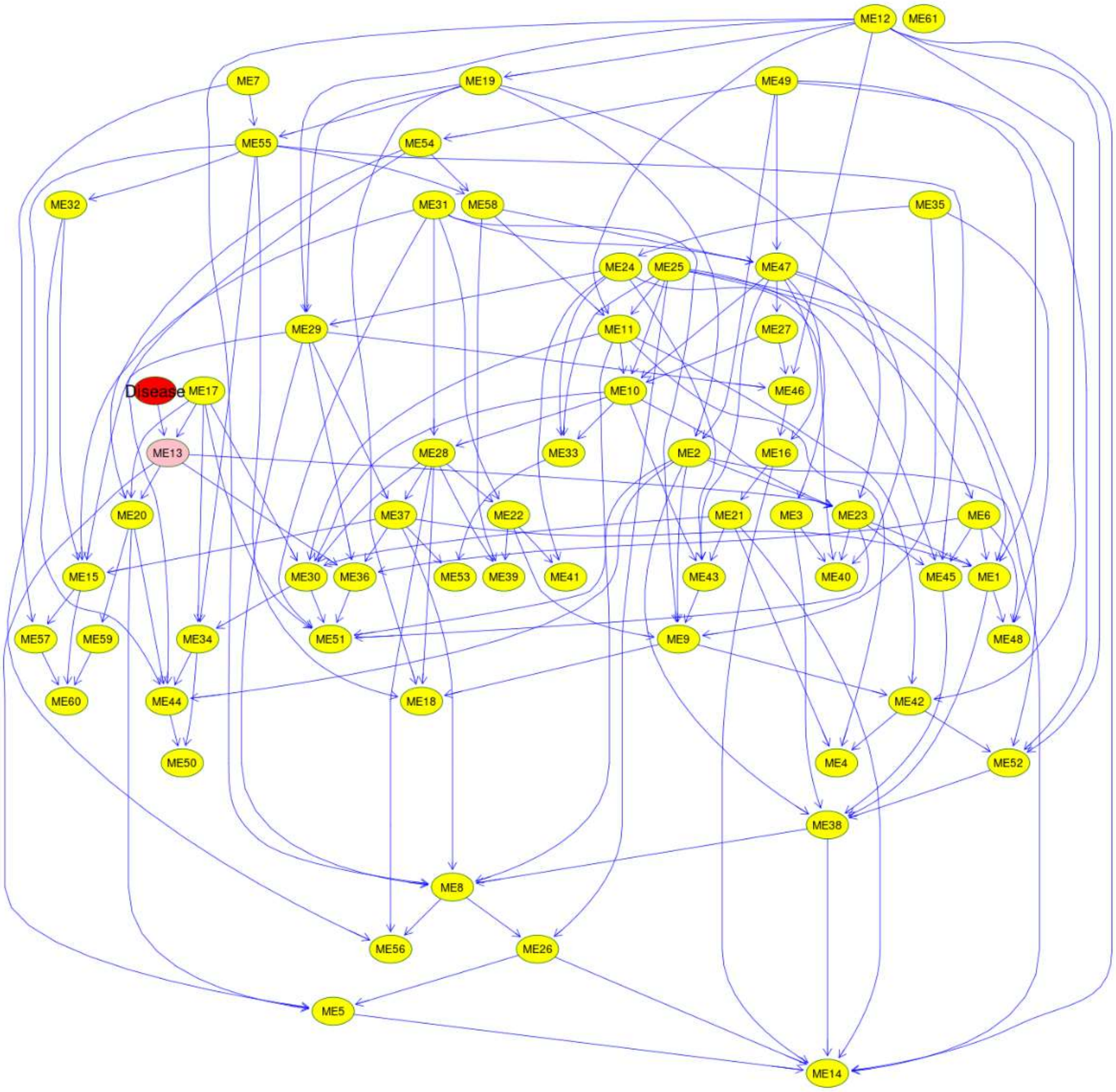


FIGURE 4.16 THE BAYESIAN NETWORK FITTED TO THE EIGENGENES. EACH NODE REPRESENTS AN EIGENGENE OF A MODULE. THE ARCS MODEL THE PROBABILISTIC DEPENDENCIES BETWEEN THE MODULES. THE "DISEASE" NODE IS SET TO 1 FOR RRMS AND 0 FOR HEALTHY CONTROLS, AND ITS CHILDREN ARE HIGHLIGHTED IN PINK

iv. Decision tree: A decision tree was not fitted to the model due to the Bayesian network results; we investigated the gene signature of Module 13.

4.4.1.3 Construction of PPI Network of Common DEGs for RRMS and Healthy Controls from two Approaches

We compared the resulted 185 genes from Modules 13 to the 130 differentially expressed genes from SAM. All 130 genes were common between the two methodological approaches, so we choose to keep the additional 55 genes and we proceed to construct the PPI network genes, using the STRING App in the Cytoscape software, with the 185 DEGs and examine for hub. (Figure 4.17) A total of 179 genes/nodes with 43 edges were enriched in the construction of the PPI network.

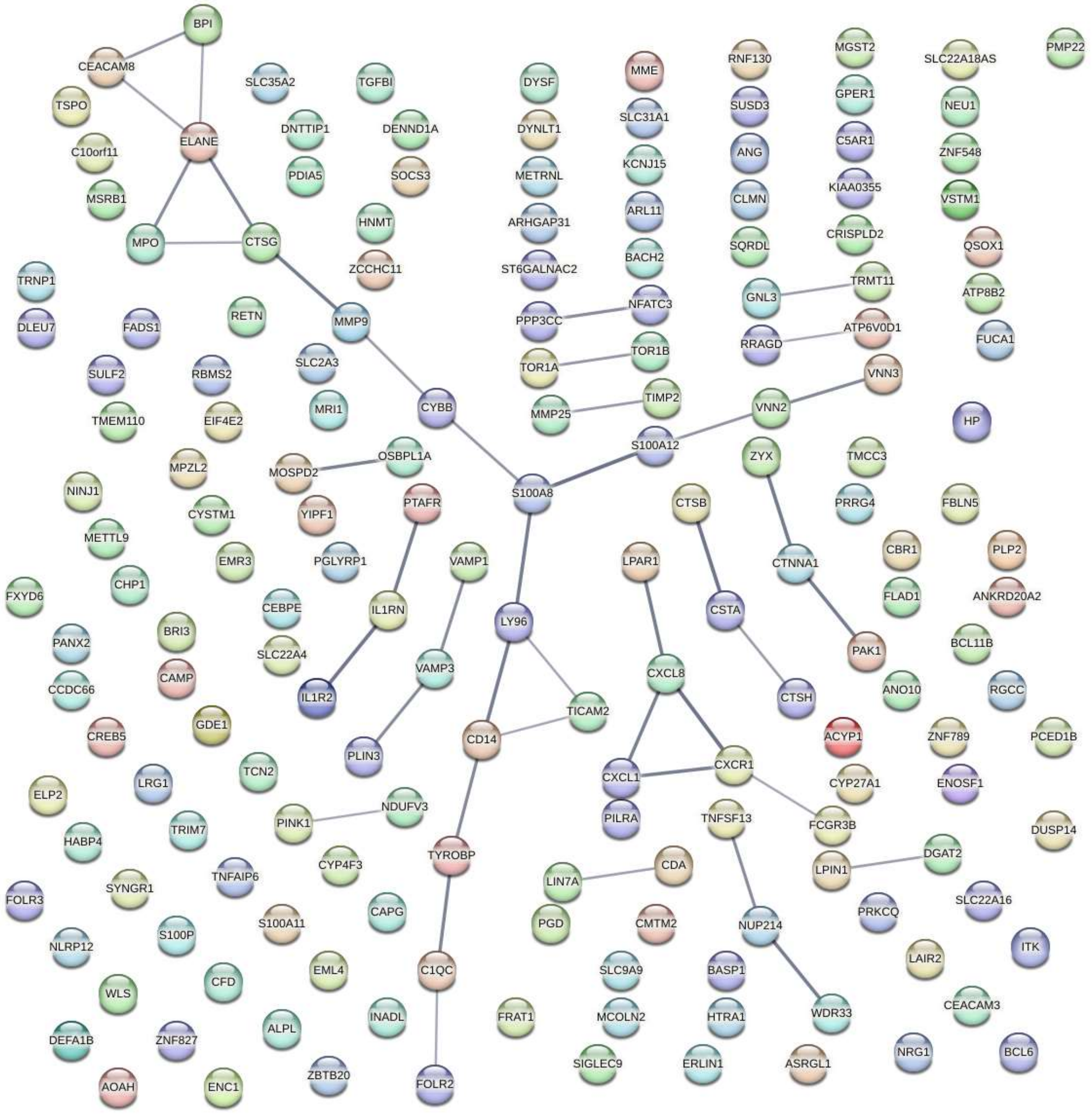


FIGURE 4.17 PPI NETWORK FROM 185 DIFFERENTIALLY EXPRESSED GENES

4.4.1.4 Critical Subnetworks and Identification of Hub Genes for Primary Progressive MS (RRMS) vs Healthy Controls Patients

Hub genes were identified by 11 topological analysis methods from the CytoHubba, a Cytoscape plugin, where the top 20 genes were selected for each method. The 44 resulted genes (Table4.8) were found in the intersection of all methods and were selected as RRMS related hub genes, providing a minimal gene set toward potential clinical testing. We also obtained one clustering module of 16 genes with the highest score from the PPI network of all DEGs (Figure 4.18) by MCODE algorithm. It was found that 8 genes from 16 were included in 44 hub genes contained in this module (Table4.8)

CytoHubba & MCODE: Hub genes by 11 topological analysis methods or Hub genes by CytoHubba and MCODE algorithm*
*IL1RN/C1QC/CSTA/CTNNA1/**CTSG**/CTSH/**CXCL8**/CXCR1/**ELANE**/IL1R2/LY96/**MMP9**/MOSPD2/**MPO**/OSBPL1A/PAK1/PTAFR/S100A/TIMP2/WDR33/FBLN5/ARL11/ATP8B2/BACH2/C10orf11/**CAMP**/CEACAM3/CYP4F3/DUSP14/DYSF/HNM/PCED1B/PINK1/SLC22A16/TNFSF13/TRNP1/ZNF789/LPAR1/**S100A12**/TNFAIP6/ZYX/CD14/**CXCL1**/MMP25*

Table4.8 Identification Of Hub Genes *8 hub genes Cytohubba & MCODE (in bold);44 genes Cytohubba

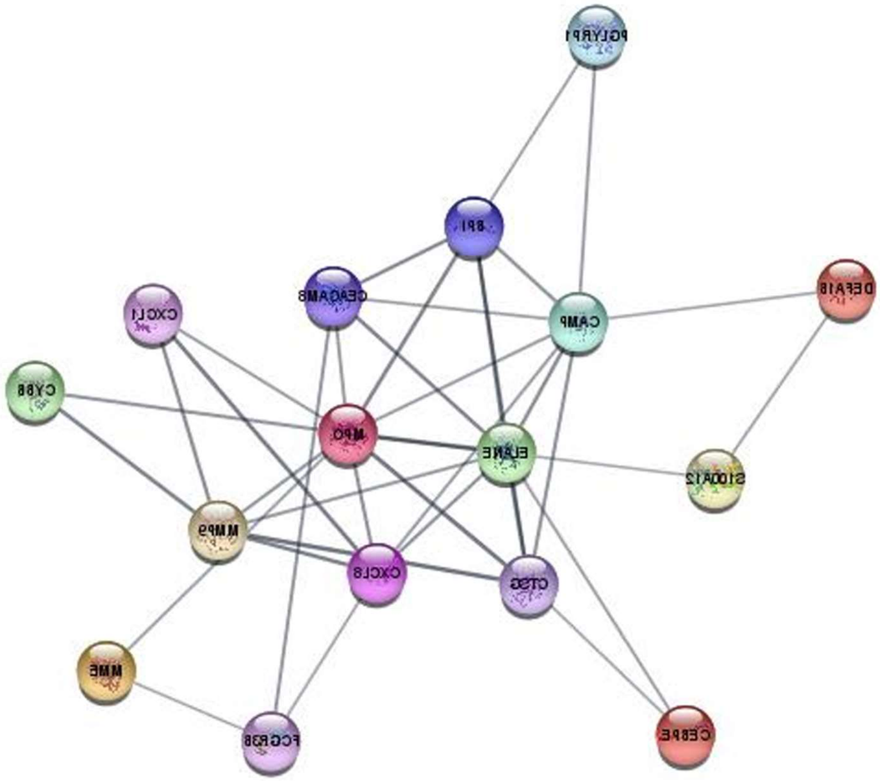


Figure 4.18 The highest score clustering module was generated by MCODE, with 16 genes.

4.4.1.5 Statistical Evaluation

The resulting genomic signature of 44 hub genes is used to assess the classification and generalization ability of the model. The final gene signature arrived from GSE136411 dataset which was used as a training dataset (N = 132, RRMS n= 85, CTR n=47) and testing dataset (N = 56, RRMS n= 36, CTR n=20). Dataset E-MTAB-4890 was used to access the generalization ability of the resulted gene signature as an independent dataset. It consists of (N = 92, RRMS n = 52, CTR n=40). Then, the classification model was established by support vector machine (SVM).

By applying 10fold cross-validation in the model, 46 out of the 56 samples were correctly classified, with a classification accuracy of 83%, model sensitivity to CTR of 0.60%, specificity of 94%, and area under the ROC curve (AUC) was 0.83 (Figure 4.19 a). Furthermore, the established model was used to predict the samples in the validation data sets to test the prediction ability of the model.
In the validation group the samples were classified, with a classification accuracy of 78%, moreover, the sensitivity was 65 % and specificity of the model was 94%, and the area under the receiver operating characteristic (ROC) curve was 0.84 (Figure 4.19 b).

a

GSE136411	Real CTR	Real RRMS	
Predict CTR	12	2	
Predict MS	8	32	Totals
Totals	20	34	54
Correct	12	32	44
Sensitivity (%)	60		
Specificity (%)		94	
AUC	0.83		

b

E-MTAB-4890	Real CTR	Real RRMS	
Predict CTR	22	3	
Predict MS	18	49	Totals
Totals	30	52	82
Correct	22	49	71
Sensitivity (%)	65		
Specificity (%)		94	
AUC	0.84		

FIGURE 4.19 CONSTRUCTION OF DIAGNOSTIC MODEL AND VALIDATION OF MODEL. A) CLASSIFICATION RESULTS AND ROC CURVES OF SAMPLES BY DIAGNOSTIC MODEL IN TRAINING DATA SET. B) CLASSIFICATION RESULTS AND ROC CURVES OF SAMPLES BY DIAGNOSTIC MODEL IN E-MTAB-4890

The diagnostic prediction model constructed in this study can effectively distinguish patients in relapsing remitting stage of the disease and that the 33 out of 44 hub genes can be used as reliable biomarkers for RRMS diagnosis.

4.4.1.6 Biological interpretation

Each selected Illumina probe set was mapped to an annotation of Entrez Gene ID and Gene Symbol using the online tool WebGestalt (2013) (http://www.webgestalt.org/2013/). Considering the case RRMS *versus* controls, an over-representation analysis of the resulted 130-DEG-gene signature was performed in WebGestalt (2019). The enriched biological process categories are presented in Table 4.9A, where the enriched pathway categories are presented in Figure 4.20A

Gene Set	Description	P Value	FDR
GO:0036230	granulocyte activation	<2.2e-16	<2.2e-16
GO:0002446	neutrophil mediated immunity	<2.2e-16	<2.2e-16
GO:0002237	response to molecule of bacterial origin	4.15E-07	0.00012
GO:0006959	humoral immune response	8.51E-07	0.00017
GO:0001819	positive regulation of cytokine production	1E-06	0.00017
GO:0009620	response to fungus	1.9E-05	0.00269
GO:0006766	vitamin metabolic process	3.3E-05	0.00403
GO:0050727	regulation of inflammatory response	3.8E-05	0.00404
GO:0071706	tumor necrosis factor superfamily cytokine production	4.8E-05	0.00458
GO:0042107	cytokine metabolic process	9E-05	0.00755
GO:0098542	defense response to other organism	9.8E-05	0.00755
GO:0050900	leukocyte migration	0.00014	0.00962
GO:0071216	cellular response to biotic stimulus	0.00015	0.00962
GO:0031348	negative regulation of defense response	0.00031	0.01868
GO:0007249	I-kappaB kinase/NF-kappaB signaling	0.00042	0.02287

GO:0043062	extracellular structure organization	0.00043	0.02287
GO:0060191	regulation of lipase activity	0.00047	0.02287
GO:0051047	positive regulation of secretion	0.00048	0.02287
GO:0050663	cytokine secretion	0.00063	0.0284
GO:0006732	coenzyme metabolic process	0.00075	0.03043
GO:0001906	cell killing	0.00075	0.03043
GO:0060326	cell chemotaxis	0.00086	0.03314
Note: Common GO-biological processes between DEG-signatures and hub-gene-signatures are highlighted in blue.			

TABLE 4.9A GO-BIOLOGICAL PROCESSES ANALYSIS OF 130 DEGS IN THE CASE OF RRMS *VS* CONTROLS

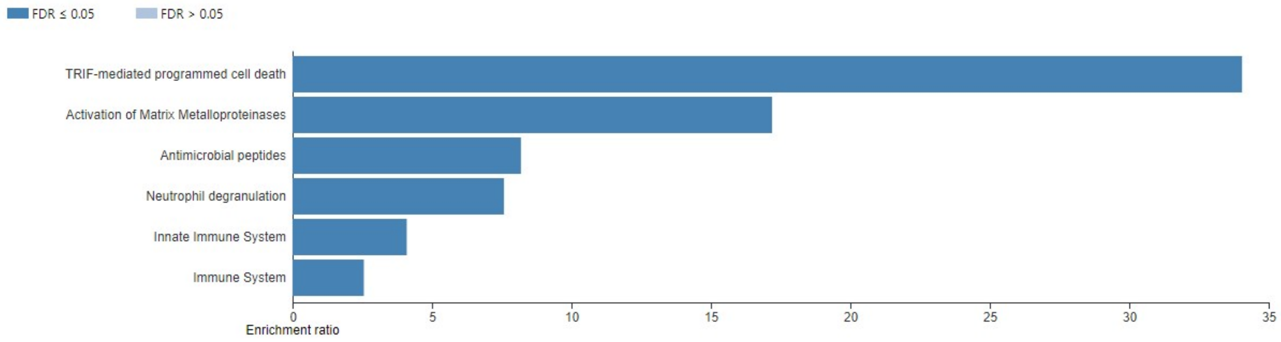


FIGURE 4.20A. REACTOME PATHWAY ANALYSIS OF 130 DEGS IN THE CASE OF RRMS *VS* CONTROLS

Moreover, we also performed an over-representation analysis to further explore the functional information (biological processes, pathways) of the 16-hub-gene signature, which may be more specific to RRMS stage. The analysis of the gene signature was performed using the online tool WebGestalt (2019).

Gene Set	Description	P Value	FDR
GO:0036230	granulocyte activation	<2.2e-16	<2.2e-16
GO:0002446	neutrophil mediated immunity	<2.2e-16	<2.2e-16
GO:0006959	humoral immune response	2.64E-11	7.49E-09
GO:0098542	defense response to other organism	5.40E-09	1.1E-06
GO:0009620	response to fungus	1.29E-07	2.2E-05
GO:0035821	modification of morphology or physiology of other organism	3.25E-07	4.2E-05
GO:0001906	cell killing	3.46E-07	4.2E-05
GO:0002237	response to molecule of bacterial origin	4.94E-07	5.3E-05
GO:0043900	regulation of multi-organism process	9.21E-07	8.7E-05
GO:0050900	leukocyte migration	2E-06	0.00017
GO:0051702	interaction with symbiont	5.2E-05	0.00403

GO:0071216	cellular response to biotic stimulus	5.9E-05	0.00417
GO:0045926	negative regulation of growth	9.3E-05	0.00611
GO:0042107	cytokine metabolic process	0.00015	0.00925
GO:0050727	regulation of inflammatory response	0.00038	0.02171
GO:0045730	respiratory burst	0.00041	0.02177
GO:0043062	extracellular structure organization	0.00056	0.02825
Note: Common GO-biological processes between DEG-signatures and hub-gene-signatures are highlighted in blue.			

TABLE 4.9B GO-BIOLOGICAL PROCESSES ANALYSIS OF THE 16-HUB-GENE SIGNATURE IN THE CASE OF RRMS *VS* CONTROLS

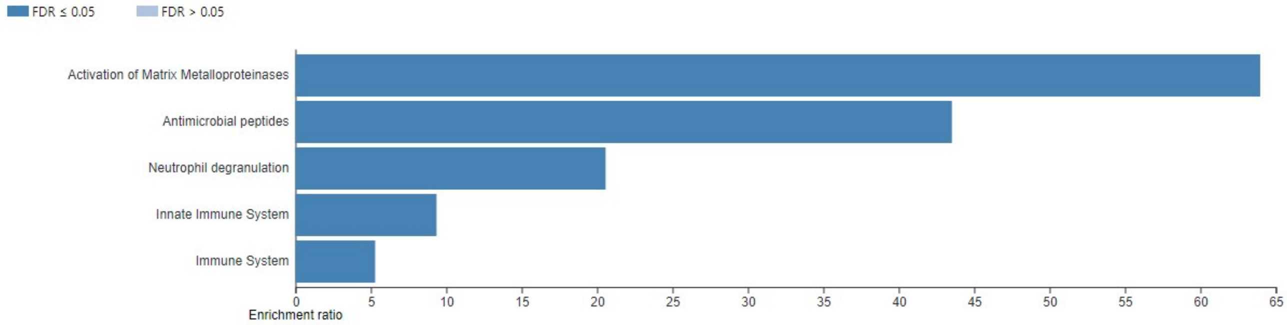


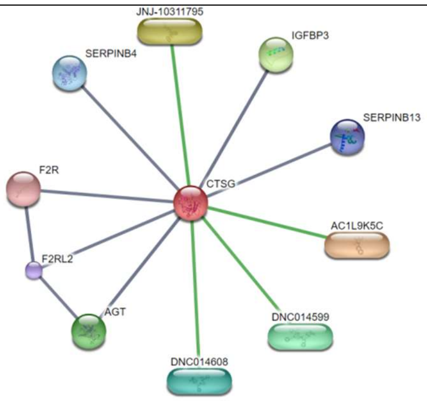
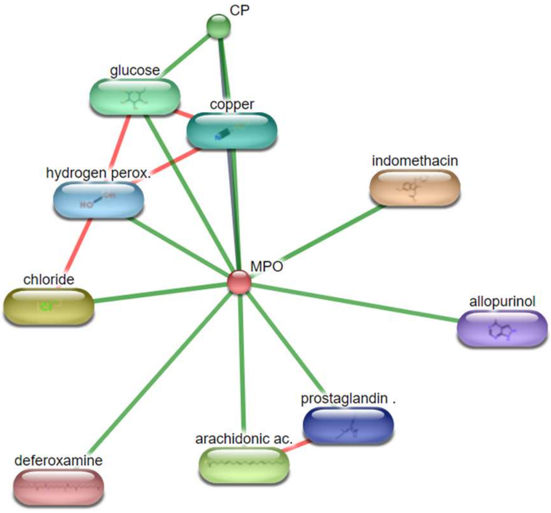
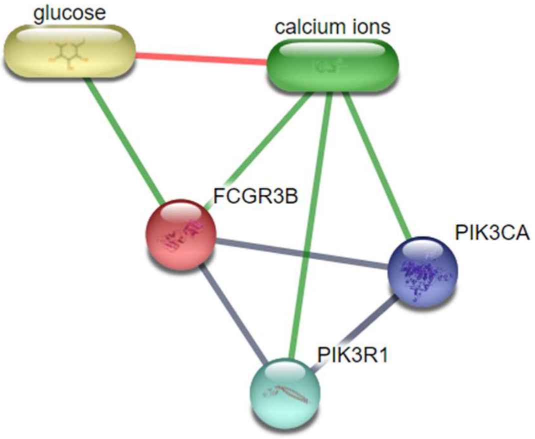
FIGURE 4.20B REACTOME PATHWAY ANALYSIS OF THE 16-HUB-GENE SIGNATURE IN THE CASE OF RRMS *VS* CONTROLS

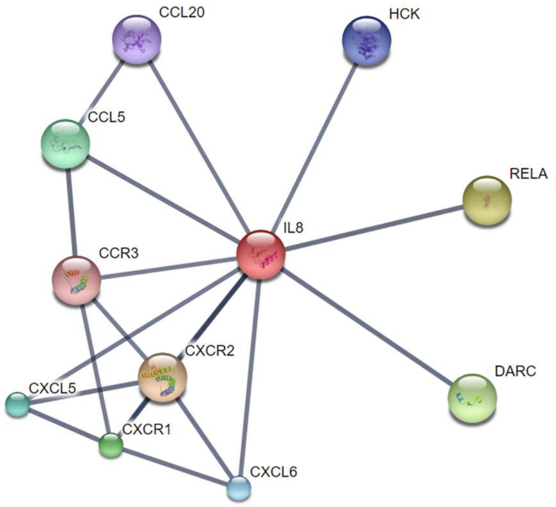
As illustrated in above tables (Table 4.9A, 4.9B) and figures (Figures 4.20A, 4.20B), both the 130-gene signature and the 16-hub-gene signature provide overlapped processes and pathways related to immune system processes. This is an expected finding, since MS is characterized by immune dysregulation, which results in the infiltration of the CNS by immune cells, triggering demyelination, axonal damage, and neurodegeneration [101]. Interestingly, these GO biological processes have greater statistical significance in the 16-hub-gene signature, while the opposite happens with the Reactome pathways. In addition, the unique pathway (antimicrobial peptides) and GO biological processes (eight) of the 16-hub-gene signature provide a narrower range of immune system components and mechanisms. [paper in preparation]

4.4.1.7 Candidate drugs targeting hub genes

Using the DGIdb database we explore drug-gene interactions of the 16 hub genes that derived from the MCODE analysis. 8 genes out of 16, were found common between CYTOHUBBA and MCODE. The drugs for possibly addressing patients in the Relapsing Remitting stage of MS are shown in Table4.10. We used the STITCH database, in order to construct downstream networks of the genes that have a drug relationship, to investigate the additional effects caused by inhibitors of these genes. All networks are also included in Table4.10. The network setting were “Experiments”,” Databases”,” Coexpression” and confidence was set to high=0,9.

GENE SYMBOL	**DRUGS**	**NETWORK** Protein-protein interactions: Grey Chemical-protein interactions: Green Interactions between chemicals: Red

GENE SYMBOL	DRUGS	**NETWORK** Protein-protein interactions: Grey Chemical-protein interactions: Green Interactions between chemicals: Red
CTSG	• MANNITOL • CHEMBL374027	 JNJ-10311795 IGFBP3 SERPINB4 SERPINB13 F2R CTSG AC1L9K5C F2RL2 AGT DNC014599 DNC014608
MPO	• DIMETHYL SULFOXIDE • PSORALEN • TOLMETIN • DICLOFENAC • DOXYCYCLINE • ASULACRINE • NIMESULIDE • PYRAZINAMIDE • PROPYLTHIOURACIL • FLUDARABINE • LORATADINE • OCTREOTIDE • TRIMETHOPRIM • THEOPHYLLINE • LITHIUM • LIDOCAINE • TENECTEPLASE • FLUTAMIDE • FENTANYL	 CP glucose copper hydrogen perox. indomethacin chloride MPO allopurinol prostaglandin . arachidonic ac. deferoxamine
FCGR3B	• PREDNISOLONE • ALDESLEUKIN • METHIMAZOLE • THALIDOMIDE • FENTANYL • SODIUM CHLORIDE • PUROMYCIN • EPOETIN ALFA • CYCLOSPORINE • INDOMETHACIN • MAFOSFAMIDE • PROGESTERONE • CHOLECALCIFEROL • METHOTREXATE • PENICILLIN G POTASSIUM • DOXORUBICIN • HEPARIN • LACTULOSE • GELDANAMYCIN	

GENE SYMBOL	**DRUGS**	**NETWORK** Protein-protein interactions: Grey Chemical-protein interactions: Green Interactions between chemicals: Red
CXCL8/IL8	• ABX-IL8 • HUMAX-IL8 • LEFLUNOMIDE • YANGONIN • E319 • FOSCARNET • NAPROXEN • ALDRIN • COLCHICINE • MIDAZOLAM • FENTANYL • ACETAMINOPHEN • CORONOPILIN • DIPYRIDAMOLE • IBUPROFEN • IONOMYCIN • CHLORDANE • DANAZOL • CHEMBL1902074 • OMEPRAZOLE • DINITRO CRESOL • QUESTIOMYCIN B • FENRETINIDE • HEPTACHLOR • PYROGALLOL • CANERTINIB • HYDROQUINONE • ENDOSULFAN • EMODIN • LANSOPRAZOLE • RETINAL • HARMINE HYDROCHLORIDE • PACLITAXEL • BEVACIZUMAB • PAMIDRONIC ACID • TALC • TRETINOIN • SUNITINIB • CETUXIMAB • CHEMBL1579130 • ALPRAZOLAM • METHIMAZOLE • RETINOL • RIBAVIRIN • TERFENADINE • DICYCLOHEXYLCARBODIIMIDE • CEFTRIAXONE • ASPIRIN • CLARITHROMYCIN • DACARBAZINE	 CCL20 HCK CCL5 RELA IL8 CCR3 CXCR2 DARC CXCL5 CXCR1 CXCL6

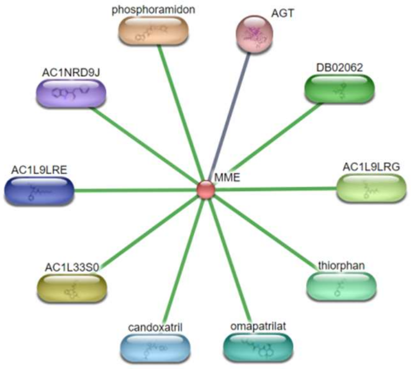
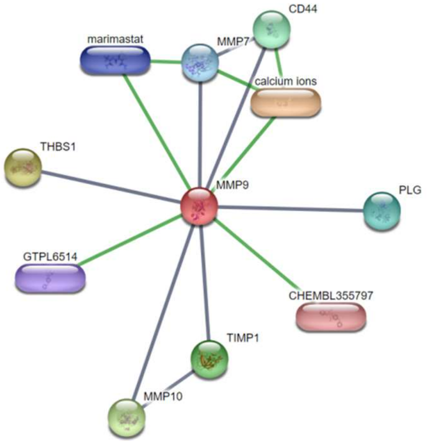
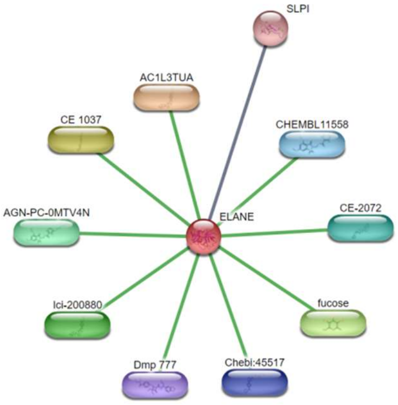
GENE SYMBOL	**DRUGS**	**NETWORK** Protein-protein interactions: Grey Chemical-protein interactions: Green Interactions between chemicals: Red
	• PENTOXIFYLLINE • CIDOFOVIR • BROXURIDINE • TROGLITAZONE • DICHLORVOS • VERAPAMIL	
MME	• CANDOXATRIL • LCZ696 • PEPINEMAB • SAMPATRILAT • SLV-334 • GALLOPAMIL • ILEPATRIL • SACUBITRIL	 phosphoramidon AGT AC1NRD9J DB02062 AC1L9LRE MME AC1L9LRG AC1L33S0 thiorphan candoxatril omapatrilat
MMP9	• MARIMASTAT • PRINOMASTAT • ANDECALIXIMAB • S-3304 • CURCUMIN PYRAZOLE • TOZULERISTIDE • CURCUMIN • INCYCLINIDE • BEVACIZUMAB • CARBOXYLATED GLUCOSAMINE • DEMETHYLWEDELOLACTONE • CELECOXIB	 CD44 marimastat MMP7 calcium ions THBS1 MMP9 PLG GTPL6514 CHEMBL355797 TIMP1 MMP10
ELANE	• SIVELESTAT • DEPELESTAT • SYMPLOSTATIN 5 • CHEMBL310871 • NICOTINE • TIPRELESTAT • ERDOSTEINE • NIFEDIPINE	 SLPI AC1L3TUA CE 1037 CHEMBL11558 AGN-PC-0MTV4N ELANE CE-2072 Ici-200880 fucose Dmp 777 Chebi:45517

TABLE4.10 GENES THAT HAVE DRU

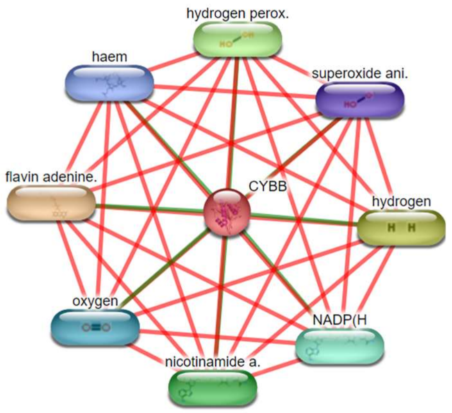
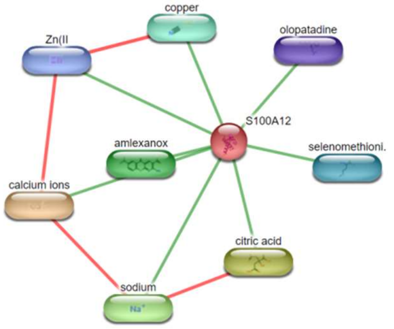
GENE SYMBOL	DRUGS	NETWORK Protein-protein interactions: Grey Chemical-protein interactions: Green Interactions between chemicals: Red
CYBB	• APIGENIN • CHRYSIN • LUTEOLIN	 hydrogen perox. haem superoxide ani. flavin adenine. CYBB hydrogen oxygen NADP(H nicotinamide a.
S100A12	• ATOGEPANT • RIMEGEPANT • METHOTREXATE • EPTINEZUMAB • UBROGEPANT	 copper Zn(II olopatadine S100A12 amlexanox selenomethioni. calcium ions citric acid sodium

TABLE4.10 GENES THAT HAVE DRUG INTERACTIONS AND INHIBITOR NETWORKS OF THE GENES THAT HAVE A DRUG RELATIONSHIP CASE OF RRMS VS CONTROLS

4.4.2 Case Secondary progressive MS (SPMS) *vs* Healthy Controls

After normalization, our dataset consists of 93 samples (SPMS N=26 and HC N=67) and 10.160 gene with their expression values.

4.4.2.1 Significance Analysis of Microarrays (SAM)

We conducted a Significance Analysis of Microarrays (SAM) on our filtered dataset so as to find differentially expressed genes based on T-statists. The cutoff for significance is determined by a tuning parameter delta, chosen by the user based on the false positive rate. One can also choose a fold change parameter, to ensure that called genes change at least a pre-specified amount.

SAM	SPMS Untreated vs Control		
	upregulated	*downregulated*	*FINAL GENES*
	37	8	45

TABLE4.11 DIFFERENTIALLY EXPRESSED GENES SPMS UNTREATED VS CONTROL

After the analysis with SAM, we concluded in 45 Differentially expressed genes considering the case SPMS Untreated-Control. We proceed with the Pigengene methodology.

4.4.2.2 Pigengene Methodology

We apply the Pigengene methodology streps on the 10.160 genes that derived from the preprocessed dataset for the case Primary Progressive MS (SPMS) *vs* Healthy Controls:

i. Weighted correlation network: Weighted Coexpression network analysis (WGCNA) was applied to group related genes into gene modules (clusters) based on their coexpression patterns in MS.

ii. Eigengenes: We computed an eigengene for each module as a weighted average of the expression of all genes in that module. (Figure 4.28 B). Module 5 is negatively associated with the Interferon treatment, whereas Module 6 is positively associated with Interferon treatment. To validate this, we modeled the probabilistic dependencies between the eigengenes using a BN (Figure 4.29). We used Bayesian networks as probabilistic predictive models to determine the state.

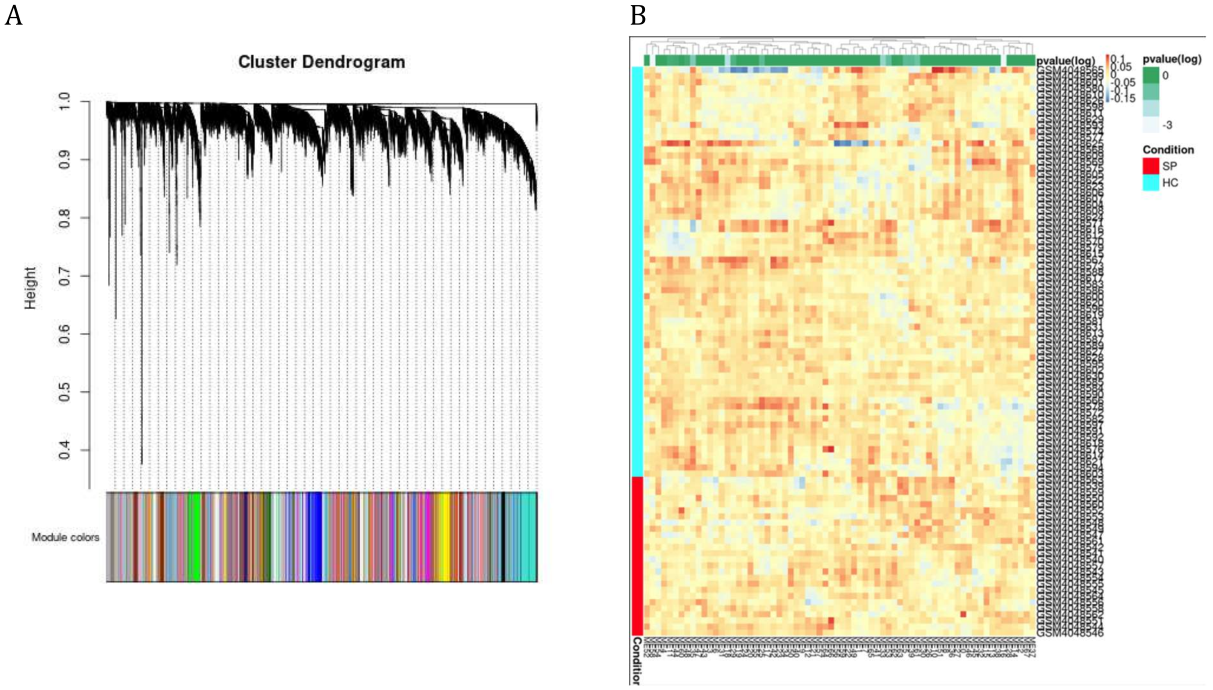


Figure 4.21 A) Modules dendrogram B) The two (2) eigengenes that are differentially expressed ME5, ME16, ME33, ME35, ME39, ME38, ME58, the intensity of the colors in each heatmap corresponds to the normalized average expression. Each column corresponds to an eigengene. Each row shows the expression of a case from the SPMS vs HC dataset.

iii. Bayesian network: We fitted a Bayesian network to the eigengenes to determine the relationships of the modules with each other and with the state of the samples. Descendants of the "Disease" node, the variable that models the state, show high dependency between these eigengenes and the state type and suggest that they have useful biological information that can explain the differences between the two states. We trained a Bayesian network to model the probabilistic dependencies between the modules. Several individual networks from random staring networks were built (no.1000) by optimizing their score. Then, we inferred a consensus network from the ones with relatively "higher" scores. The default hyper-parameters and arguments are then selected. Each module eigengene is represented by a node (observed random variable). To model the condition, we added "Disease" as an observed random variable to the network.

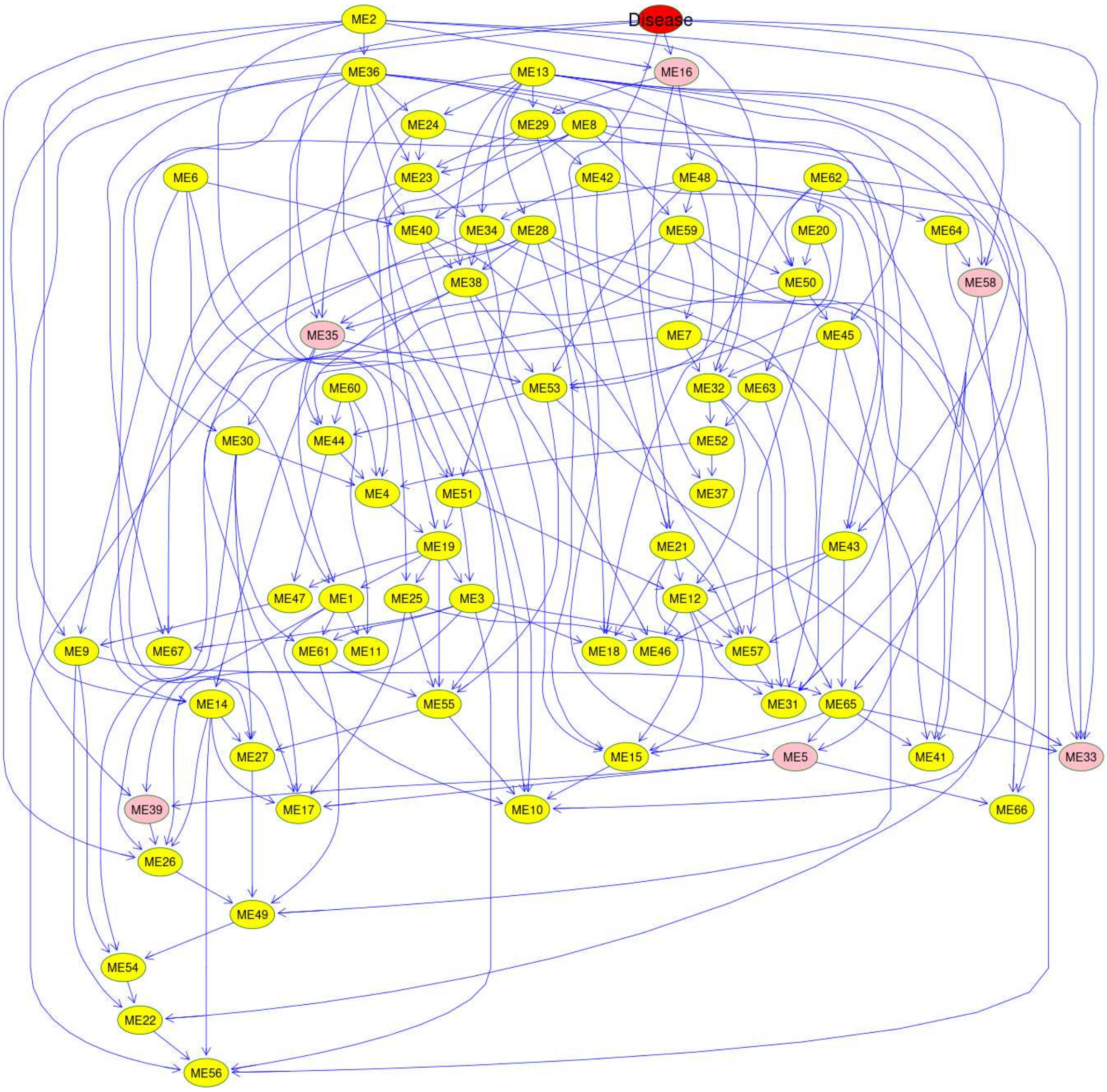


Figure 4.22 The Bayesian network fitted to the eigengenes. Each node represents an eigengene of a module. The arcs model the probabilistic dependencies between the modules. The "Disease" node is set to 1 for SPMS and 0 for HC, and its children are highlighted in pink.

iv. Decision tree: A decision tree is fitted to the two children of the Disease node in our Bayesian network (R package C50 version 0.1.0-24). We used the data to infer the topology of the tree and the corresponding parameters. The algorithm automatically selected the ME58 eigengene (modules 58). Module eigengenes are used to build a classifier that distinguishes two or more classes. Each eigengene is a weighted average of the expression of all genes in the module, where the weight of each gene corresponds to its membership in the module. Each module might contain dozens to hundreds of genes, and hence the final classifier might depend on the expression of a large number of genes. In practice, it is desirable to reduce the number of necessary genes by a decision tree. The inferred decision tree had a relatively high predictive accuracy (Figure 4.30).

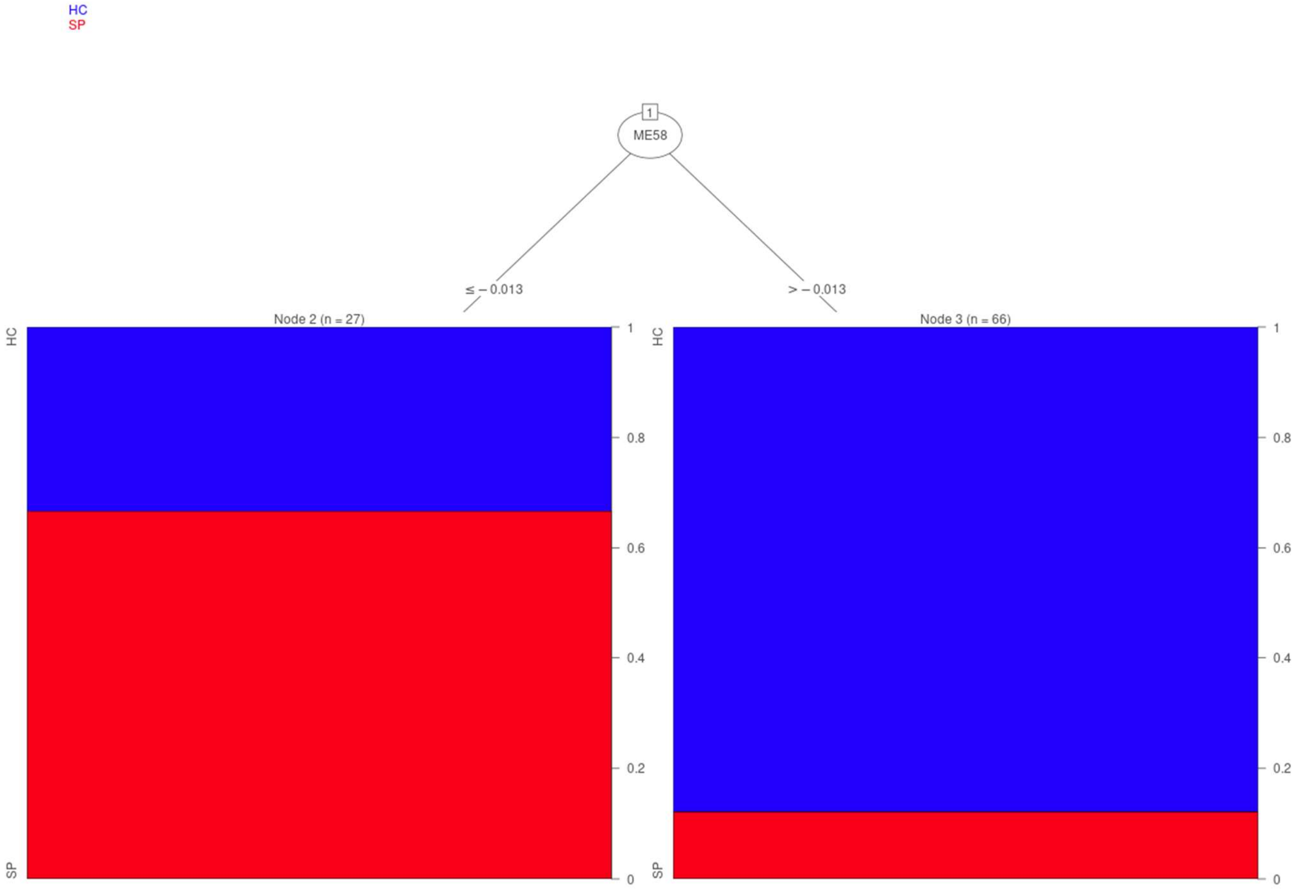


FIGURE 4.23 THE DECISION TREE FOR DISTINGUISHING SPMS FROM HC CASES. IF THE NORMALIZED EIGENGENE OF A CASE IS GREATER OR EQUAL THAN -0.013, IT IS CLASSIFIED AS SPMS. IF IT IS LESS THAN -0.013, IT IS CLASSIFIED AS HC. AT THE FIXED THRESHOLDS SHOWN ABOVE, THIS TREE CORRECTLY CLASSIFIED 760CASES (82%) IN THE DATASET. (MISCLASSIFIED 9 HC AND 8 SPMS)

4.4.2.3 Construction of PPI Network of Common DEGs for SPMS and Healthy Controls from two Approaches

We compared the resulted 79 genes from Modules 58 to the 45 differentially expressed genes from SAM. All 45 genes were common between the two methodological approaches, so we choose to include the additional 32 genes and proceed to construct the PPI network using the STRING App in the Cytoscape software. A total of 79 genes/nodes with 6 edges were enriched in the construction of the PPI network. (Figure 4.31).

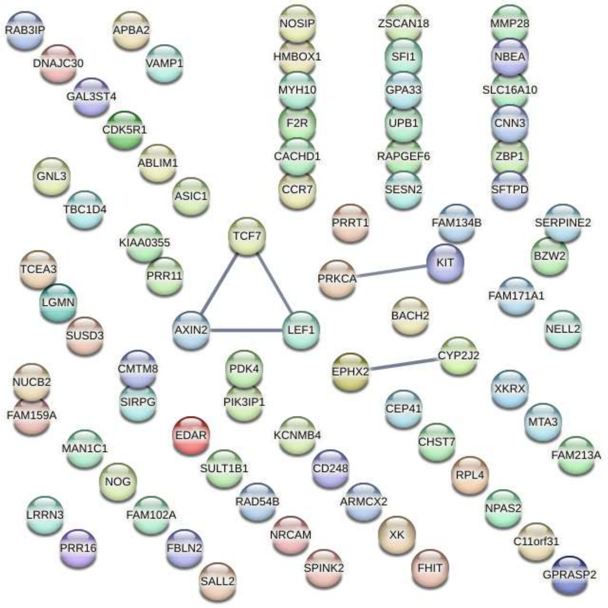


FIGURE 4.24 PPI NETWORK FROM 79 DIFFERENTIALLY EXPRESSED GENES

4.4.2.4 Critical Subnetworks and Identification of Hub Genes for SPMS and Healthy Controls Patients

Hub genes were identified by 11 topological analysis methods from the CytoHubba, a Cytoscape plugin, where the top 20 genes were selected for each method. The 24 resulted genes (Table4.16) were found in the intersection of all methods and were selected as SPMS related hub genes, providing a minimal gene set toward potential clinical testing. We also obtained one clustering module with the highest score from the PPI network of all DEGs (Figure 4.32) by MCODE algorithm. (Table4.16)

CytoHubba & MCODE: Hub genes by 11 topological analysis methods or Hub genes by CytoHubba and MCODE algorithm*
*GPRASP2|**AXIN2**|BACH2|BZW2|CEP41|CYP2J2|DNAJC30|EDAR|EPHX2|FAM102A|KIAA0355|**LEF1**|MYH10|NBEA|NPAS2|PDK4|TBC1D4|TCEA3|**TCF7**|XK|CNN3|AL3ST4|SALL2|ZBP1*

TABLE4.12 IDENTIFICATION OF HUB GENES *3 HUB GENES CYTOHUBBA & MCODE (IN BOLD);24 GENES CYTOHUBBA

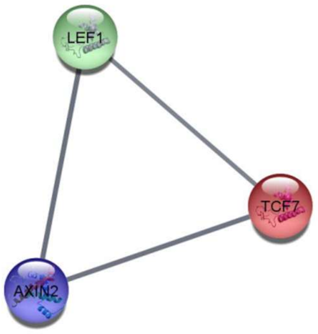


FIGURE 4.25 THE HIGHEST SCORE CLUSTERING MODULES WERE GENERATED BY MCODE, WITH 3 GENES

4.4.2.5 Statistical Evaluation

The resulting genomic signature of 24 hub genes is used to assess the classification and generalization ability of the model. The final gene signature arrived from GSE136411 dataset which was used as a training dataset (N =75 , SPMS n= 21, CTR n=54) and testing dataset (N = 18, SPMS n= 5, CTR n=13). Dataset E-MTAB-4890 was used to access the generalization ability of the resulted gene signature as an independent dataset. It consists of (N = 61, SPMS n = 21, CTR n=40). Then, the classification model was established by support vector machine (SVM).

By applying 10fold cross-validation in the model, 13 out of the 18 samples were correctly classified, with a classification accuracy of 72%, model sensitivity to CTR of 84%, specificity of 40%, and area under the ROC curve (AUC) was 0.57 (Figure 4.26 a). Furthermore, the established model was used to predict the samples in the validation data sets to test the prediction ability of the model.
In the validation group the samples were classified, with a classification accuracy of 82%, moreover, the sensitivity was 98 % and specificity of the model was 52%, and the area under the receiver operating characteristic (ROC) curve was 0.81 (Figure 4.26 b).

a

GSE136411	Real CTR	Real SPMS	
Predict CTR	11	3	
Predict MS	2	2	Totals
Totals	13	5	18
Correct	11	2	13
Sensitivity (%)	84		
Specificity (%)		40	
AUC	0.57		

b

E-MTAB-4890	Real CTR	Real SPMS	
Predict CTR	37	10	
Predict MS	3	11	Totals
Totals	40	22	63
Correct	39	11	51
Sensitivity (%)	98		
Specificity (%)		52	
AUC	0.81		

FIGURE 4.26 CONSTRUCTION OF DIAGNOSTIC MODEL AND VALIDATION OF MODEL. A) CLASSIFICATION RESULTS AND ROC CURVES OF SAMPLES BY DIAGNOSTIC MODEL IN TRAINING DATA SET. B) CLASSIFICATION RESULTS AND ROC CURVES OF SAMPLES BY DIAGNOSTIC MODEL IN E-MTAB-4890.

Our diagnostic model does not perform well as we can see from the Results in Figure. After inspecting our data, we noticed that the density plots showed the feature's distribution for all features over the two classes, and there is really not much discriminative power between conditions. The extracted features are overlapping between the two classes, and we might have a "garbage in, garbage out" issue, more than a "this is not enough data" issue. The imbalance between the majority class Controls and Secondary Progressive shows that building the classifier using the data as it is, would in most cases give us a prediction model that always returns the majority class. The classifier would be biased.

We performed oversampling. It makes no sense to create instances based on our current minority class and then exclude an instance for validation, pretending we didn’t generate it using data that is still in the training set. We balance the dataset by oversampling the minority class. First, we start cross-validating. This means that at each iteration we first exclude the sample to use as test set, and then oversample the remaining of the minority class. We are not using the same data for training and testing. Therefore, we will obtain more representative results. The same holds even if we use other cross-validation methods, such as leave one out cross-validation. By applying 10fold cross-validation in the model and up-sampling the resulted model is shown in Figure 4.

a

GSE136411	Real CTR	Real SPMS	
Predict CTR	10	1	
Predict MS	3	4	Totals
Totals	13	5	18
Correct	10	4	14
Sensitivity (%)	76		
Specificity (%)		80	
AUC	0.75		

b

E-MTAB-4890	Real CTR	Real SPMS	
Predict CTR	31	8	
Predict MS	9	13	Totals
Totals	40	21	61
Correct	31	13	44
Sensitivity (%)	76		
Specificity (%)		62	
AUC	78		

Figure 4.27 Construction of diagnostic model and validation of model. a) Classification results and ROC curves of samples by diagnostic model in training data set. b) Classification results and ROC curves of samples by diagnostic model in E-MTAB-4890

There were 14 out of the 18 samples correctly classified, with a classification accuracy of 70%, model sensitivity to CTR of 76%, specificity of 80%, and area under the ROC curve (AUC) was 0.75. (Figure 4.27 a). In the validation group the samples were classified, with a classification accuracy of 72%, moreover, the sensitivity was 76 % and specificity of the model was 62%, and the area under the receiver operating characteristic (ROC) curve was 0.78 (Figure 4.27 b).

4.4.2.6 Biological interpretation

Each selected Illumina probe set was mapped to an annotation of Entrez Gene ID and Gene Symbol using the online tool WebGestalt (2013) (http://www.webgestalt.org/2013/). Considering the case SPMS *versus* controls, an over-representation analysis of the resulted 45-DEG-gene signature was performed in WebGestalt (2019). The enriched biological process categories are presented in Table 4.13A, where the enriched pathway categories are illustrated in Figure 4.28A.

Gene Set	Description	P Value	FDR
GO:0060326	cell chemotaxis	5E-05	0.0449
GO:0017145	stem cell division	0.0001	0.0478

Table 13.a GO-biological processes analysis of 45 DEGs in the case of SPMS *vs* controls

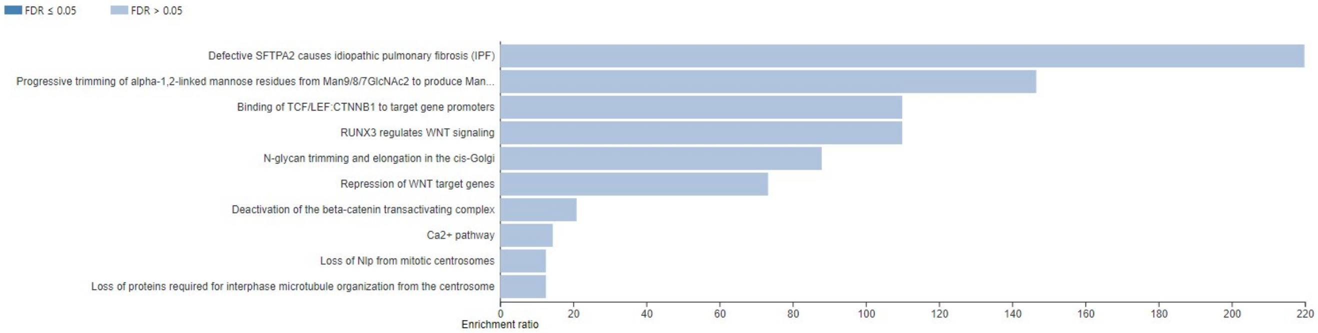


Figure 4.28a. Reactome pathway analysis of 45 DEGs in the case of SPMS vs controls

Furthermore, we also performed an over-representation analysis to explore the functional properties (GO biological processes, pathways) of the 3-hub-gene signature, which may be more specific to SPMS stage. The analysis of the gene signature was performed using the online tool WebGestalt (2019).

Gene Set	Description	P Value	FDR
GO:1904837	beta-catenin-TCF complex assembly	7.50E-09	6.4E-06
GO:0070670	response to interleukin-4	1.6E-05	0.00691
GO:0198738	cell-cell signaling by wnt	3E-05	0.00844
GO:0002200	somatic diversification of immune receptors	5.9E-05	0.01244
GO:0061053	somite development	0.00011	0.01939
GO:0042476	odontogenesis	0.00021	0.03035

TABLE 4.13B GO-BIOLOGICAL PROCESSES ANALYSIS OF THE 3-HUB-GENE SIGNATURE IN THE CASE OF SPMS *VS* CONTROLS

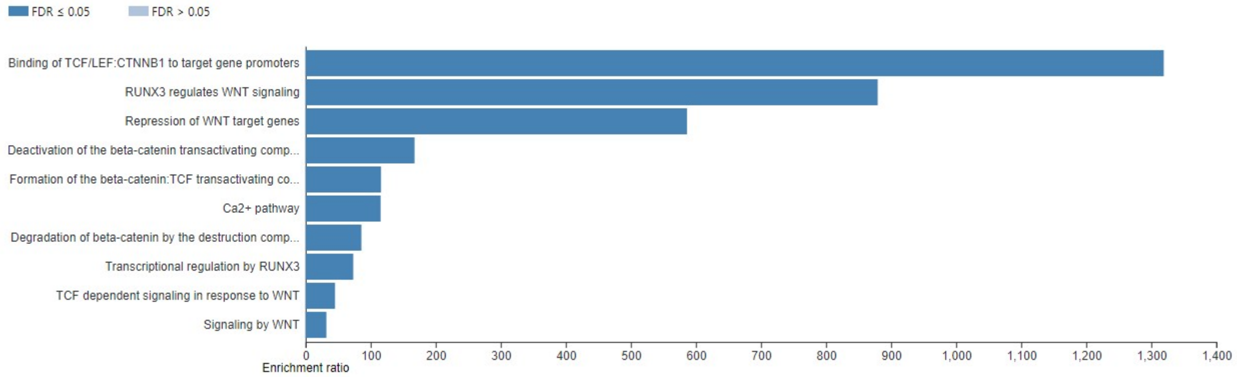


FIGURE 4.28B REACTOME PATHWAY ANALYSIS OF THE 3-HUB-GENE SIGNATURE IN THE CASE OF SPMS *VS* CONTROLS.

As depicted in above tables (Table 4.13A, 4.13B) and figures (Figures 4.28A, 4.28B), five out of ten enriched pathways provided by the 3-hub-gene signature are also enriched in the 45-DEG-gene signature, but without statistical significance. Regarding the enriched GO biological processes provided by both gene signatures, no overlap exists between these enriched processes. Figure 4.28B illustrates the dominance of the WNT signaling pathways, but also the non-canonical WNT Ca2+ signaling, which are implicated in inflammatory response [103,104]. Lengfeld et al [105] report that Wnt signaling pharmacologic enhancement may be helpful to restrain blood–brain barrier (BBB) damage and central nervous system (CNS) immune cell infiltration in multiple sclerosis. Disruption of the BBB, that sometimes results from a dysregulated Wnt/β-catenin signaling pathway under various pathophysiological conditions, is also a determing and early feature of MS that directly damages the CNS, promotes immune cell infiltration, and influences clinical outcomes [106] [paper in preparation]

4.4.2.7 Candidate drugs targeting hub genes

Using the DGIdb database we explore drug-gene interactions of the 10 hub genes that derived from the CYTOHUBBA analysis. The 3 genes cluster given from MCODE did not produce any results, so we chose to examine the hub genes resulted from CYTOHUBBA. The drugs for possibly addressing patients in the Secondary Progressive stage of MS are shown in Table4.14. We used the STITCH database, in order to construct downstream networks of the genes that have a drug relationship, to investigate the additional effects caused by inhibitors of these genes. All networks are also included in Table4.14 The network setting were “Experiments”,” Databases”,” Coexpression” and confidence was set to high=0,9.

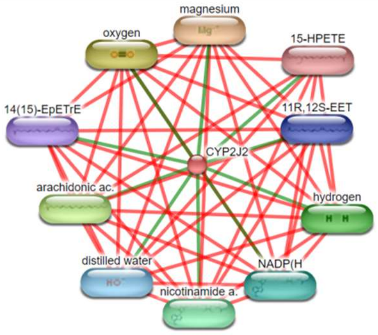
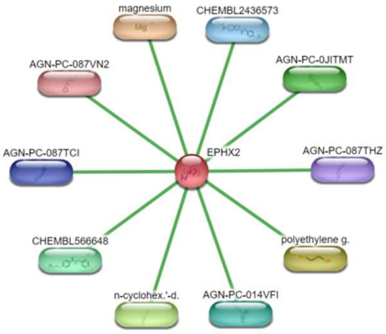
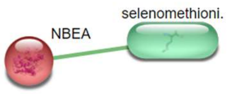
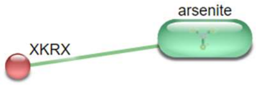
GENE SYMBOL	DRUGS	NETWORK Protein-protein interactions: Grey, Chemical-protein interactions: Green Interactions between chemicals: red.
CYP2J2	• TERFENADINE • THIORIDAZINE • TACROLIMUS • DICLOFENAC • AMIODARONE • NABUMETONE • ASTEMIZOLE • ALBENDAZOLE • MESORIDAZINE • DANAZOL	 magnesium oxygen 15-HPETE 14(15)-EpETrE 11R,12S-EET CYP2J2 arachidonic ac. hydrogen distilled water NADP(H nicotinamide a.
EPHX2	• FULVESTRANT • 6BIO • AR9281 • LITHIUM • ALOE-EMODIN • ALOIN	 magnesium CHEMBL2436573 AGN-PC-087VN2 AGN-PC-0JITMT AGN-PC-087TCI EPHX2 AGN-PC-087THZ CHEMBL566648 polyethylene g. n-cyclohex.'-d. AGN-PC-014VFI
NBEA	• METFORMIN	
XKRX	• ENSITUXIMAB	

TABLE4.14 GENES THAT HAVE DRUG INTERACTIONS AND INHIBITOR NETWORKS OF THE GENES THAT HAVE A DRUG RELATIONSHIP CASE OF SPMS VS CONTROLS.

4.4.3 Case Primary Progressive MS (PPMS) *vs* Healthy Controls

After normalization, our dataset consists of 102 samples (PPMS N=35 and HC N=67) and 10.160 gene with their expression values.

4.4.3.1 Significance Analysis of Microarrays (SAM)

We conducted a Significance Analysis of Microarrays (SAM) on our filtered dataset so as to find differentially expressed genes based on T-statists. The cutoff for significance is determined by a tuning parameter delta, chosen by the user based on the false positive rate. One can also choose a fold change parameter, to ensure that called genes change at least a pre-specified amount.

SAM	PPMS Untreated vs Control		
	upregulated	*downregulated*	*FINAL GENES*
	13	38	51

TABLE4.15 DIFFERENTIALLY EXPRESSED GENES PPMS UNTREATED *VS* CONTROL

After the analysis with SAM, we concluded in 51 Differentially expressed genes considering the case PPMS Untreated- Control. We proceed with the Pigengene methodology.

4.4.3.2 Pigengene Methodology

We apply the Pigengene methodology streps on the 10.160 genes that derived from the preprocessed dataset for the case Primary Progressive MS (PPMS) vs Healthy Controls:

i. Weighted correlation network: Weighted Coexpression network analysis (WGCNA) was applied to group related genes into gene modules (clusters) based on their coexpression patterns in MS.

ii. Eigengenes: We computed an eigengene for each module as a weighted average of the expression of all genes in that module. (Figure 4.29 B). Module 64 is associated with the disease. To validate this, we modeled the probabilistic dependencies between the eigengenes using a BN (Figure 4.30). We used Bayesian networks as probabilistic predictive models to determine the state.

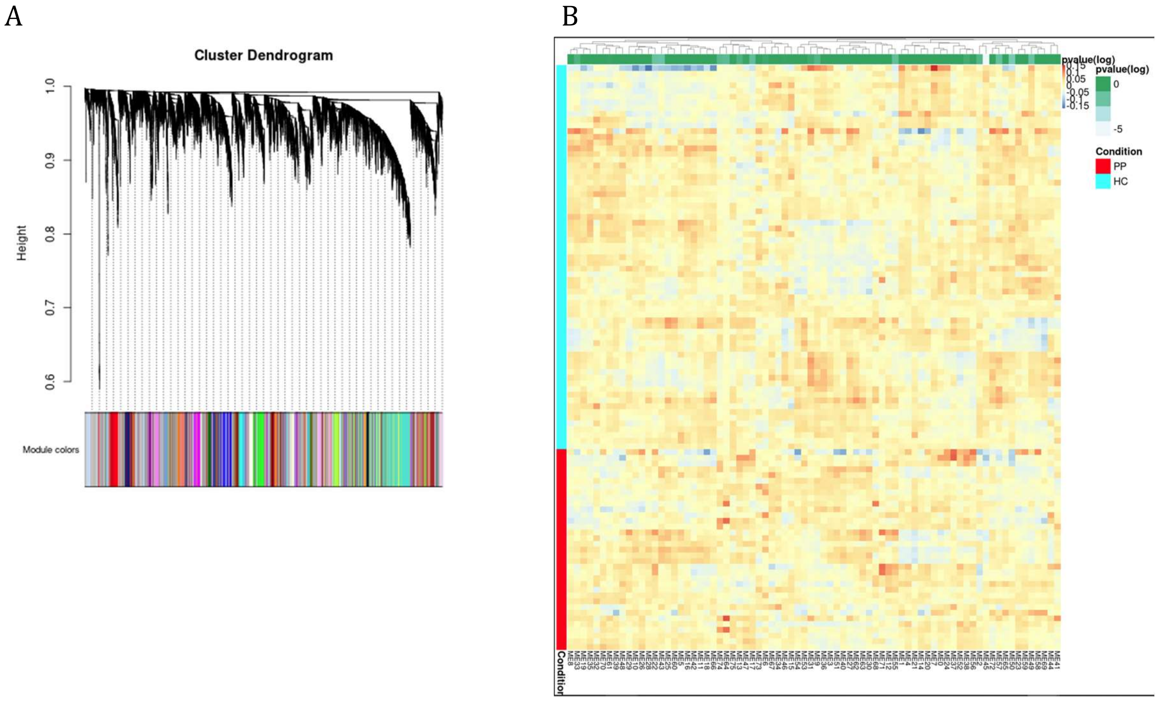


FIGURE 4.29 A) MODULES DENDROGRAM B) THE EIGENGENES THAT IS DIFFERENTIALLY EXPRESSED ME2, ME41, ME43, ME45, ME64, ME73, M74 THE INTENSITY OF THE COLORS IN EACH HEATMAP CORRESPONDS TO THE NORMALIZED AVERAGE EXPRESSION. EACH COLUMN CORRESPONDS TO AN EIGENGENE. EACH ROW SHOWS THE EXPRESSION OF A CASE FROM THE PPMS VS HC DATASET.

iii. Bayesian network: We fitted a Bayesian network to the eigengenes to determine the relationships of the modules with each other and with the state of the samples. Descendants of the “Disease” node, the variable that models the state, show high dependency between these eigengenes and the state type and suggest that they have useful biological information that can explain the differences between the two states. We trained a Bayesian network to model the probabilistic dependencies between the modules. Several individual networks from random staring networks were built (no.1000) by optimizing their score. Then, we inferred a consensus network from the ones with relatively “higher” scores. The default hyper-parameters and arguments are then selected. Each module eigengene is represented by a node (observed random variable). To model the condition, we added “Disease” as an observed random variable to the network.

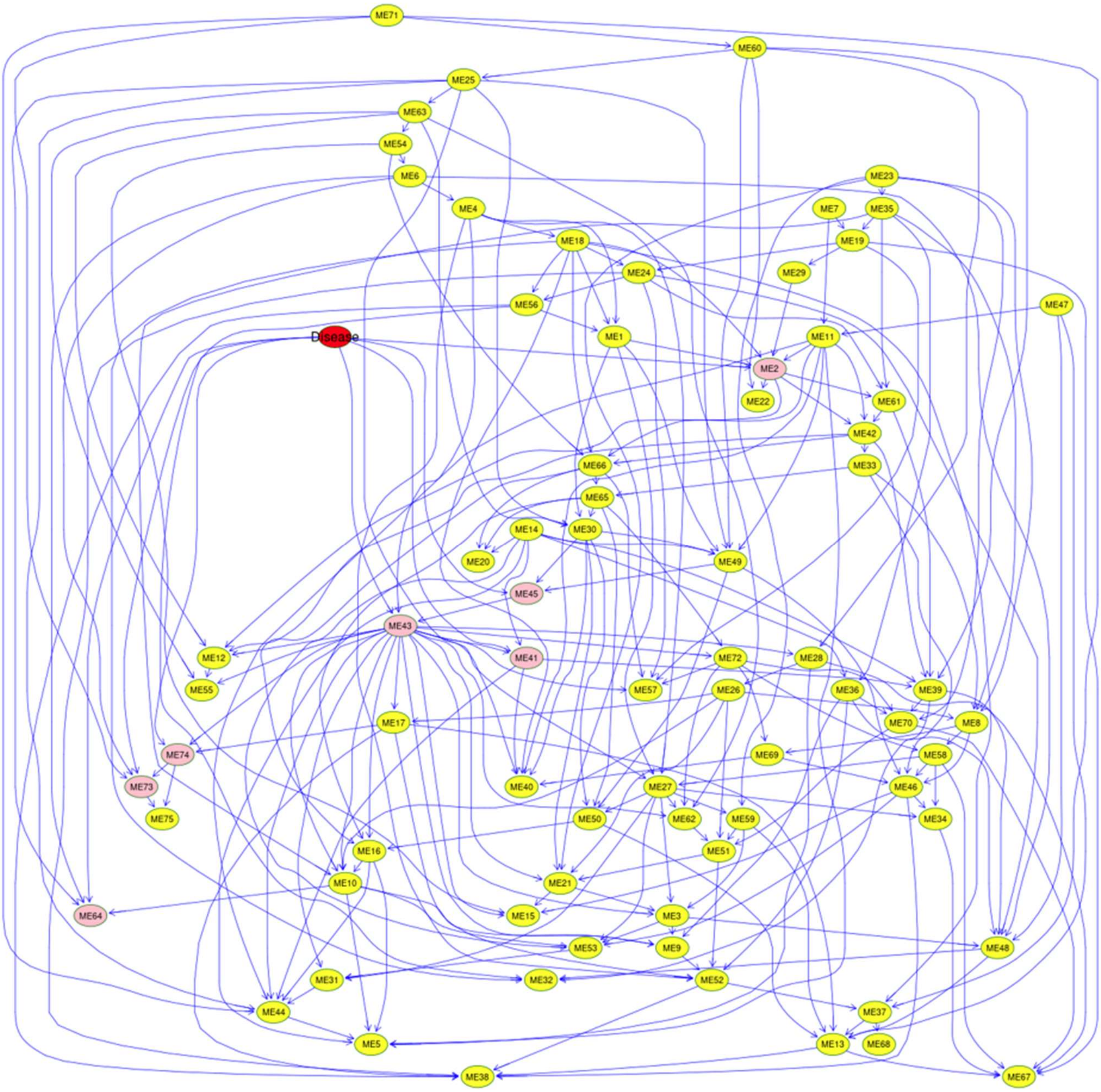


FIGURE 4.30 THE BAYESIAN NETWORK FITTED TO THE EIGENGENES. EACH NODE REPRESENTS AN EIGENGENE OF A MODULE. THE ARCS MODEL THE PROBABILISTIC DEPENDENCIES BETWEEN THE MODULES. THE "DISEASE" NODE IS SET TO 1 FOR PPMS AND 0 FOR HC, AND ITS CHILDREN ARE HIGHLIGHTED IN PINK.

iv. Decision tree: A decision tree is fitted to the two children of the Disease node in our Bayesian network (R package C50 version 0.1.0-24). We used the data to infer the topology of the tree and the corresponding parameters. The algorithm automatically selected the ME64 eigengene (modules 64). Module eigengenes are used to build a classifier that distinguishes two or more classes. Each eigengene is a weighted average of the expression of all genes in the module, where the weight of each gene corresponds to its membership in the module. Each module might contain dozens to hundreds of genes, and hence the final classifier might depend on the expression of a large number of genes. In practice, it is desirable to reduce the number of necessary genes by a decision tree. The inferred decision tree had a relatively high predictive accuracy (Figure 4.31).

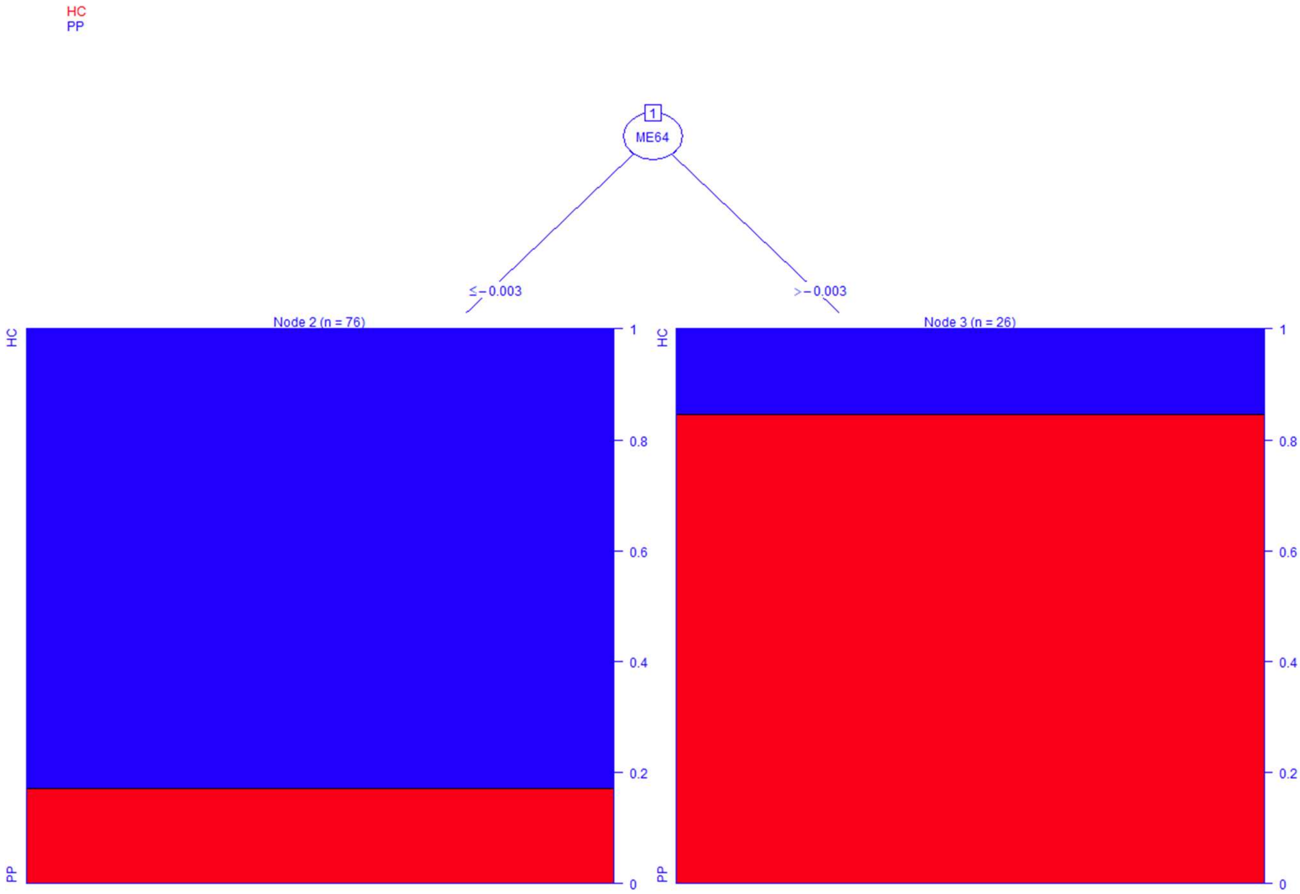


Figure 4.31 The decision tree for distinguishing PPMS from HC cases. If the normalized eigengene of a case is greater or equal than -0.003, it is classified as HC. If it is less than -0.003, it is classified as PPMS. At the fixed thresholds shown above, this tree correctly classified 86 cases (84%) in the dataset. (Misclassified 3 HC and 13 PPMS)

4.4.3.3 Construction of PPI Network of Common DEGs for PPMS and Healthy Controls from two Approaches

We compared the resulted 73 genes from Modules 64 to the 51 differentially expressed genes from SAM. All 51 genes were common between the two methodological approaches, so we choose to keep the additional 22 genes and proceed to construct the PPI network with 73 significant genes using the STRING App in the Cytoscape software. A total of 72 genes/nodes with 31 edges were enriched in the construction of the PPI network. (Figure 4.32)

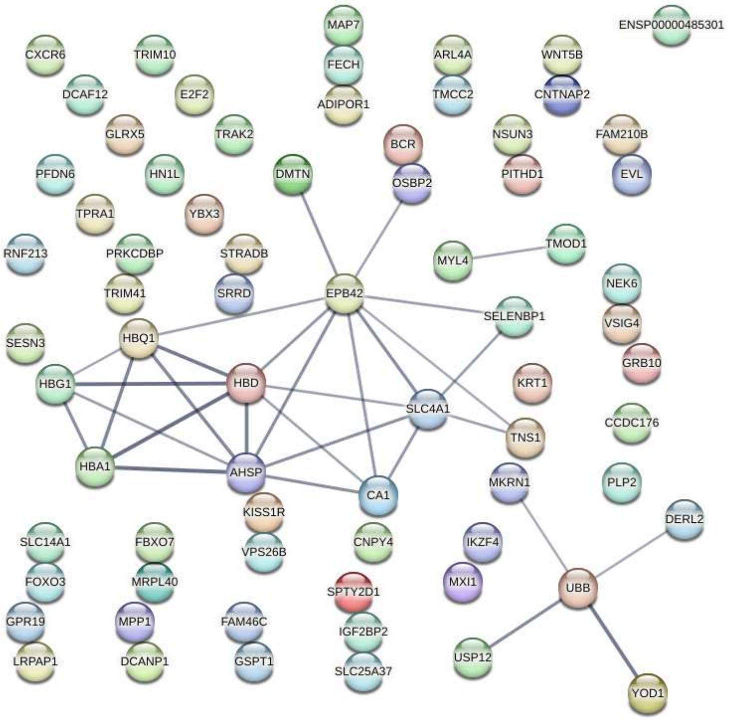


Figure 4.32 PPI network from 73 differentially expressed genes

4.4.3.4 Critical Subnetworks and Identification of Hub Genes for PPMS and Healthy Controls Patients

Hub genes were identified by 11 topological analysis methods from the CytoHubba, a Cytoscape plugin, where the top 20 genes were selected for each method. The 32 resulted genes (Table4.12) were found in the intersection of all methods and were selected as PPMS related hub genes. We also obtained one clustering module with the highest score from the PPI network of all DEGs (Figure 4.25) by MCODE algorithm. It was found that 8 genes from 10 were included in 32 hub genes were contained in this module (Table4.12).

CytoHubba & MCODE: Hub genes by 11 topological analysis methods or Hub genes by CytoHubba and MCODE algorithm*
AHSP|*DERL2*|***EPB42***|*FECH*|***HBD***|*MRPL27*|*MRPL4*|*MYL4*|***HBA1***|***HBG1***|***HBQ1***|*MRPL15*|*MRPL32*|*MRPL40*|*PFDN1*|*PFDN6*|*SLC25A37*|***SLC4A1***|*UBB*|*YOD1*|*ADIPOR1*|*BCR*|***CA1***|*CNTNAP2*|*DCANP1*\FBXO7|*IGF2BP2*|*IKZF4*|*KISS1R*|*MAP7*|*TRIM10*|*YBX3*

Table4.16 Identification Of Hub Genes *8 hub genes Cytohubba & MCODE (in bold);32 genes Cytohubba

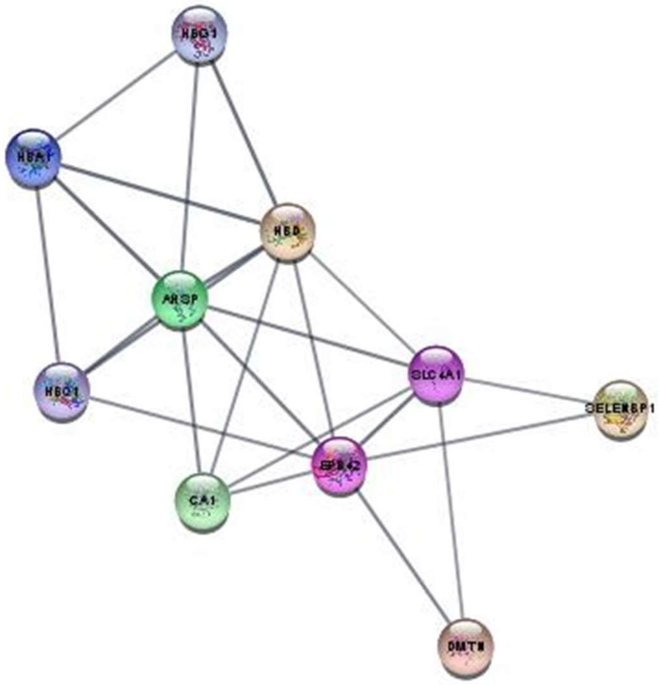


FIGURE 4.33 THE HIGHEST SCORE CLUSTERING MODULES WERE GENERATED BY MCODE, WITH 10 GENES

4.4.3.5 Statistical Evaluation

The resulting genomic signature of 32 hub genes is used to assess the classification and generalization ability of the model. The final gene signature arrived from GSE136411 dataset which was used as a training dataset (N = 72, PPMS n= 25, CTR n=47) and testing dataset (N = 30, PPMS n= 10, CTR n=20). Dataset E-MTAB-4890 was used to access the generalization ability of the resulted gene signature as an independent dataset. It consists of (N = 63, PPMS n = 23, CTR n=40). Then, the classification model was established by support vector machine (SVM).

By applying 10fold cross-validation in the model, 25 out of the 30 samples were correctly classified, with a classification accuracy of 83%, model sensitivity to CTR of 100%, specificity of 50%, and area under the ROC curve (AUC) was 0.9 (Figure 4.34 a). Furthermore, the established model was used to predict the samples in the validation data sets to test the prediction ability of the model.

In the validation group the samples were classified, with a classification accuracy of 79%, moreover, the sensitivity was 100% and specificity of the model was 43%, and the area under the receiver operating characteristic (ROC) curve was 0.9 (Figure 4.34 b).

a

GSE136411	Real CTR	Real PPMS	
Predict CTR	20	5	
Predict MS	0	5	Totals
Totals	20	10	30
Correct	20	5	25
Sensitivity (%)	100		
Specificity (%)		50	
AUC	0.9		

b

E-MTAB-4890	Real CTR	Real PPMS	
Predict CTR	40	13	
Predict MS	0	10	Totals
Totals	40	23	63
Correct	40	10	50
Sensitivity (%)	100		
Specificity (%)		43	
AUC	0.9		

Figure 4.34 Construction of diagnostic model and validation of model. a) Classification results and ROC curves of samples by diagnostic model in training data set. b) Classification results and ROC curves of samples by diagnostic model in E-MTAB-4890

As we can see our diagnostic model performs purely talking into account the specificity percentage. After inspecting our data, we noticed that the density plots showed the feature's distribution for all features over the two classes, and there is really not much discriminative power between conditions. The extracted features are overlapping between the two classes, and we might have a "garbage in, garbage out" issue, more than a "this is not enough data" issue. The control cases are twice the size of Primary Progressive so we can say that building the classifier using the data as it is, would in most cases give us a prediction model that always returns the majority class. The classifier would be biased.

We performed Under-sampling that balances the dataset by reducing the size of the abundant class. This method is used when quantity of data is sufficient. By keeping all samples in the rare class and randomly selecting an equal number of samples in the abundant class, a balanced new dataset can be retrieved for further modelling. By applying 10fold cross-validation in the model and under sampling the resulted model is shown in Figure 4.35

a

GSE136411	Real CTR	Real PPMS	
Predict CTR	14	0	
Predict MS	6	10	Totals
Totals	20	10	30
Correct	14	10	24
Sensitivity (%)	70		
Specificity (%)		100	
AUC	0.845		

b

E-MTAB-4890	Real CTR	Real PPMS	
Predict CTR	27	4	
Predict MS	13	19	Totals
Totals	40	23	63
Correct	27	19	46
Sensitivity (%)	67		
Specificity (%)		82	
AUC	0.90		

Figure 4.35 Construction of diagnostic model and validation of model. a) Classification results and ROC curves of samples by diagnostic model in training data set. b) Classification results and ROC curves of samples by diagnostic model in E-MTAB-4890

There were 24 out of the 30 samples correctly classified, with a classification accuracy of 80%, model sensitivity to CTR of 70%, specificity of 100%, and area under the ROC curve (AUC) was 0.90.(Figure 4.35 a). In the validation group the samples were classified, with a classification accuracy of 73%, moreover, the sensitivity was 67% and specificity of the model was 82%, and the area under the receiver operating characteristic (ROC) curve was 0.90 (Figure 4.35 b).

4.4.3.6 Biological interpretation

Each selected Illumina probe set was mapped to an annotation of Entrez Gene ID and Gene Symbol using the online tool WebGestalt (2013) (http://www.webgestalt.org/2013/). Considering the case PPMS *versus* controls, an over-representation analysis of the resulted 51-DEG-gene signature was performed in WebGestalt (2019). The enriched biological process categories are presented in Table 4.17A, where the enriched pathway categories are illustrated in Figure 4.36A.

Gene Set	Description	P Value	FDR
GO:0051291	protein heterooligomerization	0.00032213	0.27381
GO:0051187	cofactor catabolic process	0.00085814	0.36471
GO:1903513	endoplasmic reticulum to cytosol transport	0.0031050	0.87976
GO:0042737	drug catabolic process	0.0076230	1
GO:0032527	protein exit from endoplasmic reticulum	0.0085681	1
GO:0016999	antibiotic metabolic process	0.0090822	1
GO:0051705	multi-organism behavior	0.017844	1
GO:0015893	drug transport	0.026145	1
GO:0048872	homeostasis of number of cells	0.036027	1
GO:0072593	reactive oxygen species metabolic process	0.040215	1
Note: Common GO-biological processes between DEG-signatures and hub-gene-signatures are highlighted in blue.			

TABLE 4.17A GO-BIOLOGICAL PROCESSES ANALYSIS OF 51 DEGS IN THE CASE OF PPMS *VS* CONTROLS

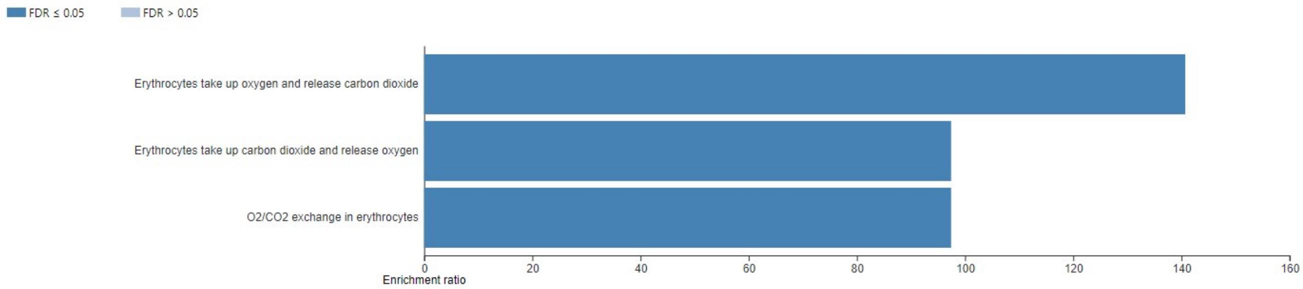


FIGURE 4.36A. REACTOME PATHWAY ANALYSIS OF 51 DEGS IN THE CASE OF PPMS *VS* CONTROLS.

Moreover, we also performed an over-representation analysis to further explore the functional information (pathways, GO biological processes) of the 10-hub-gene signature, which may be more indicative to PPMS stage. The analysis of the gene signature was performed using the online tool WebGestalt (2019).

Gene Set	Description	P Value	FDR
GO:0051187	cofactor catabolic process	4E-06	0.00344
GO:0051291	protein heterooligomerization	2.2E-05	0.00949
GO:0042737	drug catabolic process	4.1E-05	0.01056
GO:0016999	antibiotic metabolic process	5E-05	0.01056
GO:0015893	drug transport	0.00016	0.0276
GO:0048872	homeostasis of number of cells	0.00024	0.03256

GO:0072593	reactive oxygen species metabolic process	0.00027	0.03256
Note: Common GO-biological processes between DEG-signatures and hub-gene-signatures are highlighted in blue.			

TABLE 4.17B GO-BIOLOGICAL PROCESSES ANALYSIS OF THE 10-HUB-GENE SIGNATURE IN THE CASE OF PPMS VS CONTROLS

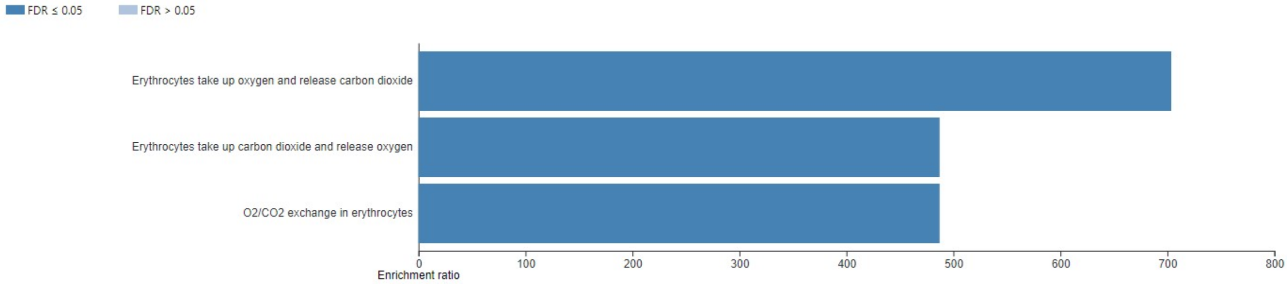


FIGURE 4.36B. REACTOME PATHWAY ANALYSIS OF THE 10-HUB-GENE SIGNATURE IN THE CASE OF PPMS VS CONTROLS.

As shown in above tables (Table 4.17A and 4.173B) and figures (Figures 4.36A and 4.36B), all the enriched processes and pathways provided by the 10-hub-gene signature are also enriched in the 51-DEG-gene signature and related to metabolic processes and O2/CO2 exchange in erythrocytes. Interestingly, these GO biological processes become statistical significance (<0.05) or the pathways have greater statistical significance (10^{-6}) in the 10-hub-gene signature. More recently, Geiger et al [102] point out to the potential role of erythrocyte (red blood cells) in the mechanisms and treatment of MS, given that release key molecules (adenosine triphosphate (ATP), nitric oxide (NO)), which are determinants in immune response, and reports suggest that release levels of these signaling molecules are often abnormal in autoimmune disease. [paper in preparation]

4.4.3.7 Candidate drugs targeting hub genes

Using the DGIdb database we explore drug-gene interactions of the 10 hub genes that derived from the MCODE analysis. 8 genes out of 10, were found common between CYTOHUBBA and MCODE. The drugs for possibly addressing patients in the Primary Progressive stage of MS are shown in Table4.18. We used the STITCH database, in order to construct downstream networks of the genes that have a drug relationship, to investigate the additional effects caused by inhibitors of these genes. All networks are also included in Table4.14. The network setting were "Experiments","Databases","Coexpression" and confidence was set to high=0,9.

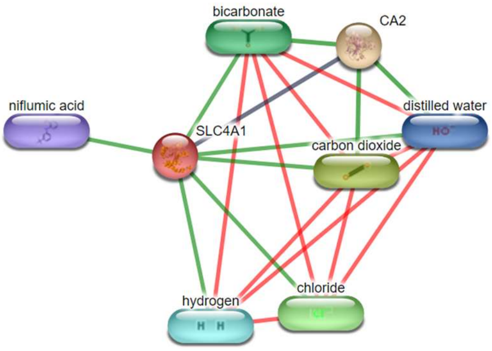
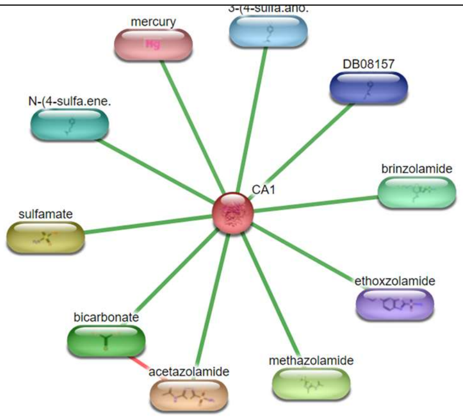
GENE SYMBOL	**DRUGS**	**NETWORK** Protein-protein interactions: Grey, Chemical-protein interactions: Green Interactions between chemicals: red.
SLC4A1	• METOPROLOL • ATENOLOL	 bicarbonate CA2 niflumic acid SLC4A1 distilled water carbon dioxide hydrogen chloride
CA1	• ACETAZOLAMIDE SODIUM • POLMACOXIB • ZONISAMIDE • METHAZOLAMIDE • ETHOXZOLAMIDE • ACETAZOLAMIDE • DICHLORPHENAMIDE • TRICHLORMETHIAZIDE • METHOCARBAMOL • CHLOROTHIAZIDE • RESORCINOL • MEDRONIC ACID • PHENOL • CURCUMIN • CATECHOL • SULFAMIDE • PARABEN • LEVETIRACETAM	 mercury 3-(4-sulfa.ano. DB08157 N-(4-sulfa.ene. brinzolamide CA1 sulfamate ethoxzolamide bicarbonate acetazolamide methazolamide

TABLE4.18 GENES THAT HAVE DRUG INTERACTIONS AND INHIBITOR NETWORKS OF THE GENES THAT HAVE A DRUG RELATIONSHIP CASE OF PPMS VS CONTROLS.

5 CONCLUSIONS

The aim of this thesis was to identify biomarkers that aid in early identification of Multiple Sclerosis disease as well as of IFNβ responders. A second aim of our study was to identify biomarkers that aid in early identification of MS stages, i.e. the relapsing-remitting form (RRMS), the secondary progressive phase (SPMS) and the primary progressive MS (PPMS). The methodological approach that we chose to implement was a combination of statistical and biological analysis. The steps that we followed was firstly accessing the Differential expression of our datasets through two different statistical methods, Significant Analysis of Microarrays (SAM) a non-parametric approach as well as "linear models for microarray data" (Limma). Then, we compared and combined our results with the PIGENGENE Methodology. Pigengene methodology enabled us to create gene coexpression networks through the identification of significant gene co expressed modules and examine the cases under study by gathering all the biological information of each module into eigengenes. In addition, Bayesian networks inference was implemented based on the eigengenes of each module, in order to elucidate the significant genes that can classify our samples under study. From the resulted gene signature, a Protein-Protein Interaction Network was created, demonstrating the relationships between genes and different topological clustering algorithms were performed (CytoHubba, MCODE) in order to conclude in a minimum set of pathways and hub-genes, that play an important role in the identification of IFNβ responders and give a chance to predict or prognose Multiple sclerosis patients outcome. Moreover, the generalization ability of the observed results was examined. The ability of how the results of a statistical analysis will generalize to an independent data set was evaluated as well as their biological significance. Finally, a good generalization performance is achieved when a gene signature is able to predict the label of unseen samples correctly. That said, every case that we examined, a new independent dataset is used and the procedure of 10 – fold cross validation is repeated. The resulted gene signature in every case, was examined for its generalization performance when it comes to the classification of unknown samples through the classification method SVM. Our approach resulted in highly connected hub genes generating four highly reliable hub-gene-signatures with high classification performance: a) 21-hub-gene signature that could predict the response of interferon beta (IFNβ) therapy in patients with MS (Accuracy = 91,49%, Sensitivity = 94.55%, Specificity = 87.15%), b) a 44-hub-gene signature that is linked to RRMS (Accuracy =83%, Sensitivity 60%, Specificity=94%,), c) a 32-hub-gene signature that is related to PPMS stage (Accuracy = 80% , Sensitivity =70% , Specificity = 100%) and d) a 24-hub-gene signature that is connected with SPMS stage (Accuracy =72% , Sensitivity =76% , Specificity =62%), demonstrating potential clinical benefit. Finally, we approached the topic of drug repurposing by examining the drug-gene relationships through different databases.

We used functional analysis to test for enrichment of both DGE-signatures and hub-gene-signatures in INF treated *versus* untreated MS, untreated MS *versus* controls, RRMS *versus* controls, PPMS *versus* controls, and SPMS *versus* controls. Our biological findings indicate that our methodological approach identifies structured (non-random) selections of genes involved in MS disease pathogenesis. Furthermore, the analysis of all examined cases provides specific aspects of immune system processes and related pathways, and also significant determinants of immune response, which highlight their importance in the design of laboratory experiments for the elucidation of disease mechanisms and also for drug discovery in MS. Moreover, our methodological approach creates highly interconnected hub-genes that are more informative than the DEGs and can be more easily validated as novel therapeutic targets or diagnostic/prognostic biomarkers in MS. Finally, our results point out that the proposed combined framework is effective in discovering of potentially causal pathways, gene networks and hub-genes. Finally, we investigated the drug repurposing by examining the drug-hub gene relationships through different databases.

We can safely say that we managed to examine relationships of transcriptomic signatures and deduce submodules of greater significance in relation to Multiple Sclerosis, the progression of the disease and future therapy. In addition, we determine how gene molecules influence each other, to improve the means of predicting "DISEASE triggering" relations/pathways and we introduced a methodology generic enough to be applied to several complex genetic diseases.

References

[1] C. Walton, R. King, L. Rechtman, W. Kaye, E. Leray and R. A. Marrie, “Rising prevalence of multiple sclerosis worldwide: Insights from the Atlas of MS, third edition,” vol. 26, no. 14, 2020.

[2] E. Leray, T. Moreau, A. Fromont and G. Edan, “Epidemiology of multiple sclerosis,” vol. 172, no. 1, 2016.

[3] M. Filipi and S. Jack, “Interferons in the treatment of multiple sclerosis: A clinical efficacy, safety, and tolerability update,” vol. 22, no. 4, 2020.

[4] J. F. Kurtzke, W. F. Page, F. M. Murphy and J. E. Norman, “Epidemiology of multiple sclerosis in US veterans: 4. Age at onset,” vol. 11, no. 4-6, 1992.

[5] Sadovnick AD and Baird PA, “Sex Ratio in Offspring of Patients with Multiple Sclerosis,” vol. 306, no. 18, 1982.

[6] M. T. Wallin, W. F. Page and J. F. Kurtzke, “Multiple Sclerosis in US Veterans of the Vietnam Era and Later Military Service: Race, Sex, and Geography,” vol. 55, no. 1, 2004.

[7] S. M. Orton, B. M. Herrera, I. M. Yee, W. Valdar, S. V. Ramagopalan, A. D. Sadovnick and G. C. Ebers, “Sex ratio of multiple sclerosis in Canada: a longitudinal study,” vol. 5, no. 11, 2006.

[8] S. V. Ramagopalan, I. M. Yee, D. A. Dyment, S. M. Orton, R. A. Marrie, A. D. Sadovnick and G. C. Ebers, “Parent-of-origin effect in multiple sclerosis: Observations from interracial matings,” vol. 73, no. 8, 2009.

[9] Y. Nagasaka, K. Dillner, H. Ebise, R. Teramoto, H. Nakagawa, L. Lilius, K. Axelman, C. Forsell, A. Ito, B. Winblad, T. Kimura and C. Graff, “A unique gene expression signature discriminates familial Alzheimer’s disease mutation carriers from their wild-type siblings,” vol. 102, no. 41, 2005.

[10] M. P. Pender and S. R. Burrows, “Epstein–Barr virus and multiple sclerosis: potential opportunities for immunotherapy,” vol. 3, no. 10, 2014.

[11] E. Garcion, N. Wion-Barbot, C. N. Montero-Menei, F. Berger and D. Wion, “New clues about vitamin D functions in the nervous system,” vol. 13, no. 3, 2002.

[12] H. E. Hanwell and B. Banwell, “Assessment of evidence for a protective role of vitamin D in multiple sclerosis,” vol. 1812, no. 2, 2011.

[13] A. Manouchehrinia, C. R. Tench, J. Maxted, R. H. Bibani, J. Britton and C. S. Constantinescu, “Tobacco smoking and disability progression in multiple sclerosis: United Kingdom cohort study,” vol. 136, no. 7, 2013.

[14] N. J. Olsen, J. H. Moore and T. M. Aune, “Gene expression signatures for autoimmune disease in peripheral blood mononuclear cells,” vol. 6, no. 3, 2004.

[15] Y. Tang, D. L. Gilbert, T. A. Glauser, A. D. Hershey and F. R. Sharp, "Blood gene expression profiling of neurologic diseases: A pilot microarray study," vol. 62, no. 2, 2005.

[16] Y. Tang, H. Xu, X. Du, L. Lit, W. Walker, A. Lu, R. Ran, J. P. Gregg, M. Reilly, A. Pancioli, J. C. Khoury, L. R. Sauerbeck, J. A. Carrozzella, J. Spilker, J. Clark, K. R. Wagner, E. C. Jauch, D. J. Chang, P. Verro, J. P. Broderick and F. R. Sharp, "Gene expression in blood changes rapidly in neutrophils and monocytes after ischemic stroke in humans: A microarray study," vol. 26, no. 8, 2006.

[17] E. Pereira, M. C. Tamia-Ferreira, R. S. Cardoso, S. S. Mello, E. T. Sakamoto-Hojo, G. A. Passos and E. A. Donadi, "Immunosuppressive therapy modulates T lymphocyte gene expression in patients with systemic lupus erythematosus," vol. 113, no. 1, 2004.

[18] M. Rotger, K. K. Dang, J. Fellay, E. L. Heinzen, S. Feng, P. Descombes, K. V. Shianna, D. Ge, H. F. Günthard, D. B. Goldstein and A. Telenti, "Genome-wide Mrna expression correlates of viral control in CD4+ T-Cells from HIV-1-infected individuals," vol. 6, no. 2, 2010.

[19] S. Schmidt, J. Rainer, S. Riml, C. Ploner, S. Jesacher, C. Achmüller, E. Presul, S. Skvortsov, R. Crazzolara, M. Fiegl, T. Raivio, O. A. Jänne, S. Geley, B. Meister and R. Kofler, "Identification of glucocorticoid-response genes in children with acute lymphoblastic leukemia," vol. 107, no. 5, 2006.

[20] Y. Tian, M. L. Apperson, B. P. Ander, D. Liu, B. S. Stomova, G. C. Jickling, R. Enriquez, M. A. Agius and F. R. Sharp, "Differences in exon expression and alternatively spliced genes in blood of multiple sclerosis compared to healthy control subjects," vol. 230, no. 1-2, 2011.

[21] L. G. van Baarsen, T. C. van der Pouw Kraan, J. J. Kragt, J. M. Baggen, F. Rustenburg, T. Hooper, J. F. Meilof, M. J. Fero, C. D. Dijkstra, C. H. Polman and C. L. Verweij, "A subtype of multiple sclerosis defined by an activated immune defense program," vol. 7, no. 6, 2006.

[22] L. Ottoboni, B. T. Keenan, P. Tamayo, M. Kuchroo, J. P. Mesirov, G. J. Buckle, S. J. Khoury, D. A. Hafler, H. L. Weiner and P. L. De Jager, "An RNA profile identifies two subsets of multiple sclerosis patients differing in disease activity," vol. 4, no. 153, 2012.

[23] N. Rubanova, G. Pinna, J. Kropp, A. Campalans, J. P. Radicella, A. Polesskaya, A. Harel-Bellan and N. Morozova, "MasterPATH: Network analysis of functional genomics screening data," vol. 21, no. 1, 2020.

[24] W. Zhang, J. Chien, J. Yong and R. Kuang, "Network-based machine learning and graph theory algorithms for precision oncology," vol. 1, no. 1, 2017.

[25] A. L. Barabási, N. Gulbahce and J. Loscalzo, "Network medicine: A network-based approach to human disease," vol. 12, no. 1, 2011.

[26] G. A. Pavlopoulos, M. Secrier, C. N. Moschopoulos, T. G. Soldatos, S. Kossida, J. Aerts, R.

Schneider and P. G. Bagos, "Using graph theory to analyze biological networks," vol. 4, no. 1, 2011.

[27] Y. Fukuoka, D. Takei, H. Ogawa, « A two-step drug repositioning method based on a protein-protein interaction network of genes shared by two diseases and the similarity of drugs,» vol. 9,2013.

[28] L. Chen, L. Yuan, K. Qian, G. Qian, Y. Zhu, C. L. Wu, H. C. Dan, Y. Xiao and X. Wang, “Identification of biomarkers associated with pathological stage and prognosis of clear cell renal cell carcinoma by co-expression network analysis,” vol. 9, no. APR, 2018.

[29] B. He, J. Yin, S. Gong, J. Gu, J. Xiao, W. Shi, W. Ding, Y. He and E. Janczewska, “Bioinformatics analysis of key genes and pathways for hepatocellular carcinoma transformed from cirrhosis,” vol. 96, no. 25, 2017.

[30] Y. Liu, H. Y. Gu, J. Zhu, Y. M. Niu, C. Zhang and G. L. Guo, “Identification of Hub Genes and Key Pathways Associated With Bipolar Disorder Based on Weighted Gene Co-expression Network Analysis,” vol. 10, 2019.

[31] D. M. Camacho, K. M. Collins, R. K. Powers, J. C. Costello and J. J. Collins, “Next-Generation Machine Learning for Biological Networks,” vol. 173, no. 7, 2018.

[32] Z. Shang, W. Sun, M. Zhang, L. Xu, X. Jia, R. Zhang and S. Fu, “Identification of key genes associated with multiple sclerosis based on gene expression data from peripheral blood mononuclear cells,” vol. 2020, no. 2, 2020.

[33] K. Chai, X. Zhang, H. Tang, H. Gu, W. Ye, G. Wang, S. Chen, F. Wan, J. Liang and D. Shen, “The Application of Consensus Weighted Gene Co-expression Network Analysis to Comparative Transcriptome Meta-Datasets of Multiple Sclerosis in Gray and White Matter,” vol. 13, 2022.

[34] E. J. deAndrés-Galiana, G. Bea, J. L. Fernández-Martínez and L. N. Saligan, “Analysis of defective pathways and drug repositioning in Multiple Sclerosis via machine learning approaches,” vol. 115, 2019.

[35] D. Nickles, H. P. Chen, M. M. Li, P. Khankhanian, L. Madireddy, S. J. Caillier, A. Santaniello, B. A. Cree, D. Pelletier, S. L. Hauser, J. R. Oksenberg and S. E. Baranzini, “Blood RNA profiling in a large cohort of multiple sclerosis patients and healthy controls,” vol. 22, no. 20, 2013.

[36] M. Xu, T. Ouyang, K. Lv and X. Ma, “Integrated WGCNA and PPI Network to Screen Hub Genes Signatures for Infantile Hemangioma,” vol. 11, 2021.

[37] A. S. Nangraj, G. Selvaraj, S. Kaliamurthi, A. C. Kaushik, W. C. Cho and D. Q. Wei, “Integrated PPI- and WGCNA-Retrieval of Hub Gene Signatures Shared Between Barrett’s Esophagus and Esophageal Adenocarcinoma,” vol. 11, 2020.

[38] G. Fiscon and P. Paci, “SAveRUNNER: an R-based tool for drug repurposing,” vol. 22, no. 1, 2021.

[39] F. Crick, "Central dogma of molecular biology," vol. 227, no. 5258, 1970.

[40] Y. F. Leung and D. Cavalieri, “Fundamentals of Cdna microarray data analysis,” vol. 19, no. 11, 2003.

[41] C. M. Bishop, Bishop, C. M. (2006). Pattern Recognition and Machine Learning. (M. Jordan, J. Kleinberg, & B. Schölkopf, Eds.)Pattern Recognition (Vol. 4, p. 738). Springer. Doi:10.1117/1.2819119Pattern Recognition and Machine Learning, vol. 4, 2006.

[42] H. H. Rashidi, N. K. Tran, E. V. Betts, L. P. Howell and R. Green, “Artificial Intelligence and Machine Learning in Pathology: The Present Landscape of Supervised Methods,” vol. 6, 2019.

[43] E. Vayena, A. Blasimme and I. G. Cohen, “Machine learning in medicine: Addressing ethical challenges,” vol. 15, no. 11, 2018.

[44] L. Blonde, K. Khunti, S. B. Harris, C. Meizinger and N. S. Skolnik, “Interpretation and Impact of Real-World Clinical Data for the Practicing Clinician,” vol. 35, no. 11, 2018.

[45] J. Rigelsford, “Pattern Recognition: Concepts, Methods and Applications,” vol. 22, no. 4, 2002.

[46] I. Iguyon and A. Elisseeff, “An introduction to variable and feature selection,” vol. 3, 2003.

[47] M. S. Rao and B. Reddy, “Comparative Analysis of Pattern Recognition Methods : An Overview,” vol. 2, no. 3, 2011.

[48] P. Mahajan, “Applications of Pattern Recognition Algorithm in Health and Medicine: A,” 2016.

[49] M. Paolanti and E. Frontoni, “Multidisciplinary Pattern Recognition applications: A review,” vol. 37, 2020.

[50] B. M. Bolstad, R. A. Irizarry, M. Åstrand and T. P. Speed, “A comparison of normalization methods for high density oligonucleotide array data based on variance and bias,” vol. 19, no. 2, 2003.

[51] I. Guyon and A. Elisseeff, “Feature Extraction, Foundations and Applications: An introduction to feature extraction,” vol. 207, 2006.

[52] Y. Saeys, I. Inza and P. Larrañaga, “A review of feature selection techniques in bioinformatics,” vol. 23, no. 19, 2007.

[53] E. R. Dougherty, “Feature-selection overfitting with small-sample classifier design,” vol. 20, no. 6, 2005.

[54] G. Chu, J. Li, B. Narasimhan, R. Tibshirani and V. Tusher, “SAM — Significance Analysis of Microarrays — Users guide and technical document,” 2011.

[55] V. G. Tusher, R. Tibshirani and G. Chu, “Significance analysis of microarrays applied to the ionizing radiation response,” vol. 98, no. 9, 2001.

[56] R. Garcia-Dias, S. Vieira, W. H. Lopez Pinaya and A. Mechelli, “Chapter 13 — Clustering analysis,” 2020.

[57] M. B. Eisen, P. T. Spellman, P. O. Brown and D. Botstein, “Cluster analysis and display of genome-wide expression patterns,” vol. 95, no. 25, 1998.

[58] A. Hartigan and M. A. Wong, “A K-Means Clustering Algorithm,” vol. 28, no. 1, 1979.

[59] J. E. Gentle, L. Kaufman and P. J. Rousseuw, “Finding Groups in Data: An Introduction to Cluster Analysis.,” vol. 47, no. 2, 1991.

[60] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander and T. R. Golub, “Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation,” vol. 96, no. 6, 1999.

[61] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” vol. 2, no. 2, 1998.

[62] H. Sanz, C. Valim, E. Vegas, J. M. Oller and F. Reverter, “SVM-RFE: Selection and visualization of the most relevant features through non-linear kernels,” vol. 19, no. 1, 2018.

[63] M. Somvanshi, P. Chavan, S. Tambade and S. V. Shinde, “A review of machine learning techniques using decision tree and support vector machine,” 2017.

[64] J. R. Quinlan, “J. Ross Quinlan_C4.5_ Programs for Machine Learning.pdf,” vol. 5, no. 3, 1993.

[65] Thomas G. Dietterich, “An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization,” vol. 40, 2000.

[66] Y. Xiao, “A Tutorial on Analysis and Simulation of Boolean Gene Regulatory Network Models,” vol. 10, no. 7, 2009.

[67] I. Albert, J. Thakar, S. Li, R. Zhang and R. Albert, “Boolean network simulations for life scientists,” vol. 3, 2008.

[68] J. H. Maindonald, “Bayesian Artificial Intelligence, Second Edition by Kevin B. Korb, Ann E. Nicholson,” vol. 79, no. 3, 2011.

[69] T. S. Detroja, H. Gil-Henn, and A. O. Samson, “Text-Mining Approach to Identify Hub Genes of Cancer Metastasis and Potential Drug Repurposing to Target Them,” J. Clin. Med., vol. 11(8), pp. 2130, Apr 2022.

[70] G. K. Smyth, “Linear models and empirical bayes methods for assessing differential expression in microarray experiments,” vol. 3, no. 1, 2004.

[71] K. Chrominski and M. Tkacz, "Comparison of high-level microarray analysis methods in the context of result consistency," vol. 10, no. 6, 2015.

[72] A. Foroushani, R. Agrahari, R. Docking, L. Chang, G. Duns, M. Hudoba, A. Karsan and H. Zare, Large-scale gene network analysis reveals the significance of extracellular matrix pathway and homeobox genes in acute myeloid leukemia: An introduction to the Pigengene package and its applications, vol. 10, 2017.

[73] A. Bhar, M. Haubrock, A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay and E. Wingender, "Coexpression and coregulation analysis of time-series gene expression data in estrogen-induced breast cancer cell," vol. 8, no. 1, 2013.

[74] M. C. Oldham, S. Horvath and D. H. Geschwind, "Conservation and evolution of gene coexpression networks in human and chimpanzee brains," vol. 103, no. 47, 2006.

[75] P. Langfelder and S. Horvath, "WGCNA: An R package for weighted correlation network analysis," vol. 9, 2008.

[76] B. Zhang and S. Horvath, "Statistical Applications in Genetics and Molecular Biology 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.

[77] https://string-db.org/

[78] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski and T. Ideker, "Cytoscape: A software Environment for integrated models of biomolecular interaction networks," vol. 13, no. 11, 2003.

[79] C. H. Chin, S. H. Chen, H. H. Wu, C. W. Ho, M. T. Ko and C. Y. Lin, "cytoHubba: Identifying hub objects and sub-networks from complex interactome," vol. 8, no. 4, 2014.

[80] C. Y. Lin, C. H. Chin, H. H. Wu, S. H. Chen, C. W. Ho and M. T. Ko, "Hubba: hub objects analyzer—a framework of interactome hubs identification for network biology.," vol. 36, no. Web Server issue, 2008.

[81] G. D. Bader and C. W. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," vol. 4, 2003.

[82] I. Xenarios, Ł. Salwínski, X. J. Duan, P. Higney, S. M. Kim and D. Eisenberg, "DIP, the Database of Interacting Proteins: A research tool for studying cellular networks of protein interactions," vol. 30, no. 1, 2002.

[83] T. Igarashi and T. Kaminuma, "Development of a cell signaling networks database.," 1997.

[84] A. Wagner and D. A. Fell, "The small world inside large metabolic networks," vol. 268, no. 1478, 2001.

[85] G. W. Flake, S. Lawrence, C. Lee Giles and F. M. Coetzee, "Self-organization and

identification of web communities," vol. 35, no. 3, 2002.

[86] Z. Zhou and A. A. Amini, "Analysis of spectral clustering algorithms for community detection: The general bipartite setting," vol. 20, 2019.

[87] https://dgidb.genome.wustl.edu/

[88] http://stitch.embl.de/

[89] https://www.ncbi.nlm.nih.gov/geo/

[90] M. Acquaviva, R. Menon, M. Di Dario, G. Dalla Costa, M. Romeo, F. Sangalli, B. Colombo, L. Moiola, V. Martinelli, G. Comi and C. Farina, "Inferring Multiple Sclerosis Stages from the Blood Transcriptome via Machine Learning," vol. 1, no. 4, 2020.

[91] M. Gurevich, G. Miron, R. Z. Falb, D. Magalashvili, M. Dolev, Y. Stern and A. Achiron, "Transcriptional response to interferon beta-1a treatment in patients with secondary progressive multiple sclerosis," vol. 15, no. 1, 2015.

[92] Z. Shang, W. Sun, M. Zhang, L. Xu, X. Jia, R. Zhang and S. Fu, "Identification of key genes associated with multiple sclerosis based on gene expression data from peripheral blood mononuclear cells," vol. 2020, no. 2, 2020.

[93] F. Ye, J. Liang, J. Li, H. Li and W. Sheng, "Development and Validation of a Five-Gene Signature to Predict Relapse-Free Survival in Multiple Sclerosis," vol. 11, 2020.

[94] A. Javed and A. T. Reder, "Therapeutic role of beta-interferons in multiple sclerosis," vol. 110, no. 1, 2006.

[95] T. Stavroula, E. S. Bei and M. E. Zervakis, "A 21 hub gene Signature in Multiple Sclerosis_2022," 2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), 2022, pp. 1-5, doi: 10.1109/BHI56158.2022.9926949.

[96] X. Feng, R. Bao, L. Li, F. Deisenhammer, B. G. Arnason and A. T. Reder, "Interferon-β corrects massive gene dysregulation in multiple sclerosis: Short-term and long-term effects on immune regulation and neuroprotection: Short running title: Interferon-β corrects gene dysregulation in multiple sclerosis," vol. 49, 2019.

[97] M. Severa, C. Farina, M. Salvetti and E. M. Coccia, "Three Decades of Interferon-β in Multiple Sclerosis: Can We Repurpose This Information for the Management of SARS-CoV2 Infection?," vol. 11, 2020.

[98] M. E. Deerhake, D. D. Biswas, W. E. Barclay and M. L. Shinohara, "Pattern Recognition Receptors in Multiple Sclerosis and Its Animal Models," vol. 10, 2019.

[99] P. A. Faye, F. Poumeaud, F. Miressi, A. S. Lia, C. Demiot, L. Magy, F. Favreau and F. G. Sturtz, “Focus on 1,25-dihydroxyvitamin D3 in the peripheral nervous system,” vol. 13, no. APR, 2019.

[100] M.K. Singh, T. F Scott, W. A LaFramboise., F. Z Hu., J. C Post. And G. D Ehrlich. “Gene expression changes in peripheral blood mononuclear cells from multiple sclerosis patients undergoing beta-interferon therapy.” Vol. 258,no.1-2,2007

[101] M. Naegele, K. Tillack, S. Reinhardt, S. Schippling, R. Martin and M. Sospedra, “Neutrophils in multiple sclerosis are characterized by a primed phenotype,” vol. 242, no. 1-2, 2012.

[102] Geiger M., Hayter E., Martin R.S., Spence D. “Red blood cells in type 1 diabetes and multiple sclerosis and technologies to measure their emerging roles”, 2022

[103] I. Jridi, K. Canté-Barrett, K. Pike-Overzet and F. J. Staal, “Inflammation and Wnt Signaling: Target for Immunomodulatory Therapy?,” vol. 8, 2021.

[104] A. D. Kohn and R. T. Moon, “Wnt and calcium signaling: β-Catenin-independent pathways,” vol. 38, no. 3-4 SPEC. ISS., 2005.

[105] J. E. Lengfeld, S. E. Lutz, J. R. Smith, C. Diaconu, C. Scott, S. B. Kofman, C. Choi, C. M. Walsh, C. S. Raine, I. Agalliu and D. Agalliu, “Endothelial Wnt/β-catenin signaling reduces immune cell infiltration in multiple sclerosis,” vol. 114, no. 7, 2017.

[106] M. D. Laksitorini, V. Yathindranath, W. Xiong, S. Hombach-Klonisch and D. W. Miller, "Modulation of Wnt/β-catenin signaling promotes blood-brain barrier phenotype in cultured brain endothelial cells," vol. 9, no. 1, 2019.

APPENDIX

DGIdb database drug-gene interaction results.

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS
STAT1	GARCINOL		DTC	
	GUTTIFERONE K		DTC	
	PICOPLATIN		CIViC	15726096
	CISPLATIN		CIViC	15726096
	CHEMBL85826		DTC	
	IPRIFLAVONE		DTC	
OASL	RIBAVIRIN		PharmGKB	21993426
CCR1	AZD4818	antagonist	ChemblInteractions	
	BMS-817399	antagonist	TTD	
	CCX354	antagonist	TdgClinicalTrial|ChemblInteractions|TTD	
	TERPYRIDINE		DTC	22957890
	CHEMBL2205805		DTC	22957890
CASP1	NIVOCASAN	inhibitor	ChemblInteractions|TTD	

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS
	EMRICASAN	inhibitor	ChemblInteractions	
	PRALNACASAN	inhibitor	TTD	17845807
	BERKELEYAMIDE C		DTC	18330993
	CHEMBL337173		DTC	
	4-CHLOROMERCURIBENZOIC ACID		DTC	
	BERKELEYDIONE		DTC	17970594
	GOSSYPOL		DTC	
	MESALAMINE		DTC	
	BERKELEYACETAL A		DTC	17970594
	DIACEREIN		TTD	
	VERMISTATIN		DTC	22295871
	BERKELEYACETAL B		DTC	17970594
	BELNACASAN		TdgClinicalTrial|TTD	
	CHEMBL578512		DTC	
	CHEMBL429095		DTC	
	JUGLONE		DTC	
	ISOBOLDINE		DTC	

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS
	CHEMBL415893		DTC	10386941
	BERKELEYAMIDE B		DTC	18330993
	BERKELEYACETAL C		DTC	17970594
	CHEMBL580421		DTC	
	BERKELEYTRIONE		DTC	17970594
CXCL10	NI-0801	inhibitor	ChemblInteractions|TTD	
	REGRAMOSTIM		NCI	11591765
	METHYLPREDNISOLONE		NCI	17220550
	ANTIBIOTIC		NCI	10634213
	RITONAVIR		NCI	11141242
	STAVUDINE		NCI	11141242
	ATORVASTATIN		NCI	10559511
	ATROPINE		NCI	15315164
	TESTOSTERONE		NCI	9681518
	OXALIPLATIN		NCI	16101140
	ELDELUMAB		TdgClinicalTrial|TTD	
	ZIDOVUDINE		NCI	11141242

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS
NT5C3A	CYTARABINE		PharmGKB	25000516
	IDARUBICIN		PharmGKB	25000516
	GEMCITABINE		PharmGKB	22838949

Table4.19 Genes that have drug interactions case for the Case “untreated MS vs Interferon treated MS patients”

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS
PEMT	CANTUZUMAB MERTANSINE		ChemblInteractions	
	HUHMFG1		ChemblInteractions	
	CANTUZUMAB RAVTANSINE		ChemblInteractions	
	SONTUZUMAB		ChemblInteractions	
	PEMTUMOMAB		ChemblInteractions	
	AR-20.5		ChemblInteractions	
LST1	ABACAVIR		PharmGKB	
B2M	PEMBROLIZUMAB		CIViC	27433843
	THYROGLOBULIN		NCI	9609129
	AMIKACIN		NCI	7672871

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS
CDA	CYTARABINE		NCI|PharmGKB	21325291|21521023 |12008078|22304580| 22379997|25003625| 19458626|23651026| 23230131|18473752
	GEMCITABINE		NCI	12477049
	DEOXYCYTIDINE		NCI	12008078
	TETRAHYDROURIDINE		NCI|TTD	2932216
	CAPECITABINE		PharmGKB	21325291|24167597| 28347776| 18473752|23736036
	AZACITIDINE		PharmGKB	25850965
S100A9	TASQUINIMOD		TTD	24162378
	PAQUINIMOD		TTD	

TABLE4.20 GENES THAT HAVE DRUG INTERACTIONS FOR THE CASE "UNTREATED MS PATIENTS VS HEALTHY CONTROLS"

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS

IL1RN	METHOTREXATE		NCI	8877917
	HALOPERIDOL		PharmGKB	27023437
	DIACEREIN		TdgClinicalTrial	
CTSG	MANNITOL		NCI	3142269
	CHEMBL374027		TTD	
CXCL8	ABX-IL8	inhibitor	ChemblInteractions|TTD	
	HUMAX-IL8	inhibitor	ChemblInteractions	
	LEFLUNOMIDE		NCI	10902750
	YANGONIN		DTC	
	E319		DTC	
	FOSCARNET		NCI	10630964
	NAPROXEN		NCI	11852880
	ALDRIN		DTC	
	COLCHICINE		DTC	
	MIDAZOLAM		NCI	9620522
	FENTANYL		NCI	9527747
	ACETAMINOPHEN		NCI	15878691
	CORONOPILIN		DTC	

	DIPYRIDAMOLE		NCI	10660968
	IBUPROFEN		TTD	
	IONOMYCIN		NCI	7510691
	CHLORDANE		DTC	
	DANAZOL		NCI	16161451
	CHEMBL1902074		DTC	
	OMEPRAZOLE		NCI	17122965
	DINITRO CRESOL		DTC	
	QUESTIOMYCIN B		DTC	
	FENRETINIDE		NCI	16979119
	HEPTACHLOR		DTC	
	PYROGALLOL		DTC	
	CANERTINIB		NCI	15956251
	HYDROQUINONE		DTC|NCI	17118622
	ENDOSULFAN		DTC	
	EMODIN		DTC	
	LANSOPRAZOLE		NCI	17122965
	RETINAL		DTC	

	HARMINE HYDROCHLORIDE		DTC	
	PACLITAXEL		NCI	9271387
	BEVACIZUMAB		PharmGKB	23584701
	PAMIDRONIC ACID		NCI	12006522
	TALC		NCI	17000556
	TRETINOIN		NCI	8900181
	SUNITINIB		PharmGKB	26387812
	CETUXIMAB		NCI	10614716|15908664 |10037173
	CHEMBL1579130		DTC	
	ALPRAZOLAM		NCI	12218154
	METHIMAZOLE		NCI	11453524
	RETINOL		DTC	
	RIBAVIRIN		DTC	
	TERFENADINE		NCI	8919641
	DICYCLOHEXYLCARBODIIMIDE		DTC	
	CEFTRIAXONE		NCI	8011012

	ASPIRIN		NCI	12576442
	CLARITHROMYCIN		NCI	12003967
	DACARBAZINE		DTC	
	PENTOXIFYLLINE		NCI	12576442
	CIDOFOVIR		NCI	10630964
	BROXURIDINE		DTC	
	TROGLITAZONE		NCI	12364456
	DICHLORVOS		DTC	
	VERAPAMIL		NCI	2686646
CXCR1	LADARIXIN	modulator	ChemblInteractions	
	REPARIXIN	allosteric modulator|modulator	ChemblInteractions|TTD	
	NAVARIXIN	antagonist	TdgClinicalTrial|TTD	
	NAVARIXIN	antagonist	ChemblInteractions	
	IBUPROFEN		TTD	
ELANE	SIVELESTAT	inhibitor	DTC|TdgClinicalTrial|TTD	23350733
	DEPELESTAT		TTD	
	SYMPLOSTATIN 5		DTC	23350733
	CHEMBL310871		DTC	17535802

	NICOTINE		NCI	8912774
	TIPRELESTAT		TTD	
	ERDOSTEINE		TdgClinicalTrial	
	NIFEDIPINE		NCI	9796781|8833599
IL1R2	ANAKINRA		TEND	
LY96	ERITORAN TETRASODIUM	antagonist	ChemblInteractions	
MMP9	MARIMASTAT	inhibitor	TdgClinicalTrial|TEND	17234180|12763661|11752352
	PRINOMASTAT	vaccine	TALC	
	ANDECALIXIMAB	inhibitor|antibody	ChemblInteractions|TTD	
	S-3304	vaccine	TALC	
	CURCUMIN PYRAZOLE		DTC	19128977
	TOZULERISTIDE		TTD	
	CURCUMIN		TTD	
	INCYCLINIDE		TdgClinicalTrial	
	BEVACIZUMAB		CIViC	26921265
	CARBOXYLATED GLUCOSAMINE		DTC	16616490

	DEMETHYLWEDELOLACTONE		DTC	22926226
	CELECOXIB		PharmGKB	22336956
MPO	DIMETHYL SULFOXIDE		NCI	1845843
	PSORALEN		NCI	15865234
	TOLMETIN		NCI	6266970
	DICLOFENAC		NCI	2173589
	DOXYCYCLINE		NCI	14564835
	ASULACRINE		NCI	1333205
	NIMESULIDE		NCI	17176264
	PYRAZINAMIDE		NCI	2832129
	PROPYLTHIOURACIL		DTC	26509551
	FLUDARABINE		NCI	15608444
	LORATADINE		NCI	17159802
	OCTREOTIDE		NCI	15003363
	TRIMETHOPRIM		NCI	7425598
	THEOPHYLLINE		NCI	8630596
	LITHIUM		NCI	8224362
	LIDOCAINE		NCI	8973808

	TENECTEPLASE		NCI	16650886
	FLUTAMIDE		NCI	16330533
	FENTANYL		NCI	8391745
PAK1	CENISERTIB		DTC	
	TAE-684		DTC	
	AZD-1152-HQPA		DTC	
	TOZASERTIB		DTC	
	RG-1530		DTC	
	ILORASERTIB		DTC	
	LAUROGUADINE		DTC	
	PF-00562271		DTC	
	MLN-8054		DTC	
	R-406		DTC	
PTAFR	RUPATADINE	antagonist	TTD	8996188
	ISRAPAFANT	antagonist	TTD	
	MINOPAFANT	antagonist	ChemblInteractions	
	LEXIPAFANT		TTD	
	DERSALAZINE		TTD	21790535

	TICLOPIDINE		TTD	
S100A8	METHOTREXATE		NCI	14722212
CEACAM3	ARCITUMOMAB		TTD	
HNMT	AMODIAQUINE	inhibitor	TTD	6789797|1203620|17222819|11752352
	ASPIRIN		PharmGKB	19178400
	METOPRINE		TTD	10592235
	DABIGATRAN		DTC	22494098
	DIPHENHYDRAMINE		TTD	
SLC22A16	FLUOROURACIL		PharmGKB	
	CYCLOPHOSPHAMIDE		PharmGKB	28036387|20179710
	DOXORUBICIN		PharmGKB	28036387|17559346|20179710
TNFSF13	ATACICEPT	inhibitor	TdgClinicalTrial|ChemblInteractions|TTD	
LPAR1	BMS-986020	antagonist	TTD	
S100A12	ATOGEPANT		TTD	
	RIMEGEPANT		TTD	
	METHOTREXATE		NCI	15077313
	EPTINEZUMAB		TTD	

	UBROGEPANT		TTD	
CD14	IC14	inhibitor	TdgClinicalTrial|ChemblInteractions|TTD	
	LOVASTATIN		NCI	7506029

TABLE4.21 GENES THAT HAVE DRUG INTERACTIONS FOR THE CASE “UNTREATED PPMS PATIENTS VS HEALTHY CONTROLS”

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS
CYP2J2	TERFENADINE	inhibitor	PharmGKB	15861034
	THIORIDAZINE		PharmGKB	19923256
	TACROLIMUS		PharmGKB	28316087
	DICLOFENAC		PharmGKB	15861034
	AMIODARONE		PharmGKB	19923256
	NABUMETONE		PharmGKB	19923256
	ASTEMIZOLE		PharmGKB	15861034
	ALBENDAZOLE		PharmGKB	19923256
	MESORIDAZINE		PharmGKB	19923256
	DANAZOL		PharmGKB	19923256
EPHX2	FULVESTRANT		DTC	23684894
	6BIO		DTC	24697244
	AR9281		TdgClinicalTrial|TTD	10592235

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS
	LITHIUM		PharmGKB	29121268
	ALOE-EMODIN		DTC	26372074
	ALOIN		DTC	26372074
NBEA	METFORMIN		PharmGKB	29650774
XK	ENSITUXIMAB		TTD	

TABLE4.22 GENES THAT HAVE DRUG INTERACTIONS FOR THE CASE "UNTREATED SPMS PATIENTS VS HEALTHY CONTROLS"

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS
SLC4A1	ATENOLOL		PharmGKB	
	METOPROLOL		PharmGKB	
BCR	IMATINIB	inhibitor	TALC|DTC|PharmGKB|FDA	15206509|22148584|12600228|23226582|20072827|24681986
	DASATINIB	inhibitor	TALC|PharmGKB|FDA	15256671
	PONATINIB HYDROCHLORIDE	inhibitor	ChemblInteractions	
	PONATINIB	inhibitor	TALC|PharmGKB|FDA	23409026
	BOSUTINIB	inhibitor	PharmGKB|FDA	
	SARACATINIB	inhibitor	TALC	
	VINCRISTINE		PharmGKB|FDA	

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS
	BUSULFAN		PharmGKB|FDA	
	OMACETAXINE MEPESUCCINATE		PharmGKB	
	BLINATUMOMAB		PharmGKB|FDA	
	NILOTINIB		PharmGKB|FDA	
	CHEMBL483847		DTC	16415863
CA1	ACETAZOLAMIDE SODIUM	inhibitor	ChemblInteractions	
	POLMACOXIB	inhibitor	TdgClinicalTrial|ChemblInteractions|TTD	
	ZONISAMIDE	inhibitor	TEND	15837316|17762320|18537527|18343915|18782051|17582922|8494570|18162396
	METHAZOLAMIDE	inhibitor	TdgClinicalTrial|ChemblInteractions|TEND|TTD	10533697|15110853|9336012|14684332|10649985
	ETHOXZOLAMIDE	inhibitor	TdgClinicalTrial|ChemblInteractions|TEND|TTD	12956733|10649985|7929150|19520577|6816217|10995826
	ACETAZOLAMIDE	inhibitor	TdgClinicalTrial|ChemblInteractions|TEND|TTD	10713865|12956733|18336310|8667211|10651143|11430635
	DICHLORPHENAMIDE	inhibitor	TdgClinicalTrial|ChemblInteractions|TEND|TTD	19019313|9336012|14684332|19648295|17228881
	TRICHLORMETHIAZIDE	inhibitor	TdgClinicalTrial|TEND	19119014|17139284|17016423

GENE	DRUG	INTERACTION_TYPES	SOURCES	PMIDS
	METHOCARBAMOL	inhibitor	ChemblInteractions	1460006
	CHLOROTHIAZIDE	inhibitor	TdgClinicalTrial|TEND	10713865|10954127
	RESORCINOL		DTC	26073005
	MEDRONIC ACID		DTC	24813742
	PHENOL		DTC|TTD	26073005
	CURCUMIN		TTD	
	CATECHOL		DTC	26073005
	SULFAMIDE		TTD	
	PARABEN		TTD	
	LEVETIRACETAM		TEND	
IGF2BP2	REPAGLINIDE		PharmGKB	
KISS1R	BENZETHONIUM CHLORIDE		DTC	17266198

TABLE4.23 GENES THAT HAVE DRUG INTERACTIONS FOR THE CASE "UNTREATED PPMS PATIENTS VS HEALTHY CONTROLS"